IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 25 June 2022, accepted 5 July 2022, date of publication 7 July 2022, date of current version 18 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3189353

RESEARCH ARTICLE

Integration of CNN in a Dynamic Model-Based Controller for Control of a 2DOF Helicopter With Tail Rotor Perturbations

MARIO C. MAYA-RODRIGUEZ[1], MARIO A. LÓPEZ-PACHECO[1], YAIR LOZANO-HERNÁNDEZ[2], VICTOR G. SÁNCHEZ-MEZA[3], LUIS A. CANTERA-CANTERA[1], AND RENÉ TOLENTINO-ESLAVA[1]

[1]Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco, Instituto Politécnico Nacional, CDMX 07738, Mexico
[2]Unidad Profesional Interdisciplinaria de Ingeniería Campus Hidalgo, Instituto Politécnico Nacional, Hidalgo 42162, Mexico
[3]Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, Instituto Politécnico Nacional, CDMX 07340, Mexico

Corresponding author: Yair Lozano-Hernández (ylozanoh@ipn.mx)

This work was supported by the Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional (SIP-IPN) under Grants 20220039, 20221135 and 20221223. Victor G. Sánchez-Meza is a Consejo Nacional de Ciencia y Tecnología (CONACYT) fellow (CVU 964590) and is thankful for the support received.

ABSTRACT This paper shows a proposal for a control scheme for the trajectory tracking problem in a Two Degree of Freedom Helicopter (2DOFH). For this purpose, a control scheme based on a feedback linearization combined with a Generalized Proportional Integral (GPI) controller is used. In order to implement linearization by feedback, it is required to know and have access to all the physical 2DOFH parameters, however, angular velocity and viscous friction are often not available. Commonly, state observers are used to know the angular velocity, however, estimating friction results out to be more complex. Therefore, we propose the use of a Convolutional Neural Network (CNN) to estimate viscous friction and angular velocity. The variables estimated by the CNN are entered into both the GPI and feedforward controllers. Thus, the system is brought to a linear representation that directly relates the GPI control to the dynamics of perturbations and non-model parameters. Finally, results of numerical simulations are shown that validate the robustness of our scheme in the presence of disturbances in the tail rotor, as well as the advantages of using a feedforward control based on a CNN.

INDEX TERMS Friction estimation, neural networks, non-linear system, tail rotor disturbance, two degrees of freedom helicopter.

I. INTRODUCTION

In recent years, rotary wing Unmanned Aerial Vehicles (UAV) have been used in areas where human actions are restricted [1]–[3]. One of the most popular rotary wing UAV is the helicopter, some of its main features are: Ability to rotate on its own axis, levitate, take off and land vertically, and move sideways or backwards while in air. [4]. Despite the number of applications in which a helicopter can be used [5], fully autonomous scale helicopters are not yet available. This is due to non-linearities, strong coupling, uncertainties and aerodynamic disturbances associated with this vehicle [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Francesco Piccialli.

Nowadays, there are different study areas focused on helicopters, being of special interest the improvement of its stability, autonomous navigation, takeoff and landing control, trajectory tracking, disturbance rejection, among others [7]–[10]. To achieve the above, it is necessary to know the behavior of the disturbances that frequently occur, which greatly impairs the Tail Rotor (TR), causing stability problems during flight and limiting maneuvering space [11], [12].

Hence, different experimental prototypes have been developed to study helicopter dynamics [13]–[15]. One of the most popular consists of a Two Degree of Freedom Helicopter (2DOFH), which recreates the dynamic behavior of the helicopter in its pitch (θ) and yaw (ψ) rotations [15], [16]. The orientations are controlled by two Motor-Propeller

Pairs (MPP) located at the ends of a rigid shaft, which can rotate on the horizontal and vertical axis [17].

Several studies have focused on the design of control algorithms for helicopters. For example, in [18] an observer based on sliding modes proposes to estimate angular velocities, obtaining results close to the real ones. Regarding the control of these systems, in [19] an output feedback algorithm H_∞ is used to study the problem of disturbance attenuation. In [20], a control scheme based on the active rejection of disturbances for a 2DOFH is proposed. The algorithm contemplates the use of GPI-type observers to estimate the speed and disturbances caused by non-modeled dynamics and external disturbances.

The design of algorithms and control schemes for UAV, such as those mentioned above, require solid mathematical model foundations to develop a controller based on the model. If you have a reliable mathematical model, it is possible to design a controller based on the model, which tends to linearize the system [21]. However, these controllers require a priori knowledge of the mathematical model parameters. Parametric identification allows an attractive option to obtain a numerical approximation of the unknown coefficients, when prior knowledge is absent [22].

UAVs are used in operations in operations where several types of disturbances exist, which are produced due to environmental conditions and their non-linear dynamics. Furthermore, parameters such as the fiction of the actuators tend to change with respect to time; furthermore, not all variables can always be measured directly. This affects the performance of control algorithms based on the dynamic model [23].

Therefore, various works have been developed focused on the implementation of parametric identification techniques in UAVs. For example, [24] carries out the experimental identification of the rotational inertia and coefficient of friction of a quadrotor, which is achieved using Levenberg Marquardt optimization and quadratic optimization to minimize error criteria. Under another scheme, in [25] an extended Kalman filter is used to identify the linear and angular velocities of a quadrotor. Finally, in [26] the inertial parameters of a helicopter are identified using recursive least squares, highlighting that the method is capable of updating the estimated parameters while the vehicle is in flight.

As mentioned, there are different techniques used for parameter estimation. It should be emphasized that the implementation of model-based control schemes requires knowledge of the variables in real time, therefore, online estimation techniques shall be used [27]. Under this premise, machine learning techniques are a reliable alternative option to perform parametric identification online, with artificial neural networks being the most used.

One of the most widely used neural networks consists of the Multi-Layer Perceptron (MLP) architecture [28], however, the accuracy of its results is affected by problems in the learning algorithm such as local minimums or lack of information. Some works on parametric identification for aeronautical systems using MLP topology are: [29] and [30], showing the feasibility of incorporating this type of schemes into aircraft control algorithms. Therefore, the MLP has been modified, resulting in a Convolutional Neural Network (CNN) algorithm, which is a deep learning technique [31]. Although CNNs are mostly used for image processing, it has been reported that CNNs can also be used for the identification of non-linear systems [32].

Therefore, in this paper we propose the use of a CNN to estimate the coefficient of viscous friction, $\dot{\theta}$ and $\dot{\psi}$, which are introduced to a feedforward control, linearizing the system around all its equilibrium points. The feedforward controller is combined with a GPI to provide robustness in the presence of disturbances in the TR.

This work is organized as follows: section 2 describes system mathematical model of the system considering disturbances in the TR. Section 3 shows the proposed control scheme for trajectory tracking. In section 4 we describe the CNN design for the estimation of $\dot{\theta}$, $\dot{\psi}$ and viscous friction. Moreover, section 5 shows the comparison of two feedforward control schemes, highlighting the differences when using information from CNN. Finally, the last section contains the conclusions of this work.

II. MATHEMATICAL MODEL

Figure 1a shows the 2DOFH prototype used in this work, which has dimensions of 74 cm long and 35 cm high. In addition, it has a mechanical coupling to generate disturbances (see Fig. 1b and c).

Fig. 2 shows the free body diagram of the 2DOFH, as well as the parameters and variables of interest in the system. F_{TR} and F_{MR} are the thrust forces generated by TR and the Main Rotor (MR), respectively. F_g corresponds to the force of gravity and L_{cm} is the length from the center of mass to the pitch axis. r_y and r_p are the distance from TR and MR to θ, respectively. Finally, ψ is the angle around the Z axis and ϕ is the angle of incidence of the TR (ϕ directly affects the thrust force of the TR).

The displacement in the X, Y and Z axes depends on the variation of the 2DOFH angles and is expressed by

$$\begin{aligned} X &= L_{cm} c\psi c\theta, \\ Y &= L_{cm} s\psi c\theta, \\ Z &= L_{cm} s\theta, \end{aligned} \tag{1}$$

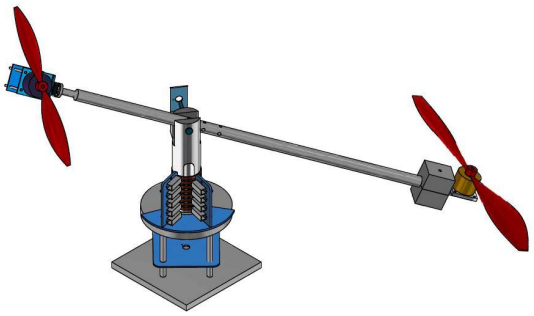
where $s\cdot = sin(\cdot)$ and $c\cdot = cos(\cdot)$. Subsequently, the dynamic model is obtained from the Euler-Lagrange (EL) form [33]

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau - B\dot{q}; \tag{2}$$

with $L = T - U$, T and U denote the kinetic and potential energies respectively, $\tau = [\tau_\theta \;\; \tau_\psi]^T$ are the generalized force associated with the generalized coordinates $q = [\theta \;\; \psi]^T$ and $B = [B_\theta \;\; B_\psi]^T$ is the viscous friction.

Next, the kinetic and potential energy are defined as:

$$\begin{aligned} T &= T_{rp} + T_{ry} + T_t, \\ U &= m_h L_{cm} g s\theta, \end{aligned} \tag{3}$$



(a) 2DOFH prototype.

(b) Coupling for disturbance in the TR.

(c) Photograph of perturbation coupling in the TR.

FIGURE 1. 2DOFH prototype design.

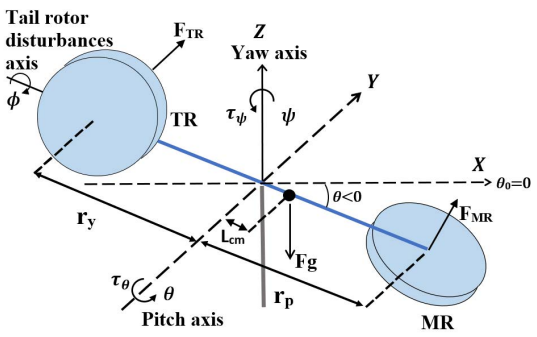


FIGURE 2. 2DOFH diagram.

where m_h is the mass of the helicopter, g is the force of gravity. T_t is the translational kinetic energy, T_{rp} and T_{ry} are the rotational kinetic energy in ψ and θ, defined as [34]:

$$\begin{aligned} T_{rp} &= \frac{1}{2}J_{eq\_p}\dot{\theta}^2, \\ T_{ry} &= \frac{1}{2}J_{eq\_y}\dot{\psi}^2, \\ T_t &= \frac{1}{2}m_h\left(V_x^2 + V_y^2 + V_z^2\right), \end{aligned} \tag{4}$$

with J_{eq_p} and J_{eq_y} equal to the moments of inertia when rotating over θ and ψ, respectively. The velocity on each axis is obtained with the rotation angles and equation (1). Therefore, total kinetic energy is defined as:

$$T = \frac{1}{2}J_{eq\_p}\dot{\theta}^2 + \frac{1}{2}J_{eq\_y}\dot{\psi}^2 + \frac{1}{2}m_hL_{cm}^2\left(\dot{\theta}^2 + (\dot{\psi}c\theta)^2\right). \tag{5}$$

Thus, equation (2) results in:

$$\begin{bmatrix} \ddot{\theta}(J_{eq\_p} + m_hL_{cm}^2) + m_hL_{cm}^2s\theta c\theta\dot{\psi}^2 + m_hgL_{cm}c\theta \\ \ddot{\psi}(J_{eq\_y} + m_h(c\theta L_{cm})^2) - 2m_hL_{cm}^2\dot{\theta}s\theta c\theta\dot{\psi} \end{bmatrix} = \begin{bmatrix} k_{pp} & k_{yy}s\phi + k_{py}c\phi \\ k_{yp} & k_{yy}c\phi + k_{py}s\phi \end{bmatrix} \begin{bmatrix} U_\theta \\ U_\psi \end{bmatrix} - \begin{bmatrix} B_\theta\dot{\theta} \\ B_\psi\dot{\psi} \end{bmatrix}, \tag{6}$$

where τ_θ and τ_ψ are the torque applied to each axis of rotation, U_θ and U_ψ are the control actions applied to the MR and TR, respectively. k_{pp} and k_{yy} are the thrust force constants in relation to the control action $[Nm/\%pwm]$ derived from the direct incidence of the rotor thrust of the rotor MR and TR on the axis θ and ψ, respectively. k_{py} and k_{yp} are the thrust force constants in relation to the control action $[Nm/\%pwm]$ derived from the torque generated by the rotation of the rotor propellers TR and MR in the axis θ and ψ [15], [35].

A. PARAMETRIC IDENTIFICATION USING CNN

The implementation of control algorithms based on the mathematical model requires knowledge of all parameters. If any parameter is unknown, the mathematical model must be adjusted to experimental data by a parameter identification algorithm. In this context, there are on-line and off-line methods based on the least squares method [36], [37].

Currently, Artificial Neural Networks (ANN) are used to model and identify parameters in dynamic systems, due to their ability to approximate nonlinear functions and predict variables [38]–[42]. Therefore, deep learning methods together with Neural Networks (NN) have been used for the identification of unknown parameters, unmodeled dynamics and external perturbations for control systems [43]–[46]. Under this approach, most identification applications use adaptive or recursive NN, another common deep neural network is the CNN [38], [47].

In particular, since the estimation of friction coefficient is a complicated task, in this work we propose the use of a CNN to estimate viscous friction and velocity at each degree of freedom of the helicopter. CNN training is performed offline (calculation of synapotic weights). After training, the CNN is implemented in line with the control scheme. In other words, viscous friction and speed are estimated online, these variables are introduced into the control scheme.

The estimation of the friction and velocity term is carried out using CNNs, for which, a different CNN is used for each of them, i.e. $\hat{\beta}_\theta = B_\theta\dot{\theta}$ and $\hat{\beta}_\psi = B_\psi\dot{\theta}$, with $\hat{\beta}_\theta$ and $\hat{\beta}_\psi$ are the output of the respectively CNN. Thus, we can estimate the velocities $\hat{\dot{\theta}}$ and $\hat{\dot{\psi}}$ in the form:

$$\begin{aligned} \hat{\dot{\theta}} &= \hat{\beta}_\theta/\overline{B}_\theta, \\ \hat{\dot{\psi}} &= \hat{\beta}_\psi/\overline{B}_\psi, \end{aligned} \tag{7}$$

where, $\overline{B}_\theta$ and $\overline{B}_\psi$ are estimated constants of the viscous friction of B_θ and B_ψ.

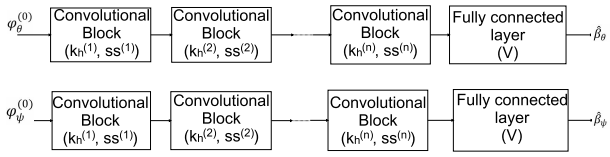


FIGURE 3. CNN for parametric estimation of $B_\theta\dot{\theta}$ and $B_\psi\dot{\psi}$.

Thus, (6) is rewritten as a function of the variables estimated by the CNN, resulting in:

$$\begin{bmatrix} \ddot{\theta}(J_{eq\_p} + m_h L_{cm}^2) + m_h L_{cm}^2 s\theta c\theta \hat{\dot{\psi}}^2 + m_h g L_{cm} c\theta \\ \ddot{\psi}(J_{eq\_y} + m_h(c\theta L_{cm})^2) - 2m_h L_{cm}^2 \hat{\dot{\theta}} s\theta c\theta \hat{\dot{\psi}} \end{bmatrix} = \begin{bmatrix} k_{pp} & k_{yy}s\phi + k_{py}c\phi \\ k_{yp} & k_{yy}c\phi + k_{py}s\phi \end{bmatrix} \begin{bmatrix} U_\theta \\ U_\psi \end{bmatrix} - \begin{bmatrix} \hat{\beta}_\theta \\ \hat{\beta}_\psi \end{bmatrix}, \quad (8)$$

To estimate the friction terms, we use the Nonlinear Auto Regresive Exogenous (NARX) model:

$$\hat{\beta} = \text{CNN}\left[\varphi^{(0)}\right], \quad (9)$$

where CNN[·] is the proposed strucuture of the CNN and $\varphi^{(0)}$ is the input, It is defined for each CNN as:

$$\begin{aligned} \varphi_\theta^{(0)}(\xi) &= \left[\tau_p(\xi) \cdots \tau_p(\xi - r_{\theta_1})\, \theta(\xi) \cdots \theta(\xi - r_{\theta_2})\right]^T, \\ \varphi_\psi^{(0)}(\xi) &= \left[\tau_y(\xi) \cdots \tau_y(\xi - r_{\psi_1})\, \psi(\xi) \cdots \psi(\xi - r_{\psi_2})\right]^T, \end{aligned} \quad (10)$$

ξ indicates the current instant of time, r_θ and r_ψ are the regression order for each data.

Fig. 3 shows the structure of the CNN, this being our proposal for the estimation of $\hat{\beta}_\theta$ and $\hat{\beta}_\psi$. Our proposal consists of two convolutional blocks ($n = 4$) followed by a fully connected layer.

Each convolutional block consists of two different layers, a convolutional layer and a subsample layer, with their respectively hyper-parameters. The first is the fundamental part of this algorithm [48], in it the convolution operation is carried out between the input of the convolutional block $\varphi^{(\ell-1)}$ and the filters of this layer $\kappa^{(\ell)}$ which theirs size are 3, thus generating the feature maps $\chi^{(\ell)}$ as follows:

$$\chi_h^{(\ell)} = \kappa_h^{(\ell)} \odot \varphi_h^{(\ell-1)}, \quad (11)$$

where the superscript (ℓ) indicates the current layer in which the operations are performed, $\odot$ represents the convolution operation and h represents how many filters there are present in the layer.

Once these maps have been generated, we proceed to use the ReLU activation function [48]:

$$\vartheta_h^{(\ell)} = max(\chi_h^{(\ell)}, 0). \quad (12)$$

The $max(\cdot,\cdot)$ operation picks up the largest element of its arguments and passes them to the next layer, this operation is applied to each element in the feature maps. The subsample layer, in which the max-pool operation is performed [49], follows this activation function, here the large value in a group of consecutive elements of the input vectors are chosen this is a simplification applied only to vectors instead of matrices, thus obtaining the output of the convolutional block as:

$$\varphi_h^{(\ell)} = \text{max-pool}(\vartheta_h^{(\ell-1)}, ss^{(\ell)}), \quad (13)$$

ss indicates the amount of elements to be considered in the max-pool operation, this value can vary in each convolutional block. These steps are repeated for the amount of convolutional blocks that are occupied within the structure.

After the convolutional block is complete, the output of CNN is generate by:

$$\hat{\beta} = V\varrho, \quad (14)$$

where V are the synaptic weights in the output layer and ϱ is the concatenated vector of the output of the last convolutional block defined as:

$$\varrho = [\varphi_1^n \;\; \varphi_1^n \;\; \cdots \;\; \varphi_1^n]. \quad (15)$$

To train the CNN, the back propagation algorithm is employed and modified in order to adapt it to the convolutional blocks.

The weights V in the final layer of CNN are fully- connected as in a MLP. The cost function to be minimized is the square error defined as:

$$J(r) = \frac{1}{2}e^2(r) = \left[\hat{\beta}(r) - \beta(r)\right]^2, \quad (16)$$

where r is each the training iteration. The update law for the output weights is:

$$V_i(r+1) = V_i(r) - \eta\frac{\partial J}{\partial V_i}, \quad (17)$$

with V_i been each element in matrix V. For the sub-sample layer, the update law here its defined as:

$$\frac{\partial J}{\partial \vartheta^{(\ell-1)}} = up\left(\frac{\partial J}{\partial \varphi^{(\ell)}}, ss^{(\ell)}\right), \quad (18)$$

where $up(\cdot,\cdot)$ is the opposite function to the *maxpool*. This operation passes the gradient to the position of the elements that in the feedforward had the largest contribution in this layer, i.e. the elements chosen by the operation *maxpool* so the gradient is propagated through the CNN.

Then, the reverse operation to ReLU function is used, where its derivative $f'(x)$ needs to be defined first as:

$$f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise.} \end{cases}$$

For the convolutional layers, there are two operations, one is to update the filters, the other one is to propagate the gradient to the previous layer. The update law for the filters is given by;

$$\frac{\partial J}{\partial \kappa_{h,a}^{(\ell)}} = \sum_{i=0}^{N-fl} \frac{\partial J}{\partial \vartheta_i^{(\ell)}} \frac{\partial \vartheta_i^{(\ell)}}{\partial \chi_i^{(l)}} \frac{\partial \chi_i^{(\ell)}}{\partial \kappa_i^{(\ell)}}, \quad (19)$$

Algorithm 1 Estimation of $\hat{\beta}_\theta$ and $\hat{\beta}_\psi$

1: Acquire the training (DTR) and testing (DTE) data-sets from the system in open loop, with $DTR \neq DTE$
2: Choose the hyper-parameters: number of layers ℓ, training epochs r, filters κ and learning rate η
3: Initialization the weights V in [0, 1] and the filters κ in $[-1, 1]$
4: Training stage:
5: **for** r times **do**
6: Forward CNN
7: Backward CNN
8: **end for**
9: Testing stage:
10: Forward CNN
11: Calculate the MSE

return Return the hyper-parameters and weights matrix used to Control simulation

where

$$\frac{\partial \vartheta_i^{(\ell)}}{\partial \chi_i^{(l)}} = f'\left(\chi_{h,i}^{(l)}\right). \tag{20}$$

So, (19) can be written as:

$$\frac{\partial J}{\partial \kappa_{h,a}^{(\ell)}} = \sum_{i=0}^{N-fl} \left(\delta_{h,i}^{(\ell)}\right) \varphi_{h,i+a}^{(\ell-1)}, \tag{21}$$

with

$$\delta_{h,i}^{(l)} = \frac{\partial J}{\partial \varphi_{h,i}^{(\ell)}} f'\left(\chi_{h,i}^{(\ell)}\right). \tag{22}$$

To obtain the gradient of the previous layers, we have:

$$\frac{\partial J}{\partial \varphi_i^{(\ell-1)}} = \sum_{a=0}^{fl-1} \frac{\partial J}{\partial \chi_{i-a}^{(\ell)}} \kappa_a^{(\ell)}, \tag{23}$$

which in terms of a convolutional operation this previous equation is:

$$\frac{\partial J}{\partial \varphi_h^{(\ell-1)}} = \delta_h^{(\ell)} \odot rot180\left(\kappa_h^{(\ell)}\right), \tag{24}$$

where $rot180(\cdot)$ is equivalent to performing a convolution of $\delta_h^{(l)}$ with the filter $K_h^{(l)}$ starting with the last element in the vector and finishing woth the first one °, $\odot$ is the convolutional operation.

The **Algorithm 1** explains the methodology to estimate the frictions terms ($\hat{\beta}_\theta$, $\hat{\beta}_\psi$). Even with ANN architectures like the one proposed in this work, the accuracy of the CNN estimation depends on the hyper-parameters, however, there is no deterministic way to choose them. Therefore, they are chosen randomly and based on trial and error, the best possible ones are selected within a given set of tests.

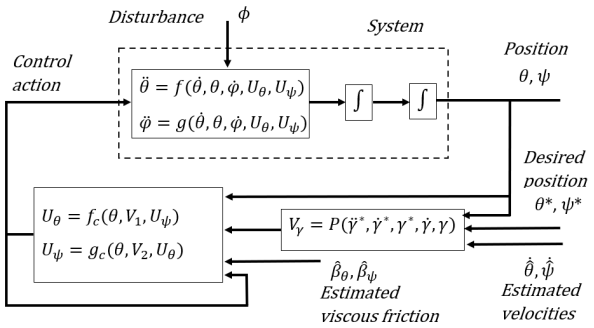


FIGURE 4. Proposed control scheme, feedforward GPI control with variables estimated by the CNN.

III. CONTROL SCHEME

The control objective is to track the trajectory in θ and ψ despite the presence of disturbances in the TR. For this, a control strategy is proposed that combines a feedback linearization with a GPI controller (see Fig. 4), this is made up of 2 control actions U_θ and U_ψ in charge of compensating the dynamics of the system (6) (with $\phi = 0$) and likewise has auxiliary control actions V_θ and V_ψ dedicated to compensating endogenous and exogenous disturbances. Thus, the system approximates a form $\ddot{\gamma} = V_i + \xi$, where ξ are the endogenous and exogenous disturbances and $\ddot{\gamma}$ the acceleration of the coordinate [3], [15].

A. STRATEGY CONTROL

From the above, based on (6) (with $\phi = 0$), the following feedforward controllers are proposed for the θ and ϕ angles:

$$\begin{aligned} U_\theta &= \Big[V_\theta(J_{eq\_p} + m_h L_{cm}^2) - k_{py} U_\psi + \hat{\beta}_\theta \dot{\hat{\theta}} \\ &\quad + m_h L_{cm}^2 s\theta c\theta \dot{\hat{\psi}}^2 + m_h g L_{cm} c\theta\Big] \frac{1}{k_{pp}}; \\ U_\psi &= \Big[V_\psi(J_{eq\_y} + m_h(c\theta L_{cm})^2) + \hat{\beta}_\psi \dot{\hat{\psi}} \\ &\quad - 2m_h L_{cm}^2 \dot{\hat{\theta}} s\theta c\theta \dot{\hat{\psi}} - k_{yp} U_\theta)\Big] \frac{1}{k_{yy}}, \end{aligned} \tag{25}$$

where V_θ and V_ψ are auxiliary controls, defined as [3], [15]:

$$\begin{aligned} V_\gamma &= \dddot{\gamma}^* - K_{3\gamma} \dot{\hat{e}}_\gamma(t) - K_{2\gamma} e_\gamma - K_{1\gamma} \int_{\tau_1=0}^{t} e_\gamma(\tau_1) d\tau_1 \\ &\quad - K_{0\gamma} \int_{\tau_1=0}^{t} \int_{\tau_2=0}^{\tau_1} e_\gamma(\tau_2) d\tau_2 d\tau_1, \end{aligned} \tag{26}$$

with $\gamma = \theta, \psi$; $e_\gamma(t) = \gamma(t) - \gamma^*(t)$; $\dot{\hat{e}}_\gamma(t) = \dot{\hat{\gamma}}(t) - \dot{\gamma}^*$; γ^*, $\dot{\gamma}^*$ and $\ddot{\gamma}^*$ are the desired position, velocity, and accelerations, respectively. $K_{\sigma\gamma}$ are the gains of the controllers, with $\sigma = 0, 1, 2, \ldots, 4$. $\dot{\hat{\gamma}}$ corresponds to the estimates of the variables $\dot{\theta}$ and $\dot{\psi}$, respectively, which are obtained using the CNN described in Section 3.

From the above, substituting (25) into (8) and solving the equation:

$$0 = -\ddot{\theta} + V_\theta + \frac{-k_{py} + k_{yy} s\phi + k_{py} c\phi}{J_{eq\_p} + m_h L_{cm}^2} U_\psi; \tag{27}$$

TABLE 1. Poles and gains.

γ	Poles	$k_{3,\gamma}$	$k_{2,\gamma}$	$k_{1,\gamma}$	$k_{0,\gamma}$
θ	$s_0 = 0.78;\ s_1 = 0.56;$ $s_{2,3} = -5.32 \pm 2.052j$	12	47.39	48.69	14.44
ψ	$s_0 = -0.5;\ s_1 = -0.78;$ $s_{1,2} = -3.2 \pm 0.204$	7.69	18.93	15.79	4.06

$$0 = -\ddot{\psi} + \left(c\phi + \frac{k_{py}}{k_{yy}} s\phi\right) V_\psi + \left(1 - c\phi - \frac{k_{py}}{k_{yy}} s\phi\right) \frac{k_{yp} U_\theta - B_\psi \dot{\psi} + 2 m_h L_{cm}^2 \dot{\theta} s\theta c\theta \dot{\psi}}{J_{eq,y} + m_h c^2\theta L_{cm}^2}, \tag{28}$$

then, substituting (26) for (27) and (28):

$$0 = \ddot{e}_\theta - K_{3\theta}\dot{\hat{e}}_\theta(t) - K_{2\theta} e_\theta - K_{1\theta} \int_{\tau_1=0}^{t} e_\theta(\tau_1) d\tau_1 - K_{0\theta} \int_{\tau_1=0}^{t} \int_{\tau_2=0}^{\tau_1} e_\theta(\tau_2) d\tau_2 d\tau_1 + \frac{-k_{py} + k_{yy} s\phi + k_{py} c\phi}{J_{eq\_p} + m_h L_{cm}^2} U_\psi; \tag{29}$$

$$0 = \left(c\phi + \frac{k_{py}}{k_{yy}} s\phi\right) \left[\ddot{\psi}^* - \frac{\ddot{\psi}}{c\phi + \frac{k_{py}}{k_{yy}} s\phi} - K_{3\psi}\dot{\hat{e}}_\psi(t) - K_{2\psi} e_\psi - K_{1\psi} \int_{\tau_1=0}^{t} e_\psi(\tau_1) d\tau_1 - K_{0\psi} \int_{\tau_1=0}^{t} \int_{\tau_2=0}^{\tau_1} e_\psi(\tau_2) d\tau_2 d\tau_1 \right] + \left(1 - c\phi - \frac{k_{py}}{k_{yy}} s\phi\right) \frac{k_{yp} U_\theta - B_\psi \dot{\psi} + 2 m_h L_{cm}^2 \dot{\theta} s\theta c\theta \dot{\psi}}{J_{eq,y} + m_h c^2\theta L_{cm}^2}. \tag{30}$$

From the above, performing the Routh-Hurwitz analysis, the controller is capable of compensating for step-type disturbances, for tracking step and ramp trajectories in yaw and parabola type for the pitch axis. So, for when ϕ is zero and by differentiating equations A and B twice, obtain

$$0 = e_\theta^{(4)} + K_{3\theta} e_\theta^{(3)} + K_{2\theta} \ddot{e}_\theta + K_{1\theta} \dot{e}_\theta + K_{0\theta} e_\theta; \tag{31}$$

$$0 = e_\psi^{(4)} + K_{3\psi} e_\psi^{(3)} + K_{2\psi} \ddot{e}_\psi + K_{1\psi} \dot{e}_\psi + K_{0\psi} e_\psi. \tag{32}$$

B. TUNING

For GPI controller tuning, the resulting dynamics are expressed in terms of the tracking error, obtaining the following characteristic polynomial [15], [50]:

$$S^4 + K_{3\gamma} S^3 + K_{2\gamma} S^2 + K_{1\gamma} S + K_{0\gamma} = 0, \tag{33}$$

the coefficients $k_{\sigma\gamma}$ are selected so that the characteristic polynomial has its roots on the left side of the complex plane (Hurwitz) [50], [51]. The gains of the GPI controller for θ and ψ are shown in the Table 1:

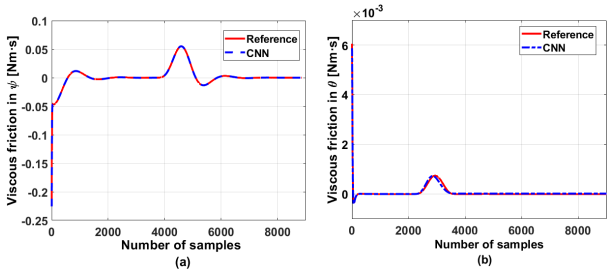


FIGURE 5. Estimation of the terms (a) $\hat{\beta}_\Psi$ and (b) $\hat{\beta}_\theta$.

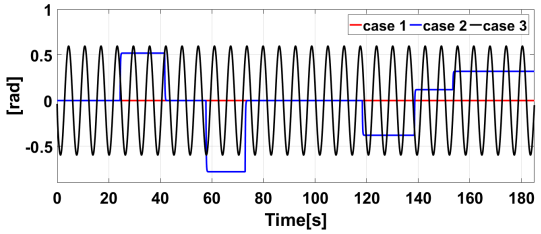


FIGURE 6. Disturbances.

TABLE 2. MSE obtained with CNN, DN and MLP.

Model	MSE for θ	MSE for ψ
CNN	4.07×10^{-4}	0.0011
DNN	3.48×10^{-3}	0.075
MLP	2.75×10^{-2}	0.14

TABLE 3. Bezier polynomial parameters.

γ	γ_i^*	γ_f^*	t_i	t_f
θ	$-0.7rad$	$0.2rad$	$45s$	$75s$
ψ	$-0.3rad$	$0.3rad$	$75s$	$105s$

IV. NUMERICAL SIMULATIONS

In this section the results of this work are described. First, the behaviour of the CNN for estimating the friction and velocity present in each coordinate are shown. Subsequently, we proceed to incorporate the CNN into the control scheme, showing the numerical results.

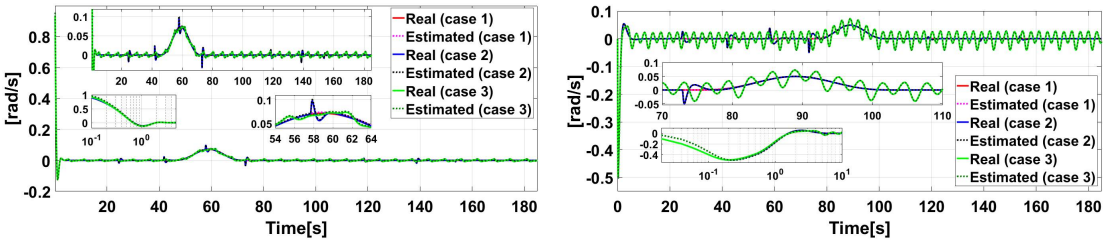
A. CNN RESULTS

Simulations for the estimation of friction terms are shown next. Two data set were used, consisting of 9250 samples of position, torque and friction. One data-set was used for training, whereas the other was used for testing. These data-set were obtained from the system in open loop and this process was made off-line. The matrix weights of the final iteration are used in control stage. For each CNN, the structure has two convolutional blocks, which implies that are 2 convolutional layers and 2 subsample layers, i.e. $n = 4$. The initialization of hyper-parameters is made within the range of $[-1, 1]$.

With respect to $\hat{\beta}_\theta$, the CNN as the regression order $r_{\theta_1} = r_{\theta_2} = 4$, there are $h = 10$ filters in each convolutional layer, the subsample length are $s^{(2)} = s^{(4)} = 2$. Estimation

TABLE 4. 2DOFH parameters.

Parameters	Variable	Magnitude	Units
Distance between the RP and the axis of rotation	rp	0.432	m
Distance between the RC and the axis of rotation	ry	0.233	m
Distance from the center of mass on the axis of rotation	L_{cm}	0.164	m
Helicopter body mass in motion at *pitch*	m_h	1.03	kg
Moment of inertia of the rotating body in *pitch*	$J_{eq,p}$	0.09359	kgm^2
Moment of inertia of the rotating body in *yaw*	$J_{eq,y}$	0.0947	kgm^2
RP force constant over *pitch*	k_{pp}	0.7901	$Nm/\%pwm$
RP force constant over *yaw*	k_{yp}	0.1426	$Nm/\%pwm$
RC force constant over *yaw*	k_{yy}	-0.5078	$Nm/\%pwm$
RC force constant over *pitch*	k_{py}	0.0329	$Nm/\%pwm$



(a) Pitch speed estimation.

(b) Yaw speed estimation.

FIGURE 7. Speed estimation by CNN.

results are presented in Fig. 5a, the Mean Square Error (MSE) obtained is 4.07×10^{-4}.

For $\hat{\beta}_\Psi$, the regression order are $r_{\theta_1} = 18$ and $r_{\theta_2} = 10$, 18 filters were used in each convolutional layer and subsample length are the same as before, $s^{(2)} = s^{(4)} = 2$. Simulation results are shown in Fig. 5b, obtaining an MSE of 0.0011.

The results obtained were compared with a MultiLayer Perceptron (MLP) and a modern Deep-Neural Network (DNN) structure [52]. It should be noted that CNN presented a lower MSE compared to MLP and DNN (see Table 2).

The DNN used for comparison contains 5 hidden layers and 80 entries for *theta*, 150 entries for *psi* and approximately 200 training epochs were needed to obtain acceptable results. The synaptic weight matrices were randomly initialized. Similar conditions were used for MLP, highlighting the better performance of the proposed CNN.

Remark 1: The coefficients obtained in Table 2 shall be interpreted alongisde with the simulation conditions and their subsequent applicability in a physical environment for experimental tests, this means that the application of the DN and MLP architectures will hardly be used since their precision. It will be subject to specific conditions in the selection of a robust digital control system that guarantees the correct estimation of the parameters necessary for the control of the helicopter. Therefore, the CNN proposed by having more relaxed characteristics such as the number of inptus, will facilitate its application in a control system.

B. TRAJECTORY TRACKING

Several numerical simulations were made for the validation of the control scheme described in (25). The control objective is to follow Bezier polynomial (γ^*) trajectories in both θ and ψ. Table 3 shows the parameters used for the reference trajectories, (γ_i^*) and (γ_f^*) correspond to the initial and final position, respectively; while (t_i) and (t_f) indicates the start and end time of the trajectories.

The simulations were carried out considering the following initial conditions: $\theta(0) = 0.92$ *rad* (equivalent to the initial state of the system is at rest), $\psi(0) = 0$ *rad*, $\dot{\theta} = \dot{\psi} = 0$ *rad*/*s*. Table 4 shows the physical parameters of 2DOFH.

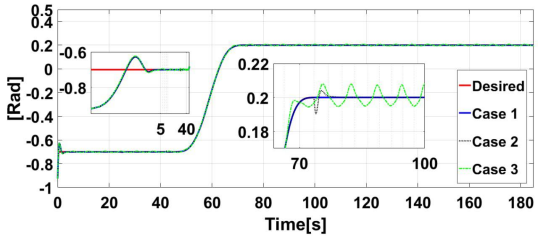
The experiments carried out were based on 3 scenarios: The first considers the RC without disturbances present; whereas the second considers step-type disturbances that occur randomly in the RC; finally, the presence of constant oscillations in ϕ is contemplated (see Fig. 6).

V. COMPARISON AND DISCUSSION

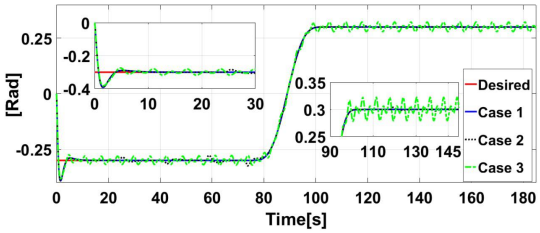
In this section, we evaluate our proposed control scheme against the one reported in [15]. Both schemes use feedforward GPI controllers; however, [15] proposes the use of an integral reconstructor to approximate $\dot{\gamma}$, which is used only within the GPI controller; therefore, an exact linearization of the system is not achieved. In contrast, the current proposal contemplates the use of a CNN to estimate the speed and friction present in each degree of freedom, achieving a better compensation of the 2DOFH dynamics.

Thus, both control schemes were subjected to the disturbances described in Section IV-B. Subsequently, the Incremental Error (IE), Incremental Absolute Error (IAE),and Mean Square Error (MSE) metrics obtained in each experiment were compared.

Fig. 7 shows behavior of the velocities in θ and ψ for cases 1, 2 and 3, which are compared with the estimation

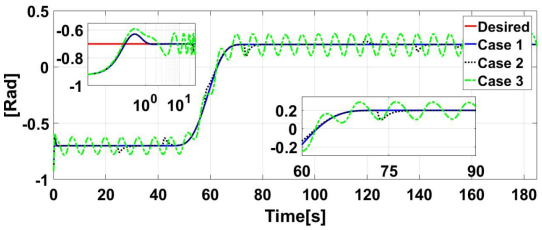


(a) Pitch position.

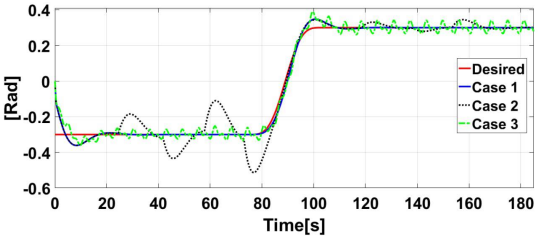


(b) Yaw position.

FIGURE 8. Results when using the variables estimated by the CNN.



(a) Pitch position.



(b) Yaw position.

FIGURE 9. Results when using an integrative reconstructor.

TABLE 5. Incremental error (IE).

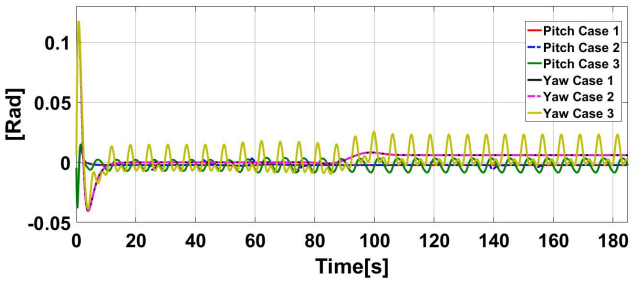
Case	Feedforward with CNN		Feedforward with integral reconstructor	
	Pitch	Yaw	Pitch	Yaw
1	-0.1135	0.2999	-0.1125	0.15
2	-0.1137	0.2999	-4.6972	17.2636
3	-0.1264	0.0821	3.9036	21.8791

TABLE 6. Incremental absolute error (IAE).

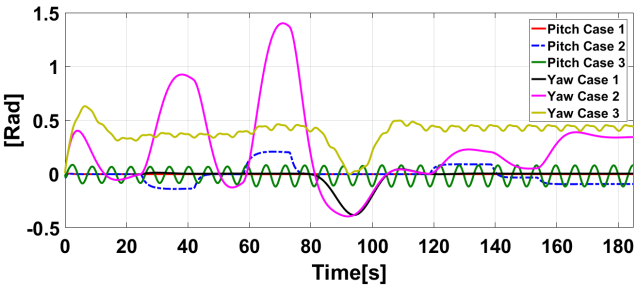
Case	Feedforward with CNN		Feedforward with integral reconstructor	
	Pitch	Yaw	Pitch	Yaw
1	0.1032	0.3263	0.0754	1.8060
2	0.1569	0.5891	1.0461	7.4910
3	0.7405	1.9566	10.7584	4.0783

TABLE 7. Mean of the squared error (MSE).

Case	Feedforward with CNN		Feedforward with integral reconstructor	
	Pitch	Yaw	Pitch	Yaw
1	$44.89_{x\,10^{-6}}$	$0.187_{x10^{-3}}$	$0.0478_{x10^{-3}}$	$0.5977_{x10^{-3}}$
2	$45.94_{x10^{-6}}$	$0.2\ _{x10^{-3}}$	$0.2653_{x10^{-3}}$	$4.27_{x10^{-3}}$
3	$62.59_{x10^{-6}}$	$0.31_{x10^{-3}}$	$4.212_{x10^{-3}}$	$1.075_{x10^{-3}}$



(a) IE of the feedforward controller combined with the CNN.



(b) IE of the feedforward controller combined with an integrative reconstructor.

FIGURE 10. IE for θ and ψ.

given by CNN. The CNN takes approximately 0.5s to converge to the real value of the velocity (even in case 3). However, it is observed that the network has a lower performance in the presence of aggressive disturbances. Despite the above, the estimation is quite good and favors the performance of the control scheme.

Figs. 8 and 9 show the results obtained for the trajectory tracking in θ and ψ, respectively. The results correspond to the cases defined in Section IV-B. Fig. 8 shows the system behaviour before a feedforward GPI controller in combination with the estimation of speed and friction by means of a CNN; for its part, Fig. 9 shows the behavior of the helicopter when using the scheme described in [15]. It is observed that the proposed controller better approximates the desired trajectories. We must highlight the considerable improvement in the angle θ; however, it seems that the angle ψ (case 3) has a similar behavior in both schemes, which is due to the direct incidence of RC disturbances in ψ, although the oscillations are of smaller amplitude when CNN is used.

The results of the IE, IAE and MSE metrics are shown in Tables 5, 6 and 7, respectively. The indices were applied in both control schemes to quantitatively evaluate their performance. It should be noted that for case 1, the IE and the IAE are lower when using an integral reconstructor; however, in the presence of disturbances, the feedforward scheme in conjunction with CNN performs better.

Fig. 10 shows the behavior of IE, which presents oscillations closer to zero with the scheme feedforward with CNN compared to the feedforward with an integral reconstructor.

VI. CONCLUSION

The simulation results showed that the feedforward controller in combination with the CNN provides higher performance compared to the scheme reported in (25). It should be noted that both schemes retain the active disturbance rejection properties of a GPI controller; however, the use of CNN reduces the need for integral reconstructors and allows the system to be approximated to a more linear representation.

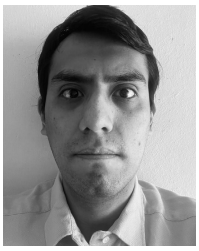
The proposed CNN showed a good estimation of the parameters even when the system was subjected to disturbances in the TR. Despite the deviation that usually occurs in the estimation when subjecting the system to disturbances, the friction and speed obtained allowed a good compensation of the 2DOFH dynamics. Hence, the robustness of the proposed scheme against disturbances in the TR is demonstrated. Also, we must emphasize that CNN does not update its weights in real time, which reduces the computational requirements for its future implementation. Finally, it is important to mention that the use of CNN to estimate velocities and frictions consumes a higher computational cost compared to the integral reconstructor.

REFERENCES

[1] G. Lu, "Aggressive attitude control of unmanned rotor helicopters using a robust controller," *J. Intell. Robot. Syst.*, vol. 80, no. 1, pp. 165–180, Oct. 2015.
[2] H. T. Berie and I. Burud, "Application of unmanned aerial vehicles in earth resources monitoring: Focus on evaluating potentials for forest monitoring in Ethiopia," *Eur. J. Remote Sens.*, vol. 51, no. 1, pp. 326–335, Jan. 2018.
[3] Y. L. Hernández, O. G. Frías, N. Lozada-Castillo, and A. L. Juárez, "Control algorithm for taking off and landing manoeuvres of quadrotors in open navigation environments," *Int. J. Control, Autom. Syst.*, vol. 17, no. 9, pp. 2331–2342, Sep. 2019.
[4] W. Johnson, *Helicopter Theory*. Chelmsford, MA, USA: Courier Corporation, 2012.
[5] F. L. M. D. Santos, B. Peeters, H. Van der Auweraer, L. C. S. Góes, and W. Desmet, "Vibration-based damage detection for a composite helicopter main rotor blade," *Case Stud. Mech. Syst. Signal Process.*, vol. 3, pp. 22–27, Jun. 2016.
[6] Z. Li, H. H. Liu, B. Zhu, H. Gao, and O. Kaynak, "Nonlinear robust attitude tracking control of a table-mount experimental helicopter using output feedback," *IEEE Trans. Ind. Electron.*, vol. 62, no. 9, pp. 5665–5676, Sep. 2015.
[7] M. He, J. He, and S. Scherer, "Model-based real-time robust controller for a small helicopter," *Mech. Syst. Signal Process.*, vol. 146, Jan. 2021, Art. no. 107022.
[8] C.-T. Lee and C.-C. Tsai, "Improved nonlinear trajectory tracking using RBFNN for a robotic helicopter," *Int. J. Robust Nonlinear Control*, vol. 20, no. 10, pp. 1079–1096, Jul. 2010.
[9] L. M. Belmonte, R. Morales, A. Fernández-Caballero, and J. A. Somolinos, "A tandem active disturbance rejection control for a laboratory helicopter with variable-speed rotors," *IEEE Trans. Ind. Electron.*, vol. 63, no. 10, pp. 6395–6406, Oct. 2016.
[10] Z. Chen, D. Wang, Z. Zhen, B. Wang, and J. Fu, "Take-off and landing control for a coaxial ducted fan unmanned helicopter," *Aircr. Eng. Aerosp. Technol.*, vol. 89, no. 6, pp. 764–776, Oct. 2017.
[11] T. M. Fletcher and R. E. Brown, "Main rotor-tail rotor interaction and its implications for helicopter directional control," *J. Amer. Helicopter Soc.*, vol. 53, no. 2, pp. 125–138, Apr. 2008.
[12] R. Rashad, A. El-Badawy, and A. Aboudonia, "Sliding mode disturbance observer-based control of a twin rotor MIMO system," *ISA Trans.*, vol. 69, pp. 166–174, Jul. 2017.
[13] G. G. Neto, F. D. S. Barbosa, and B. A. Angelico, "2-DOF helicopter controlling by pole-placements," in *Proc. 12th IEEE Int. Conf. Ind. Appl. (INDUSCON)*, Nov. 2016, pp. 1–5.
[14] J. M. A. Pena, S. A. R. Paredes, J. S. V. Martinez, and Y. Aguilar-Molina, "A laboratory prototype tandem helicopter with two degrees of freedom," *IEEE Access*, vol. 9, pp. 39618–39625, 2021.
[15] V. G. Sánchez-Meza, Y. Lozano-Hernández, and O. O. Gutiérrez-Frías, "Modeling and control of a two DOF helicopter with tail rotor disturbances," in *Proc. Int. Conf. Mechatronics, Electron. Automot. Eng. (ICMEAE)*, Nov. 2020, pp. 79–84.
[16] Q. Ahmed, A. I. Bhatti, S. Iqbal, and I. H. Kazmi, "2-sliding mode based robust control for 2-DOF helicopter," in *Proc. 11th Int. Workshop Variable Struct. Syst. (VSS)*, Jun. 2010, pp. 481–486.
[17] J. C. R. Murcia and L. F. C. Alfonso, "Modelling, identification and control of a 2 DOF helicopter prototype," in *Proc. IEEE 4th Colombian Conf. Autom. Control (CCAC)*, Oct. 2019, pp. 1–6.
[18] P. Lambert and M. Reyhanoglu, "Observer-based sliding mode control of a 2-DOF helicopter system," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2018, pp. 2596–2600.
[19] J. Gadewadikar, F. Lewis, K. Subbarao, and B. M. Chen, "Structured H-infinity command and control-loop design for unmanned helicopters," *J. Guid., Control, Dyn.*, vol. 31, no. 4, pp. 1093–1102, Jul. 2008.
[20] H. Rojas-Cubides, J. Cortés-Romero, H. Coral-Enriquez, and H. Rojas-Cubides, "Sliding mode control assisted by GPI observers for tracking tasks of a nonlinear multivariable twin-rotor aerodynamical system," *Control Eng. Pract.*, vol. 88, pp. 1–15, Jul. 2019.
[21] J. D. J. Rubio, "Robust feedback linearization for nonlinear processes control," *ISA Trans.*, vol. 74, pp. 155–164, Mar. 2018.
[22] W. Eric and P. Luc, *Identification of Parametric Models: From Experimental*. Berlin, Germany: Springer, 2010.
[23] A. Chovancová, T. Fico, L. Chovanec, and P. Hubinsk, "Mathematical modelling and parameter identification of quadrotor (a survey)," *Proc. Eng.*, vol. 96, pp. 172–181, Jan. 2014.
[24] L. Derafa, T. Madani, and A. Benallegue, "Dynamic modelling and experimental identification of four rotors helicopter parameters," in *Proc. IEEE Int. Conf. Ind. Technol.*, Dec. 2006, pp. 1834–1839.
[25] N. Abas, A. Legowo, Z. Ibrahim, N. Rahim, and A. M. Kassim, "Modeling and system identification using extended Kalman filter for a quadrotor system," *Appl. Mech. Mater.*, vols. 313–314, pp. 976–981, Mar. 2013.
[26] K. S. Hatamleh, O. Ma, and R. Paz, "A UAV model parameter identification method: A simulation study," *Int. J. Inf. Acquisition*, vol. 6, no. 4, pp. 225–238, Dec. 2009.
[27] J.-P. Corriou, "Models and methods for parametric identification," in *Process Control*. London, U.K.: Springer, 2004.
[28] J. A. Lopez and E. Caicedo, "Parametric identification using multilayer perceptron," in *Proc. Int. Conf. Ind. Electron. Control Appl.*, 2005, p. 4.
[29] S. S. Shamsudin and X. Q. Chen, "Identification of an unmanned helicopter system using optimised neural network structure," *Int. J. Intell. Syst. Technol. Appl.*, vol. 17, no. 3, pp. 1–19, 2012.
[30] S. S. Shamsudin and X. Q. Chen, "Recursive Gauss–Newton based training algorithm for neural network modelling of an unmanned rotorcraft dynamics," *Int. J. Model., Identificat. Control*, vol. 17, nos. 1–2, pp. 223–241, 2012.
[31] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10, NJ, USA, 1995, p. 1995.
[32] W. Yu and M. Pacheco, "Impact of random weights on nonlinear system identification using convolutional neural networks," *Inf. Sci.*, vol. 477, pp. 1–14, Mar. 2019.
[33] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. New York, NY, USA: Springer, 2010.
[34] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, vol. 3. New York, NY, USA: Wiley, 2006.

[35] E. V. Kumar, G. S. Raaja, and J. Jerome, "Adaptive PSO for optimal LQR tracking control of 2 DoF laboratory helicopter," *Appl. Soft Comput.*, vol. 41, pp. 77–90, Apr. 2016.
[36] K. J. Åström and P. Eykhoff, "System identification—A survey," *Automatica.*, vol. 7, no. 4, pp. 123–162, 1971.
[37] L. A. C. Cantera, L. Luna, C. Vargas-Jarillo, and R. Garrido, "Parameter estimation of a linear ultrasonic motor using the least squares of orthogonal distances algorithm," in *Proc. 16th Int. Conf. Electr. Eng., Comput. Sci. Autom. Control (CCE)*, Sep. 2019, pp. 1–6.
[38] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, Nov. 2018, Art. no. e00938.
[39] F. Capraro, D. Patino, S. Tosetti, and C. Schugurensky, "Neural network-based irrigation control for precision agriculture," in *Proc. IEEE Int. Conf. Netw., Sens. Control*, Apr. 2008, pp. 357–362.
[40] I. Pavlenko, J. Trojanowska, V. Ivanov, and O. Liaposhchenko, "Scientific and methodological approach for the identification of mathematical models of mechanical systems by using artificial neural networks," in *Proc. Int. Conf. Innov., Eng. Entrepreneurship*. Portugal: Springer, 2018, pp. 299–306.
[41] O. Adeyemi, I. Grove, S. Peets, Y. Domun, and T. Norton, "Dynamic neural network modelling of soil moisture content for predictive irrigation scheduling," *Sensors*, vol. 18, no. 10, p. 3408, Oct. 2018.
[42] Y. Bao, J. M. Velni, A. Basina, and M. Shahbakhti, "Identification of state-space linear parameter-varying models using artificial neural networks," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5286–5291, 2020.
[43] H. Hassanpour, B. Corbett, and P. Mhaskar, "Integrating dynamic neural network models with principal component analysis for adaptive model predictive control," *Chem. Eng. Res. Des.*, vol. 161, pp. 26–37, Sep. 2020.
[44] N. Lanzetti, Y. Z. Lian, A. Cortinovis, L. Dominguez, M. Mercangöz, and C. Jones, "Recurrent neural network based MPC for process industries," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 1005–1010.
[45] X. Yang and X. Zheng, "Adaptive NN backstepping control design for a 3-DOF helicopter: Theory and experiments," *IEEE Trans. Ind. Electron.*, vol. 67, no. 5, pp. 3967–3979, May 2020.
[46] X. Zheng and X. Yang, "Command filter and universal approximator based backstepping control design for strict-feedback nonlinear systems with uncertainty," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 1310–1317, Mar. 2020.
[47] H. Su, W. Qi, C. Yang, J. Sandoval, G. Ferrigno, and E. D. Momi, "Deep neural network approach in robot tool dynamics identification for bilateral teleoperation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2943–2949, Apr. 2020.
[48] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Aug. 2017, pp. 1–6.
[49] C.-Y. Lee, P. Gallagher, and Z. Tu, "Generalizing pooling functions in CNNs: Mixed, gated, and tree," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 863–875, Apr. 2018.
[50] Y. Lozano and O. Gutierrez, "Design and control of a four-rotary-wing aircraft," *IEEE Latin Amer. Trans.*, vol. 14, no. 11, pp. 4433–4438, Nov. 2016.
[51] K. Ogata, *Ingeniería de Control Moderna*, vol. 41. London, U.K.: Pearson, 2003.
[52] I. R. Widiasari and L. E. Nugroho, "Deep learning multilayer perceptron (MLP) for flood prediction model using wireless sensor network based hydrology time series data mining," in *Proc. Int. Conf. Innov. Creative Inf. Technol. (ICITech)*, Nov. 2017, pp. 1–5.

MARIO C. MAYA-RODRIGUEZ received the B.S. degree in control and automation engineering from the Instituto Politécnico Nacional (IPN), Mexico City, Mexico, in 2015, and the M.Sc. degree in automatic control from the Center for Research and Advanced Studies (CINVESTAV), IPN, in 2017, where he is currently pursuing the Ph.D. degree in automatic control. His research interests include transfer learning, meta learning, and identification and control of systems.



MARIO A. LÓPEZ-PACHECO received the B.S. degree in mechatronic engineering from the Universidad Tecnológica de la Mixteca, Mexico, in 2014, and the M.S. and Ph.D. degrees in automatic control from the CINVESTAV, Instituto Politécnico Nacional (IPN), Mexico City, in 2017 and 2021, respectively. He currently works as an A-Level Professor at the ESIME, IPN. His research interests include deep learning, and modeling and control of systems.

YAIR LOZANO-HERNÁNDEZ was born in Hidalgo, Mexico. He received the B.S. degree in control and automation from the School of Electrical and Mechanical Engineering, National Polytechnic Institute (ESIME-IPN), in 2013, and the M.Sc. and Ph.D. degrees from the Professional School of Engineering and Advanced Technologies of the National Polytechnic Institute of Mexico (UPIITA-IPN), Mexico, in 2016 and 2019, respectively. He is currently a Researcher at the Interdisciplinary Professional Unit of Engineering Campus Hidalgo (UPIIH), IPN. His research interests include automatic control, control system design for UAV, non-linear system control, and underactuated systems.

VICTOR G. SÁNCHEZ-MEZA received the M.Sc. degree in advanced technology at UPIITA-IPN, Mexico, in 2021, where he is currently pursuing the Ph.D. degree in advanced technology. He was a Control and Automation Engineer with the ESIME-IPN, Mexico, in 2018. He is a CONACYT Fellow. His current research interests include dynamic systems control, UAV control, robotic systems, and mechatronics.

LUIS A. CANTERA-CANTERA received the B.Eng. degree in automation and control engineering from the ESIME-IPN, Mexico, in 2013, the M.Sc. degree in digital systems from the CITEDI-IPN, Mexico, in 2016, and the Ph.D. degree from the Automatic Control Department, CINVESTAV-IPN, Mexico, in 2021. He is currently a Research Professor at the Automation and Control Engineering Department, ESIME-IPN. His research interests include mathematical modeling, parameter estimation, and nonlinear control theory.

RENÉ TOLENTINO-ESLAVA received the B.Eng. in mechanical engineering from the ESIME Azcapotzalco, IPN, in 2001, and the M.Sc. degree in mechanical engineering from the ESIME Zacatenco, IPN, in 2004. Since 2002, he has been working at the Automation and Control Engineering Department, ESIME, where he is currently a Professor. His research interests include fluid flow, flow measurement, and energy saving in turbomachinery.