International Journal of Electrical and Computer Engineering (IJECE)
Vol. 12, No. 4, August 2022, pp. 3607~3619
ISSN: 2088-8708, DOI: 10.11591/ijece.v12i4.pp3607-3619

Adaptive proportional integral derivative deep feedforward network for quadrotor trajectory-tracking flight control

El Ayachi Chater[1], Halima Housny[2], Hassan El Fadil[2]
[1]LASTIMI Laboratory, Higher School of Technology, Mohammed V University, Rabat, Morocco
[2]ISA Laboratory, National School of Applied Sciences, Ibn Tofail University, Kenitra, Morocco

Article Info

Article history:

Received Apr 29, 2021
Revised Mar 21, 2022
Accepted Apr 2, 2022

Keywords:

Adaptive proportional integral derivative
Deep neural network
Feedforward network
Multidimensional particle swarm optimization
Quadrotor system

ABSTRACT

When the controlled system is subject to parameter variations and external disturbances, a fixed-parameter proportional integral derivative (PID) controller cannot ensure its stabilization. In this case, its control requires online parameter adjustment. Specifically, as the quadrotor is a multi-input multi-output, nonlinear, and underactuated system, robust control is necessary to ensure efficient trajectory tracking flights. In this paper, an adaptive proportional integral derivative (APID) controller is proposed to control the quadrotor systems. This APID-based control strategy uses a two hidden layer deep feedforward network (DFN), where the one-step secant algorithm is chosen for initializing the DFN parameters. All the design steps of the proposed adaptive controller are described. The multidimensional particle swarm optimization (PSO) algorithm is used for tuning the DFN parameters. Then, using two simulation efficiency tests, a comparison between the proposed PSO-based APID-DFN, the (non-optimized) APID-DFN, the feedforward network APID, and the fixed-parameter PID controllers proves much efficiency of the proposed adaptive controller. The results illustrate that the proposed PSO-based APID-DFN controller can ensure good quadrotor system stabilization and achieve minimum overshoot and faster convergence speed for all quadrotor motions. Thus, the proposed control strategy could be considered an additional intelligent method-based adaptive control for unmanned aerial vehicles.

Corresponding Author:

El Ayachi Chater
LASTIMI Lab., Higher School of Technology, Mohammed V University
Rabat, Morocco
Email: elayachi.chater@est.um5.ac.ma

1. INTRODUCTION

Over the last years, the research control community has shown an increasing interest in flying vehicles without onboard human pilots known as unmanned aerial vehicles (UAVs). UAVs' civilian applications have increased in diverse fields due to their low cost. Their application ranges from homeland security, disaster relief, and weather forecasting to power line inspection and precision agriculture [1]–[3]. Besides, as in the outdoors environment, the aerial vehicles are exposed to adverse atmospheric conditions, reliable and robust control strategies are necessary.

Many papers deal with the control problem of the multi-rotor UAV systems. Some of these works propose the development of linear controllers like the proportional integral derivative (PID) and linear-quadratic regulators (LQR) [4], [5]. Besides, other works suggest the development of nonlinear methods to ensure UAV system stability. Among the latter, we can list the backstepping control approach [6], [7] and the sliding mode approaches [8], [9]. In addition, intelligent control strategies have been vastly used to stabilize the quadrotor system, such as fuzzy logic control [10]–[12], artificial neural network (ANN) [13], and adaptive fuzzy inference system [14], [15]. Specifically, due to their simplicity and high reliability, the proportional integral derivative (PID) controllers have been widely used in industrial systems. However, the conventional PID controllers are not always satisfying in nonlinear systems such as the highly coupled quadrotor system.

Several research works are still proposed for stabilizing and controlling the quadrotor system flight. For instance, a real-time experimental test is implemented to evaluate the PID controller for the quadrotor system. First, the PID controller parameters were selected to have the desired energy consumption [16]. Then, a PID flight controller for a small aerial vehicle was presented in [17], where the proposed algorithm was deployed using Bluetooth® low energy connection via a personal area network (PAN). Besides, a design and implementation solution that utilizes low-cost components with a PID control for a quadrotor system was proposed with acceptable experimental test results [4].

Indeed, the most crucial step for the PID controller design is finding appropriate gains that permit the stabilization of the nonlinear system despite system parameter variation and external disturbances. Thus, several methods are utilized for tuning the PID control gains. For instance, the PID parameters could be determined using the classical Zeigler-Nichols (ZN) method with tests for attitude stabilization of a quadrotor system. However, despite the easy implementation of the ZN method, it could generally generate an unacceptable overshoot. Therefore, optimization techniques are usually used to (offline) tune the PID controller gains according to a fitness function. Among these techniques, particle swarm optimization (PSO) [18], genetic algorithm (GA) [19], and ant colony optimization (ACO) are usually utilized [20]. These techniques are shown to be better than the ZN method. However, they only provide constant control parameters, which could lead to small control quality in the presence of external disturbances. To overcome this drawback, the online tuning of the PID control parameters is more effective. Mainly, combining the PID control strategy with intelligent techniques improves the quadrotor control performances. Thus, several recent published works have incorporated intelligent algorithms to ensure online PID parameter adjustment, such as fuzzy logic control (FLC) [21], [22] and neural networks (NN) [23]–[25].

Artificial neural networks (ANNs) have been widely studied due to their system identification and control design advantages. The deep neural network (DNN) is primarily defined as an exciting area of neural network (NN), which has gained popularity in recent years. The architecture of DNN is based on a multi-layer NN that contains multiple hidden layers. A typical model of DNN is the deep feedforward network (DFN). This self-learning algorithm allows filtering information through multiple hidden layers in forwarding directions using the ANN algorithm. Significantly, the DNN has enabled significant progress in sound and image processing applications, including feature detection, facial recognition, object identifications, computer vision, and text classification [26]–[28]. Besides, potential applications of DNN are numerous in control system engineering [29]. NN and similar approaches such as DNN and DFN can provide better results when used to online tune the PID controller parameters [13], [23]–[25] for controlling the quadrotor system. For instance, an ANN was utilized to adjust the PID parameters [23], where a comparative study of three types of training methods (Bayesian regularization, Levenberg-Marquardt, and scaled conjugate gradient) was performed to minimize the mean square error (MSE) when controlling the roll, pitch, yaw, and altitude of a quadrotor system. Then, as the proposed adaptive ANN-based PID can improve the quadrotor tracking performances, a decentralized PID neural network (PIDNN) control scheme was proposed to stabilize a quadrotor's attitude in the presence of the Dryden model of wind disturbance. Simultaneously, a conventional PID controller was utilized in the outer loop to generate the inner-loop reference path [24]. Then, it was shown that this controller could reject the external disturbances with good stability. An adaptive PID controller was also proposed to stabilize a quadrotor system's attitude and position with unknown variable payloads. It was shown that the neural network could deal with the unknown variable payload by online tuning of the PID control parameters [25]. A comparative analytical method has also shown the advantages of ANNs over a UAV attitude controller's PID conventional control method [13].

Inspired by the above works and considering their result limitations, this paper aims to present a new control strategy based on an adaptive PID deep feedforward network (APIDDFN) for quadrotor UAVs. Specifically, the proposed learning control solution uses: i) deep feedforward network composed of two hidden layers, ii) the one-step secant (OSS) method selects the deep feedforward network's initial weights and bias, iii) a multidimensional particle swarm optimization (PSO) algorithm to adjust the elaborated deep feedforward network (DFN), and iv) a quadrotor (model-based) simulation setup in Dryden wind turbulence and assuming a payload mass change during the vehicle motions. It is worth noticing that the system model is always highly simplified, the relation between the body-fixed and earth-fixed velocities is omitted, and external disturbances are ignored in almost previous works related to controlling the quadrotor UAVs. No simplifications are used in this work.

Our new result may be summarized as follows: using the DFN learning algorithm, a performance evaluation obtained through analyses and a model-based simulation setup shows that combining a PID controller with the DFN algorithm allows designing an efficient adaptive controller that can stabilize the aerial vehicle Euler angles and guarantee a perfect trajectory tracking for arbitrary linear position reference trajectory, despite payload mass variations and external wind disturbances. Thus, the proposed control strategy could be considered an additional controller for nonlinear highly coupled quadrotor UAVs.

This paper is organized as follows: the underactuated quadrotor dynamics, the Dryden wind model, and the APIDDFN control design are described in section 2. Then, section 3 presents the simulation results and discussion. Finally, section 4 concludes the paper.

2. RESEARCH METHOD

In this section, the APIDDFN controller design is presented for the quadrotor system in the presence of the Dryden wind disturbance and payload mass change. For clarity, the quadrotor dynamics and the Dryden wind turbulence models are first presented. Then, the whole control design steps are described in detail.

2.1. Quadrotor dynamic modelling

A quadrotor is equipped with four rotors: two rotors (1 and 3) rotate in a counterclockwise direction, while the other rotors (2 and 4) rotate in a clockwise direction. This four-rotor actuation can generate six motions: three linear translations (along the x, y, and z axes) and three angular rotations, namely the φ-roll, θ-pitch, and ψ-yaw Euler angles. Thus, the quadrotor is an underactuated highly coupled system with its six degrees of freedom (6DOF) and only four input actuators.

Now, to establish the quadrotor model, two coordinate systems should be considered: an earth-fixed coordinate frame ($F_e = (O, X_e, Y_e, Z_e)$), and a body-fixed coordinate frame ($F_b = (G, X_b, Y_b, Z_b)$), whose origin is the vehicle center of gravity (CoG). For clarity, Figure 1 shows the quadrotor X structure. The quadrotor system model may be summarized as (1) [30]:

$$\begin{cases} \ddot{x} = \left(c_\varphi s_\theta c_\psi + s_\varphi s_\psi\right)\frac{T}{m} + \frac{1}{m} f_{wx} \\ \ddot{y} = \left(c_\varphi s_\theta s_\psi - s_\varphi c_\psi\right)\frac{T}{m} + \frac{1}{m} f_{wy} \\ \ddot{z} = c_\varphi c_\theta \frac{T}{m} - g + \frac{1}{m} f_{wz} \\ \ddot{\varphi} = \frac{\tau_\varphi}{j_x} + \frac{j_y - j_z}{j_x} \dot{\theta}\dot{\psi} \\ \ddot{\theta} = \frac{\tau_\theta}{j_y} + \frac{j_z - j_x}{j_y} \dot{\varphi}\dot{\psi} \\ \ddot{\psi} = \frac{\tau_\psi}{j_z} + \frac{j_x - j_y}{j_z} \dot{\varphi}\dot{\theta} \end{cases} \tag{1}$$

where x, y and z denote the cartesian three-space position, φ, θ, and ψ denote the aerial vehicle orientation, T is the total thrust, m denotes the vehicle mass, g is the gravitational acceleration, j_x, j_y, and j_z denote the moments of inertia along the three axes x, y, and z, respectively, τ_φ, τ_θ, and τ_ψ denote the propeller torques, and f_{wx}, f_{wy}, and f_{wz} denotes the aerodynamic force. For simplicity, we adopted the following notations: $c_\beta := cos\ (\beta)$, and $s_\beta := sin\ (\beta)$, for $\beta \in [\frac{-\pi}{2}, \frac{\pi}{2}]$.

2.2. The Dryden wind turbulence

The turbulence process is stochastic and is expressed by velocity spectra [31]. In this work, we apply the most frequently used model called the Dryden turbulence model [32]. Without loss of generality, we admit that the turbulence field is fixed in time and space. However, we assume using a random, homogenous, and isotropic turbulent wind. The Dryden turbulence model is characterized by its power spectral densities (PSD). These power spectral densities correspond to the longitudinal, lateral, and vertical linear velocity components for the MIL-F-8785C model [31] and are defined by (2):

$$\Phi_x(w) = \frac{2{\sigma_x}^2 L_x}{\pi V} \frac{1}{1+\left(L_x \frac{w}{V}\right)^2}, \quad \Phi_y(w) = \frac{{\sigma_y}^2 L_y}{\pi V} \frac{1+3\left(L_y \frac{w}{V}\right)^2}{\left[1+\left(L_y \frac{w}{V}\right)^2\right]^2}, \quad \Phi_z(w) = \frac{{\sigma_z}^2 L_z}{\pi V} \frac{1+3\left(L_z \frac{w}{V}\right)^2}{\left[1+\left(L_z \frac{w}{V}\right)^2\right]^2} \tag{2}$$

where the turbulence intensities (σ_x, σ_y and σ_z) correspond to the turbulence magnitude in the longitudinal, lateral, and vertical axes, respectively. The turbulence scale length (L_x, L_y and L_z) represent the turbulence field length in the longitudinal, lateral, and vertical axis, respectively. w is the time-frequency, and V is the air velocity.

Implementing the Dryden wind model consists of passing the band-limited Gaussian white noise signal through a shaping filter function to generate an output signal with spectral properties. The turbulence signal must have the same frequency spectrum as the PSD of the Dryden wind. For this reason, we should use the following filters defined by (3) [33]:

$$H_x(s) = \sigma_x \sqrt{\frac{2L_x}{\pi V}} . \frac{1}{1+\frac{L_x}{V}s}, \quad H_y(s) = \sigma_y \sqrt{\frac{L_y}{\pi V}} . \frac{1+\frac{\sqrt{3}L_y}{V}s}{\left(1+\frac{L_y}{V}s\right)^2}, \quad H_z(s) = \sigma_z \sqrt{\frac{L_z}{\pi V}} . \frac{1+\frac{\sqrt{3}L_z}{V}s}{\left(1+\frac{L_z}{V}s\right)^2} \tag{3}$$

Continuous Dryden filter is a low-pass filter, where any frequency higher than the cutoff frequency is eliminated. The filter cutoff frequency is given by the ratio of turbulence scale length to the airspeed. Then, according to [31], at low altitude, the turbulence scale lengths and intensities are defined by the following equation where W_{20} denotes the wind speed at the height of 6 meters, and h denotes the quadrotor altitude.

$$L_z = h, \quad L_x = L_y = \frac{h}{(0.177+0.000823h)^{1.2}}, \sigma_z = 0.1W_{20}, \quad \sigma_x = \sigma_y = \frac{0.1W_{20}}{(0.177+0.000823h)^{0.4}} \tag{4}$$

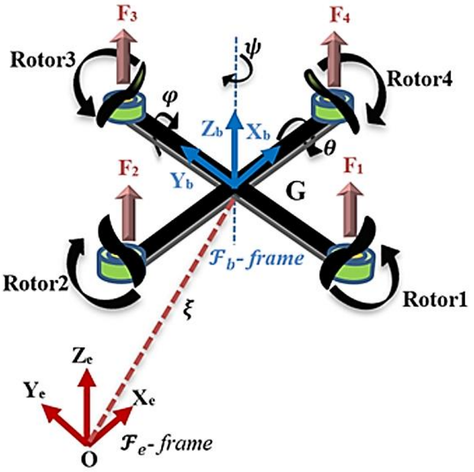


Figure 1. Quadrotor system X-structure with its coordinate frames

2.3. Adaptive PID deep feedforward network design

In this work, the control of the quadrotor system is based on an APIDDFN. This controller is designed by combining the conventional PID control strategy with the DFN algorithm. The DFN algorithm is the most common kind of neural network architecture mapping input to output through hidden nodes using a sequence of layered transformations that connect the neurons of each layer to those of the next layer. For clarity, algorithm 1 gives the main steps of the entire controller design process. Then, a description of this algorithm is given below.

Algorithm 1. PSO-APIDDFN algorithm steps for controlling the quadrotor system

```
1  Collect the quadrotor control system dataset
2  Set the deep feedforward network (DFN) architecture
3  Initialize the DFN parameters (weights and bias)
4  Design the adaptive DFN controller
5  Tune the DFN parameters using the multidimensional PSO algorithm
```

The first step in the APIDDFN design is collecting the training data sets necessary to carry out the learning process. Indeed, we use the PID control strategy to collect the data set required to design the deep feedforward networks (DFNs), namely DFN_x, DFN_y, DFN_z, DFN_φ, DFN_θ and DFN_ψ. The data sets collected from the quadrotor system response are composed of 2-dimensional vector $[e\ U\ ]^T$ where e is the controller error that denotes the difference between the actual and desired output value, and U is the PID controller output for the corresponding system state.

One of the essential features that affect the resultant network in the learning task is the network architecture, as the hidden layer number and neuron number in each layer are essential in the network design. One hidden layer and a few neurons number may be insufficient. However, multi-layer architecture with many neurons may imply a superior computation time and may not be more efficient for solving complex problems.

In the proposed APIDDFN, each DFN has four layers: one input layer, two hidden layers, and one output layer. The input layer has two neurons corresponding to error signal e and the feedback value from the current system output. The output layer has three (two) neurons that consist of the k_p, k_i and k_d (k_p and k_d) PID (PD) gains. The structure of the proposed deep feedforward network is shown in Figure 2. Roughly, each connection has weight and bias values that are represented by matrices of numbers, where $In = [In_1\ In_2]^T$ denotes the network input vector, and $W^k(b^k)$, for $k = \{1,2,3\}$, denotes the matrix value that represents the weight (bias) of the connection between consecutive layers. The mathematical equations of the DFN algorithm can be described as follows:

- Layer 0 (input layer):

$$h^0 = In \tag{5}$$

where In is the network input vector. In this paper, as the input layer comprises two neurons, the input vector dimension is (2x1).

- Layer 1 to layer 2 (hidden layers):

$$h^i = \Phi(b^i + W^i h^{i-1}) \tag{6}$$

For each hidden layer i, $i \in \{1,2\}$, W^i is the layer weight that represents the strength of the connection between the neurons of the hidden layer i and the previous layer $(i-1)$, b^i is the neuron bias added to produce the net input, and Φ is the hidden layer activation function. Noting that, W^i is a $(r\text{x}c)$ matrix and b^i is a $(r\text{x}1)$ vector, where r is the current layer neuron number, and c is the previous layer neuron number.

In this work, W^1 is a (3x2) matrix, b^1 and b^2 are a (3x1) vectors, W^2 is a (3x3) matrix. The activation function applied to each hidden layer is the sigmoid function, which is defined by (7):

$$\Phi(z) = \frac{1}{1+e^{-z}} \tag{7}$$

- Layer 3 (output layer):

$$y_p = \Psi(b^3 + W^3 h^2) \tag{8}$$

where Ψ is the output layer activation function and $y_p = [out_1\ out_2\ out_3]^T$ denotes the network output vector. In this work, the output vector dimension is (3(2)x1), W^3 is an (3(2)x3) matrix, b^3 is an (3(2)x1) vector and the linear activation (i.e., no activation function) is chosen as activation function.

The next step consists of initializing the deep feedforward network weights and biases. The initial training of DFN parameters is the process that permits finding the weights and bias values for all network layers to map the input data to the associated output. This training operation needs running several iterations making minor changes to the DFN parameters until a minimum of a cost function is reached. Many algorithms can be used, such as Levenberg-Marquardt (LM) and the backpropagation algorithms that use Jacobian derivatives, known as the fastest algorithm [34]. Roughly, the backpropagation training function could use the gradient derivatives, such as gradient descent (GD), the Broyden, Fletcher, Goldfarb, and Shanno (BFGS) quasi-newton method, or the one-step secant (OSS) algorithm. In this work, after several tests, we have chosen the one-step secant (OSS) algorithm as it was shown to ensure satisfying results using less memory and computational time compared with the BFGS algorithm [35].

In this paper, the APIDDFN is carried out to stabilize the quadrotor system for trajectory tracking objective. The APIDDFN control method comprises the conventional PID control strategy and the deep feedforward network (DFN) algorithm. Especially for the quadrotor system, six APIDDFNs controllers denoted as $APDDFN_x$, $APDDFN_y$, $APIDDFN_z$, $APIDDFN_\varphi$, $APIDDFN_\theta$ and $APIDDFN_\psi$, are designed. Thus, six DFNs controllers are used to online adjusting the gains of the PID (PD) controllers for z, φ, θ, and ψ (x and y). Figure 3 shows the structure of the $APIDDFN_z$ for the z system-state, where U_1 is the controller output, z is the current system output, and e_z is the difference between the desired trajectory z_{des} and its actual value z. For clarity, a global control system schema is provided in Figure 4.

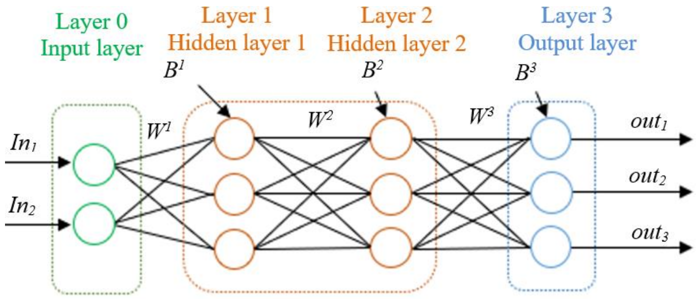


Figure 2. Structure of the designed deep feedforward network

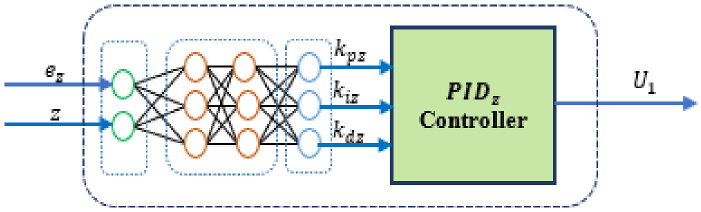


Figure 3. Structure of the APIDDFNz controller designed for *z*-state

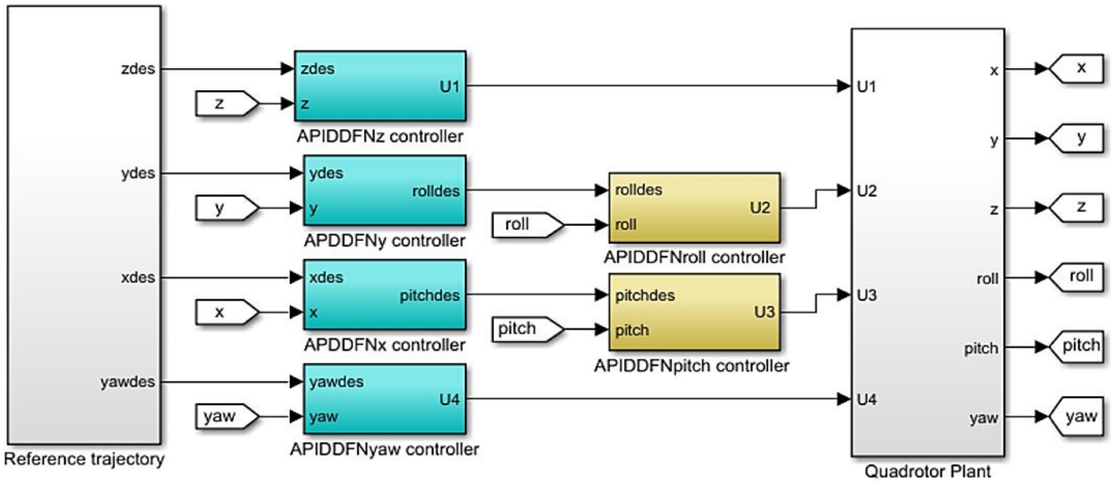


Figure 4. Global control system schema

Besides, once the network architecture has been chosen, the weights W^k and biases b^k should be trained in each layer, for $k \in \{1,2,3\}$. Appropriate values are obtained for these parameters by minimizing a cost function that computes the difference between the predicted output obtained and the actual output value. Then, without a loss of generality, we choose the integral of time-weighted absolute error (ITAE) as a performance criterion in the considered control problem. Thus, letting $e(t)$ denote the error between the target value of the output and the actual output value, the ITAE performance criterion is defined by (9):

$$ITAE = \int_0^\infty t|e(t)|dt \qquad (9)$$

The next step consists of adjusting the weights and bias to better dynamic behavior for the closed-loop quadrotor system until the chosen cost function reaches a minimal value. Thus, we use the multidimensional PSO algorithm to find the 33 (29) DFN parameters' optimum values. Indeed, PSO is a population-based stochastic optimization algorithm that has experienced many enhancements [36]. Thus, PSO has been extensively utilized to solve optimization problems, where it has proven its effectiveness in different research areas [37]. To describe the specific DFN parameter tuning process, algorithm 2 gives the multidimensional PSO algorithm main steps. Each step of algorithm 2 may be described as follows:

First, the PSO algorithm parameters should be selected. Practically, to tune the optimum value of the 33 (29) DFN parameters, the population number n is set to 200, and the maximum iteration is set to 5. However, the right choice of the search space parameter (ss), which defines the search space limits during the tuning process, allows computing the parameters' maximum possible values to be tuned. For this reason, we choose a different search space parameter for each system state. Then, the PSO algorithm is initialized with random particles that correspond to the DFN weights and bias parameters' possible values. In steps 3 and 4, the particles are evaluated to compute the local best fitness $\chi^1(t)$, which is known as the cognitive component, and the global best fitness $\chi^2(t)$, which is known as the social component. In step 5, the PSO algorithm computes the values of actual position $\chi(t)$ and actual velocity $v(t)$ of the particles, which affects the movement of the particles according to (10):

$$\begin{cases} v(t+1) = \gamma\, v(t) + c_1 r_1(\chi^1(t) - \chi(t)) + c_2 r_2(\chi^2(t) - \chi(t)) \\ \chi(t+1) = \chi(t) + v(t+1) \end{cases} \tag{10}$$

where γ is the inertia weight used to adjust the impact of the previous velocity value upon the current one. (c_1, c_2) numbers are the cognitive and social constants, and (r_1, r_2) numbers are arbitrarily chosen in $[0{,}1]$. Afterward, if the maximum iteration number is not yet reached, the algorithm returns to step 2. Otherwise, algorithm 2 provides the optimum values of DFN parameters.

Algorithm 2: Multidimensional PSO algorithm for the DFN

```
    Begin
1       Set the PSO parameters (population number, search space, maximum of iteration ...)
        Loop:
2           Initialize random particles (DFN weights and bias)
3           Evaluate the local best fitness for each particle
4           Evaluate the global best fitness
5           Update the velocity and position of particles
        Until maximum iteration is reached
```

3. RESULTS AND DISCUSSION

The completely dynamic control system setup is implemented using the MATLAB simulation tool to test and show the proposed controller's validity for the quadrotor aerial vehicle. First, the feedforward network (FN) and the DFN structures adjust the PID/PD controller gains for each quadrotor system state. Then, the elaborated final DFN weights and biases are optimized using the PSO algorithm. Besides, a comparison between all the above controllers is provided to show the controller's effectiveness. Finally, to highlight the proposed PSO-based APIDDFN controller validity, an efficiency test is performed. The quadrotor aerial vehicle's payload is assumed to be changing during its motions and subjected to the Dryden turbulence wind.

The quadrotor parameters are chosen similar to those used in [38]. Then, the proposed controller is conducted to stabilize the quadrotor attitude and ensure the following x, y, and z trajectory-tracking:

$$x_{des} = \begin{cases} 1 & t \in [25s \quad 55s] \\ 0 & otherwise \end{cases},\; y_{des} = \begin{cases} 1 & t \in [45s \quad 75s] \\ 0 & otherwise \end{cases},\; z_{des} = \begin{cases} 1 & t \in [5s \quad 95s] \\ 0 & otherwise \end{cases} \tag{11}$$

Then, we aim to show the proposed APIDDFN controller effectiveness compared to three other control methods implemented to stabilize the quadrotor system using the following simulation scenario.

3.1. Quadrotor control by the conventional PID controller

The first control method consists of controlling the quadrotor system by a fixed-parameter PID/PD controller. Table 1 summarizes the PID/PD gains obtained for each system state using the MATLAB PID Tuner. Thus, to design the FN and DFN systems, the training dataset is collected from error and output values when the above-tuned PID controller controls the quadrotor system.

3.2. Quadrotor control by the adaptive PID feedforward network

The second method consists of controlling the quadrotor by the APIDDFN, where six APIDFN controllers are implemented. Each APIDFN controller comprises a PID and a feedforward network (FN) algorithm containing three layers: one input layer, one hidden layer, and one output layer. The one-step secant (OSS) algorithm is used to initialize the FN weight and bias. Table 2 summarizes the 21(17) FN parameters obtained for z, φ, θ, and ψ (x and y) quadrotor system states.

Table 1. PID/PD controller initial gains

	x	y	z	φ	θ	ψ
Kp	0.1	0.1	7.5	0.2339	0.2339	0.00274
Ki	--	--	3.075	0.0929	0.0929	0.0001
Kd	0.2	0.2	5.5	0.08638	0.08638	0.1

Table 2. FN system parameters

	W^1	b^1	W^2	b^2
x	$\begin{bmatrix} -1.7715 & -1.2277 \\ -1.0460 & 2.1203 \\ 1.8570 & 1.5275 \end{bmatrix}$	$\begin{bmatrix} 2.7265 \\ 0.0176 \\ 2.4471 \end{bmatrix}$	$\begin{bmatrix} -0.1914 & -0.0024 & -0.0470 \\ -0.5157 & -0.0110 & -0.0492 \end{bmatrix}$	$\begin{bmatrix} 0.2428 \\ 0.5652 \end{bmatrix}$
y	$\begin{bmatrix} -0.1625 & -2.1594 \\ -0.8279 & 2.1314 \\ 1.0827 & -1.4031 \end{bmatrix}$	$\begin{bmatrix} -2.4831 \\ 0.1042 \\ 2.7171 \end{bmatrix}$	$\begin{bmatrix} 0.0856 & 0.0326 & 0.2804 \\ -0.1094 & -0.0268 & -0.0232 \end{bmatrix}$	$\begin{bmatrix} -0.1916 \\ -0.0954 \end{bmatrix}$
z	$\begin{bmatrix} -1.8357 & -1.3429 \\ -1.4657 & -1.9203 \\ 2.2023 & 0.2091 \end{bmatrix}$	$\begin{bmatrix} 2.6015 \\ -0.0144 \\ 2.6620 \end{bmatrix}$	$\begin{bmatrix} -0.1721 & 0.0042 & -0.3515 \\ 0.4011 & -0.0161 & -0.1328 \\ 0.1481 & -0.0101 & -0.1933 \end{bmatrix}$	$\begin{bmatrix} 0.5038 \\ -0.3017 \\ 0.0980 \end{bmatrix}$
φ	$\begin{bmatrix} 1.9734 & -1.4623 \\ 1.8060 & -1.3916 \\ 2.3032 & -0.9046 \end{bmatrix}$	$\begin{bmatrix} -2.3548 \\ -0.0444 \\ 2.3236 \end{bmatrix}$	$\begin{bmatrix} 0.0567 & 0.0694 & -0.5032 \\ -0.3739 & 0.1919 & -0.9850 \\ -0.8101 & 0.1823 & -0.3795 \end{bmatrix}$	$\begin{bmatrix} 0.5841 \\ 0.5969 \\ -0.4967 \end{bmatrix}$
θ	$\begin{bmatrix} 1.3642 & 1.8727 \\ 0.8303 & 2.2216 \\ -1.5594 & -1.7780 \end{bmatrix}$	$\begin{bmatrix} -2.5868 \\ 0.0926 \\ -2.4978 \end{bmatrix}$	$\begin{bmatrix} -0.8604 & 0.1266 & 0.3287 \\ 0.1905 & 0.0896 & 0.7230 \\ 0.6688 & -0.1112 & -0.8360 \end{bmatrix}$	$\begin{bmatrix} -0.5630 \\ 0.8973 \\ -0.1628 \end{bmatrix}$
ψ	$\begin{bmatrix} 1.8379 & 0.9877 \\ 2.1095 & -0.9542 \\ -1.6207 & 1.7556 \end{bmatrix}$	$\begin{bmatrix} -2.7657 \\ 0.1496 \\ -2.3730 \end{bmatrix}$	$\begin{bmatrix} -0.1470 & -0.0861 & -0.2114 \\ 0.0432 & -0.0209 & -0.0446 \\ -0.0323 & 0.0070 & 0.0184 \end{bmatrix}$	$\begin{bmatrix} -0.2902 \\ 0.1077 \\ 0.0518 \end{bmatrix}$

3.3. Quadrotor control by the adaptive PID deep feedforward network

The third method consists of controlling the quadrotor system using the APIDDFN that comprises one additional hidden layer than the APIDFN. The OSS algorithm is also used as a training method for initializing the DFN parameters. Table 3 provides the 33 (29) DFN parameters for z, φ, θ, and ψ (x and y) quadrotor system state.

Table 3. DFN system parameters

	W^1	b^1	W^2	b^2	W^3	b^3
x	$\begin{bmatrix} -1.2635 & -1.8703 \\ -1.4752 & -1.8558 \\ 1.3386 & 1.7651 \end{bmatrix}$	$\begin{bmatrix} 2.5904 \\ 0.0019 \\ 2.6482 \end{bmatrix}$	$\begin{bmatrix} -1.0528 & -1.0635 & 1.0501 \\ 1.4393 & -0.8603 & -1.0094 \\ -0.2118 & 1.2808 & 1.3327 \end{bmatrix}$	$\begin{bmatrix} 2.1729 \\ -0.0216 \\ -2.1444 \end{bmatrix}$	$\begin{bmatrix} -0.6447 & -0.8081 & -0.9887 \\ -0.3314 & -0.1791 & -0.2768 \end{bmatrix}$	$\begin{bmatrix} 0.2324 \\ 0.2182 \end{bmatrix}$
y	$\begin{bmatrix} 2.2750 & 0.8316 \\ 1.4493 & 1.6726 \\ 0.9982 & -1.9617 \end{bmatrix}$	$\begin{bmatrix} -2.4105 \\ -0.1869 \\ 2.7044 \end{bmatrix}$	$\begin{bmatrix} 1.5614 & -1.2710 & 0.0251 \\ -1.7011 & 0.3893 & 1.2190 \\ -0.1804 & -1.1829 & 0.8886 \end{bmatrix}$	$\begin{bmatrix} -2.2945 \\ 0.1096 \\ 2.8061 \end{bmatrix}$	$\begin{bmatrix} -0.1279 & -0.1177 & -0.3330 \\ 0.8719 & 0.6333 & 0.2293 \end{bmatrix}$	$\begin{bmatrix} 0.3245 \\ -0.0110 \end{bmatrix}$
z	$\begin{bmatrix} -1.9363 & -1.4086 \\ -1.8222 & 1.4957 \\ -0.1388 & 2.1700 \end{bmatrix}$	$\begin{bmatrix} 2.4772 \\ 0.0698 \\ -2.6938 \end{bmatrix}$	$\begin{bmatrix} 0.5245 & 1.2985 & 1.4103 \\ 0.8245 & -0.9529 & -1.5775 \\ 0.6241 & 1.4304 & 0.9536 \end{bmatrix}$	$\begin{bmatrix} -2.0345 \\ 0.1698 \\ 2.2061 \end{bmatrix}$	$\begin{bmatrix} 0.4751 & 0.1462 & -0.1361 \\ 0.3874 & 0.3852 & 0.0725 \\ 0.7497 & 0.3510 & -0.0788 \end{bmatrix}$	$\begin{bmatrix} 0.4341 \\ -0.0046 \\ 0.5572 \end{bmatrix}$
φ	$\begin{bmatrix} -2.3148 & -0.7212 \\ 0.4887 & -2.3122 \\ -2.2385 & -0.9351 \end{bmatrix}$	$\begin{bmatrix} 2.4183 \\ -0.2637 \\ -2.4162 \end{bmatrix}$	$\begin{bmatrix} -0.3337 & -1.5494 & 0.0845 \\ 0.9354 & -1.3129 & -1.2057 \\ -0.0742 & -1.9902 & -0.5102 \end{bmatrix}$	$\begin{bmatrix} 2.4315 \\ 0.0086 \\ 1.9356 \end{bmatrix}$	$\begin{bmatrix} -0.4162 & -0.1092 & 0.5896 \\ -0.2739 & 0.0601 & 0.0690 \\ 0.3692 & -0.3557 & -0.3936 \end{bmatrix}$	$\begin{bmatrix} -0.0923 \\ 0.0684 \\ 0.4138 \end{bmatrix}$
θ	$\begin{bmatrix} -1.0699 & 2.0483 \\ -2.2879 & -0.7388 \\ 1.6393 & -1.7824 \end{bmatrix}$	$\begin{bmatrix} 2.5645 \\ 0.0132 \\ 2.4319 \end{bmatrix}$	$\begin{bmatrix} 1.4605 & 0.0494 & -1.4034 \\ 1.3389 & -0.8714 & -1.0487 \\ -1.0406 & 0.0780 & -1.7284 \end{bmatrix}$	$\begin{bmatrix} -1.9564 \\ 0.0249 \\ -2.0190 \end{bmatrix}$	$\begin{bmatrix} 0.0938 & 0.3084 & 0.0920 \\ 0.5909 & -0.4233 & 0.2745 \\ 0.6534 & -0.3072 & -0.5427 \end{bmatrix}$	$\begin{bmatrix} 0.0978 \\ 0.8989 \\ 0.2059 \end{bmatrix}$
ψ	$\begin{bmatrix} 1.3434 & -1.6871 \\ 0.9313 & 2.2096 \\ 0.9701 & 3.3114 \end{bmatrix}$	$\begin{bmatrix} -2.7498 \\ 0.0267 \\ 1.2526 \end{bmatrix}$	$\begin{bmatrix} 1.9468 & -0.0490 & -0.4695 \\ 0.9199 & 0.0671 & 1.2957 \\ 0.8037 & -0.3096 & -1.9388 \end{bmatrix}$	$\begin{bmatrix} -1.9837 \\ -1.3488 \\ 1.6927 \end{bmatrix}$	$\begin{bmatrix} 0.4094 & -1.3674 & -0.2009 \\ -0.0300 & 0.1989 & 0.0365 \\ 0.0016 & 0.7188 & 0.1032 \end{bmatrix}$	$\begin{bmatrix} -0.7306 \\ 0.1025 \\ 0.6457 \end{bmatrix}$

3.4. Quadrotor control by the PSO-based adaptive PID deep feedforward network

The fourth method consists of controlling the quadrotor system using the APIDDFN controller. In this case, the control parameters are optimized using the PSO algorithm (PSO-APIDDFN). For clarity, Table 4 shows the new 33(29) PSO-tuned DFN parameters for each quadrotor system state.

Table 4. DFN system parameters tuned by the multidimensional PSO algorithm

	W^1	b^1	W^2	b^2	W^3	b^3
x	$\begin{bmatrix} 0.3320 & 4.0000 \\ -0.4583 & 3.4840 \\ 1.2157 & 1.5322 \end{bmatrix}$	$\begin{bmatrix} 3.6509 \\ -0.3025 \\ 2.2516 \end{bmatrix}$	$\begin{bmatrix} 0.2276 & 2.6606 & -2.0000 \\ -0.7656 & -0.7084 & -1.6188 \\ 2.2548 & 0.2497 & 0.7498 \end{bmatrix}$	$\begin{bmatrix} -2.1355 \\ -0.0816 \\ 1.6769 \end{bmatrix}$	$\begin{bmatrix} -0.5444 & 0.0178 & 0.1341 \\ -1.2507 & 0.0271 & -1.0000 \end{bmatrix}$	$\begin{bmatrix} 1.4377 \\ 1.0806 \end{bmatrix}$
y	$\begin{bmatrix} 0.0021 & 1.6944 \\ 2.8716 & -1.4658 \\ -0.0741 & 1.3784 \end{bmatrix}$	$\begin{bmatrix} -3.5567 \\ -0.8757 \\ -2.7979 \end{bmatrix}$	$\begin{bmatrix} 0.5380 & 1.1247 & 2.0000 \\ -2.7043 & 1.3204 & 0.9411 \\ 0.0401 & -0.0941 & 0.3346 \end{bmatrix}$	$\begin{bmatrix} -1.0000 \\ -0.6482 \\ 2.0174 \end{bmatrix}$	$\begin{bmatrix} 0.8007 & 1.3752 & 0.9639 \\ -0.4098 & 0.0989 & 1.0963 \end{bmatrix}$	$\begin{bmatrix} 2.0000 \\ -0.2658 \end{bmatrix}$
z	$\begin{bmatrix} 0 & 1.2998 \\ -0.8544 & 3.0684 \\ 1.4629 & 1.1332 \end{bmatrix}$	$\begin{bmatrix} 2.5406 \\ 1.1023 \\ 4.0000 \end{bmatrix}$	$\begin{bmatrix} -0.7441 & 1.1434 & 1.8354 \\ 0.7360 & -1.0000 & -0.3674 \\ -0.3268 & -2.1546 & -1.4486 \end{bmatrix}$	$\begin{bmatrix} -1.9879 \\ 0.9285 \\ -2.8496 \end{bmatrix}$	$\begin{bmatrix} 1.0000 & 1.4376 & -1.1577 \\ 0.1954 & -0.9893 & -0.5038 \\ -2.0000 & 0.7333 & -1.7944 \end{bmatrix}$	$\begin{bmatrix} 0.8648 \\ 1.4695 \\ -0.2768 \end{bmatrix}$
φ	$\begin{bmatrix} -2.0000 & -1.7835 \\ -1.5054 & -2.4023 \\ -0.4962 & 0.4424 \end{bmatrix}$	$\begin{bmatrix} 4.0000 \\ -0.3487 \\ -3.1380 \end{bmatrix}$	$\begin{bmatrix} 0 & -0.5467 & -0.3974 \\ 0.6310 & -0.6741 & -1.6522 \\ -3.0000 & -0.8071 & 1.0936 \end{bmatrix}$	$\begin{bmatrix} 2.9705 \\ 1.0000 \\ -2.9764 \end{bmatrix}$	$\begin{bmatrix} 0.5489 & -0.1122 & -0.1534 \\ -0.1632 & 0.8813 & 0.2801 \\ 2.0000 & 2.0000 & -1.0000 \end{bmatrix}$	$\begin{bmatrix} 1.1646 \\ 1.4386 \\ 1.0000 \end{bmatrix}$
θ	$\begin{bmatrix} 0 & -0.5770 \\ -0.3557 & 3.6338 \\ -1.2773 & -2.0000 \end{bmatrix}$	$\begin{bmatrix} -2.3539 \\ 0.0436 \\ -3.8555 \end{bmatrix}$	$\begin{bmatrix} 2.0000 & 1.8656 & -0.7121 \\ -0.2626 & -1.3522 & -0.7005 \\ -1.5814 & -1.3641 & -3.0000 \end{bmatrix}$	$\begin{bmatrix} -3.0216 \\ 1.0000 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.9232 & 0.0311 & -0.4625 \\ 2.0000 & -0.3400 & 0.0296 \\ -0.7128 & 2.0000 & 2.0000 \end{bmatrix}$	$\begin{bmatrix} -1.0655 \\ 1.0000 \\ 2.0000 \end{bmatrix}$
ψ	$\begin{bmatrix} 1.9006 & -0.6465 \\ -2.3708 & 1.0607 \\ -1.2146 & 1.5179 \end{bmatrix}$	$\begin{bmatrix} -2.3242 \\ -0.3009 \\ -1.0000 \end{bmatrix}$	$\begin{bmatrix} 1.8107 & 1.3353 & -1.5015 \\ 1.6569 & 1.6522 & -0.9362 \\ -0.7715 & 1.0510 & 0 \end{bmatrix}$	$\begin{bmatrix} -0.9729 \\ -0.0321 \\ 2.8002 \end{bmatrix}$	$\begin{bmatrix} -0.6805 & -1.2258 & 0.4413 \\ 0.9313 & -0.7401 & -0.4986 \\ -0.2053 & 0.1624 & 0.9461 \end{bmatrix}$	$\begin{bmatrix} 1.2186 \\ 0.6912 \\ 2.0000 \end{bmatrix}$

3.5. Controllers' performance comparison

It is worth noticing that the quadrotor linear acceleration involves the total thrust force and the Euler angle states. Then, it is challenging to stabilize this system in an open-loop scheme [39]. For this reason, the quadrotor system should be stabilized using a suitable control strategy. Besides using the PSO algorithm, the proposed control method allows better system performance in the closed-loop.

The time response for linear positions is shown in Figure 5. The performance characteristics comparison such as percent of overshoot (M_p), rise time (T_r), settling time (T_s), and $ITAE$ criterion between these four control strategies is summarized in Table 5. It can be seen from Figure 5 and Table 5 that the APIDDFN is slightly better than the APIDFN. Thus, it is evident that using additional hidden layers has improved the control system's performance. Besides, except for the rise time of the z and x system states, the PSO-APIDDFN controller shows better performances, a faster convergence speed, a minimum overshoot, and a minor $ITAE$ performance criterion compared to the other controllers

Then, Figure 6 shows the φ, θ, and ψ quadrotor attitudes for the PID, APIDFN, APIDDFN, and PSO-APIDDFN controllers. It is shown that the four control methods can ensure the stabilization of the quadrotor attitude. Roughly, φ and θ angle values do not exceed 4 degrees for all these control strategies as shown in Figures 6(a) and 6(b). Besides, the ψ-yaw angle value is almost null, especially for the PSO-APIDDFN controller as shown in Figure 6(c).

Table 5. Control system performance comparison

	x				y				z			
	$M_p(\%)$	$T_r(s)$	$T_s(s)$	$ITAE$	$M_p(\%)$	$T_r(s)$	$T_s(s)$	$ITAE$	$M_p(\%)$	$T_r(s)$	$T_s(s)$	$ITAE$
PID	9.432	1.763	7.376	0.491	9.432	1.763	7.376	0.722	9.975	0.6103	3.613	0.144
APIDFN	8.719	1.770	7.292	0.481	8.845	1.780	7.088	0.699	9.916	0.6102	3.609	0.143
APIDDFN	7.890	**1.702**	6.355	0.395	8.671	1.776	7.277	0.709	9.701	**0.61**	3.574	0.144
PSO-APIDDFN	**1.026**	1.751	**2.968**	**0.075**	**1.134**	**1.729**	**2.929**	**0.111**	**1.529**	0.6923	**2.144**	**0.0523**

3.6. Efficiency test

The simulation tests are also performed to illustrate the above controller efficiency when the quadrotor system is subject to external disturbances. Roughly, two types of disturbances are considered: a quadrotor mass change and external wind. First, Figure 7 shows the case of 20% mass increasing between 10 s and 30 s, and decreasing, between 60 s and 80 s as shown in Figure 7(a), and its effect on the vehicle z-altitude as shown in Figure 7(b). The total mass change caused a slight overshoot in the PSO-APIDDFN (about 5%) than the other controllers (about 16.6%). Thus, the proposed controller can compensate for the mass change by changing the PID/PD gains. Then, Figure 8 illustrates the z-altitude PID gains variation in the case of APIDFN as shown in Figure 8(a), APIDDFN as shown in Figure 8(b), and PSO-APIDDFN as shown in Figure 8(c) controllers when the quadrotor system is subject to a mass change. These subfigures show the adaptive behavior of the three controllers. However, unlike the PID controller gains that remain at the same value in all simulation time, the gains elaborated in the PSO-APIDDFN controller change according to the behavior of the quadrotor system outputs.

Secondly, we simulate the effect of the external wind on the quadrotor system translational motion. This disturbance is conceived using the above Dryden model, where the wind speed is 20 m/s. Figure 9 shows the trajectory tracking of the quadrotor system under the wind. The above results show that the PSO-APIDDFN controller has an excellent wind disturbance-rejection compared to the other control strategies, for x-state as shown in Figure 9(a), y-state as shown in Figure 9(b), and z-state as shown in Figure 9(c).

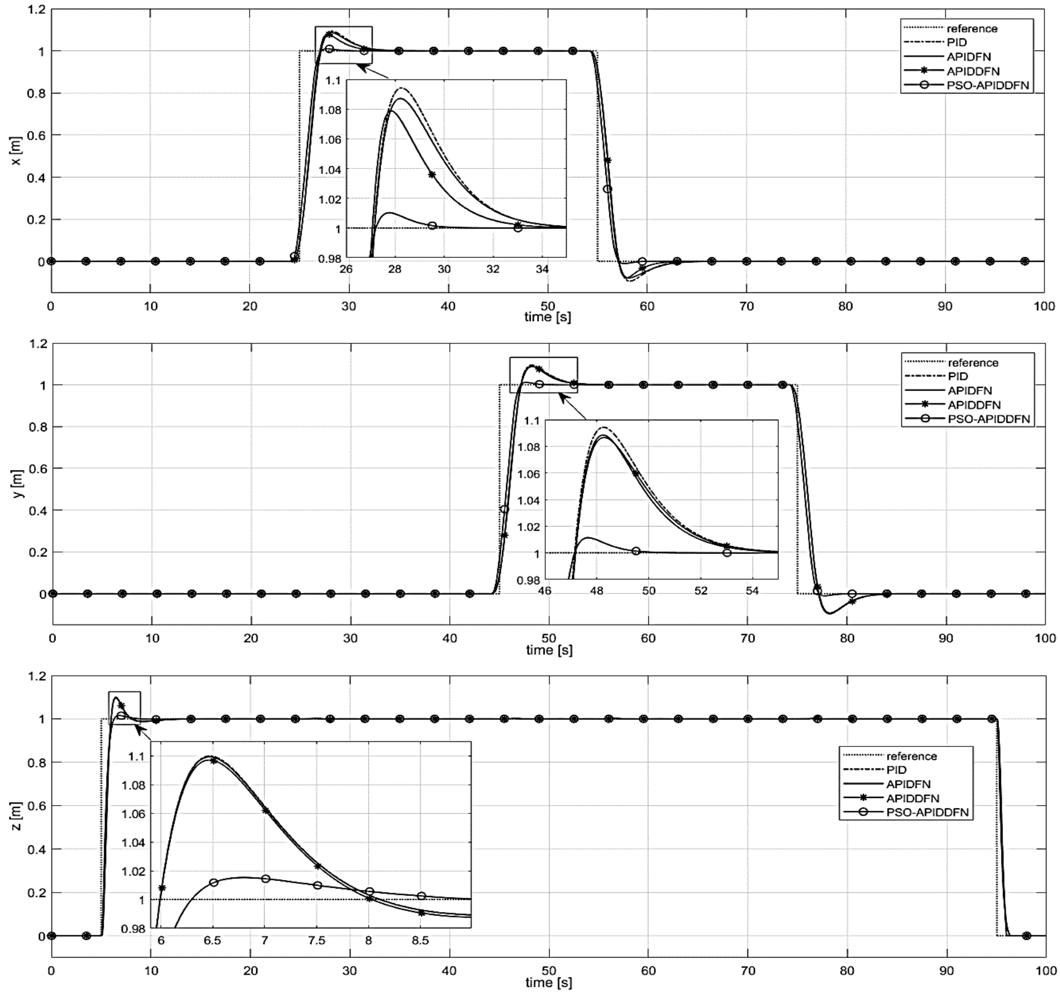
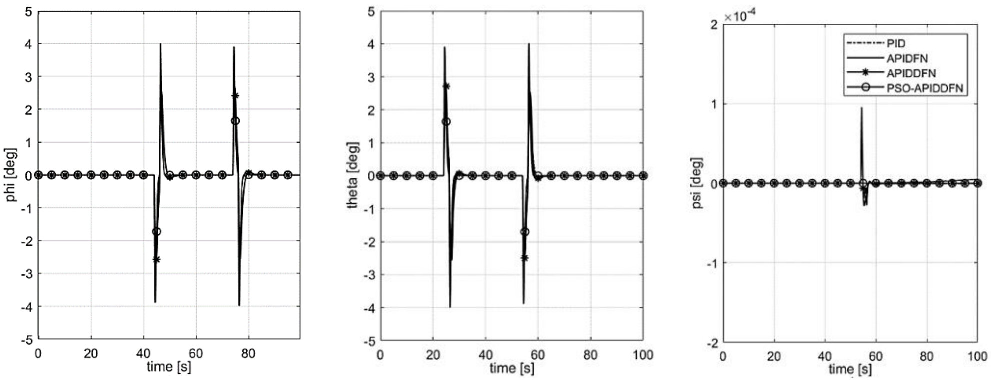
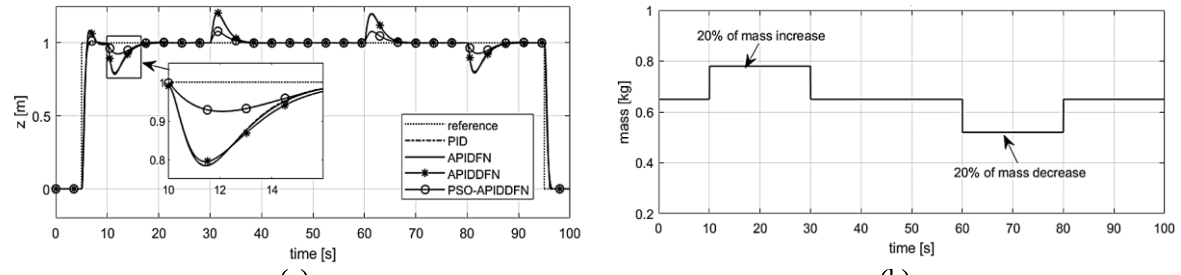


Figure 5. x, y, and z linear positions time responses



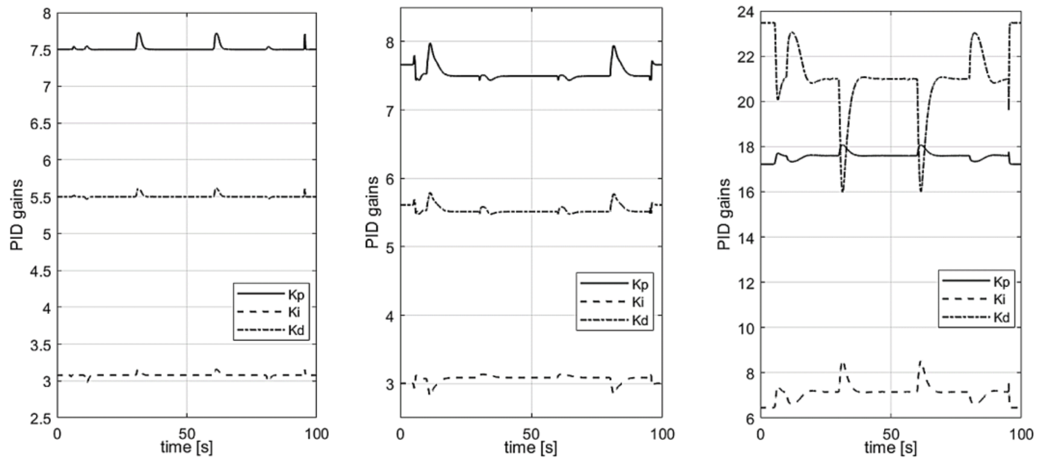
(a) (b) (c)

Figure 6. Euler angles stabilization a) φ-roll, b) θ-pitch, and c) ψ-yaw



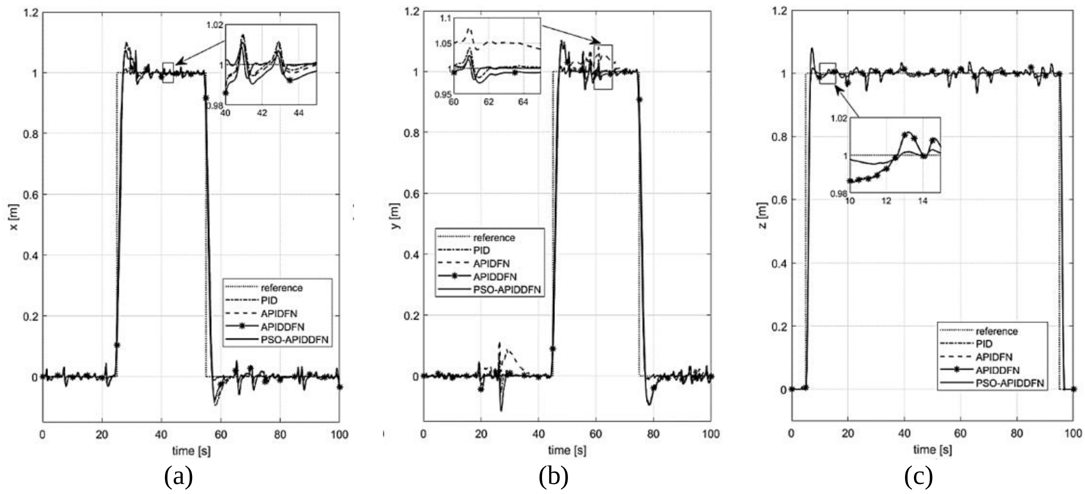
(a) (b)

Figure 7. Altitude response in case of quadrotor mass change (a) altitude response and (b) mass change



(a) (b) (c)

Figure 8. Variation of altitude PID gains in case of (a) APIDFN, (b) APIDDFN, and (c) PSO-APIDFN



(a) (b) (c)

Figure 9. Effect of the wind on the translational motion: (a) x-position, (b) y-position, and (c) z-altitude

4. CONCLUSION

In this paper, an adaptive proportional integral derivative deep feedforward network (APIDDFN) controller is proposed for stabilizing the trajectory tracking control of an underactuated quadrotor system. The deep feedforward neural network (DFN) is used to online adjusting the PID controller parameters. A one-step secant algorithm is utilized to initialize the DFN system weights and biases. Besides, a multidimensional particle swarm optimization (PSO) algorithm is utilized to optimize the DFN parameters. The simulation results illustrate that the proposed PSO-APIDDFN controller can achieve faster convergence speed and show minimum overshoot than the non-optimized adaptive PID feedforward network and adaptive PID deep feedforward network. These results demonstrate the efficiency of the proposed method in designing an intelligent controller for stabilizing the quadrotor UAV. Future works are related to integrating an observer in the control system to estimate unmeasured translational and rotational quadrotor system states and rates.

REFERENCES

[1] W. Gu, K. P. Valavanis, M. J. Rutherford, and A. Rizzo, "UAV model-based flight control with artificial neural networks: a survey," *Journal of Intelligent and Robotic Systems*, vol. 100, no. 3–4, pp. 1469–1491, Dec. 2020, doi: 10.1007/s10846-020-01227-8.

[2] S. Y. Choi and D. Cha, "Unmanned aerial vehicles using machine learning for autonomous flight; state-of-the-art," *Advanced Robotics*, vol. 33, no. 6, pp. 265–277, Mar. 2019, doi: 10.1080/01691864.2019.1586760.

[3] A. Carrio, C. Sampedro, A. Rodriguez-Ramos, and P. Campoy, "A review of deep learning methods and applications for unmanned aerial vehicles," *Journal of Sensors*, vol. 2017, pp. 1–13, 2017, doi: 10.1155/2017/3296874.

[4] A. R. Al Tahtawi and M. Yusuf, "Low-cost quadrotor hardware design with pid control system as flight controller," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 17, no. 4, pp. 1923–1930, Aug. 2019, doi: 10.12928/TELKOMNIKA.v17i4.9529.

[5] Y. Al Younes, A. Drak, H. Noura, A. Rabhi, and A. El Hajjaji, "Robust model-free control applied to a quadrotor UAV," *Journal of Intelligent and Robotic Systems*, vol. 84, no. 1–4, pp. 37–52, Dec. 2016, doi: 10.1007/s10846-016-0351-2.

[6] M. A. Basri and A. Noordin, "Optimal backstepping control of quadrotor UAV using gravitational search optimization algorithm," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 9, no. 5, pp. 1819–1826, Oct. 2020, doi: 10.11591/eei.v9i5.2159.

[7] E. Chater, H. Housny, and H. El Fadil, "Robust control design for quadrotor trajectory path tracking," in *2019 8th International Conference on Systems and Control (ICSC)*, Oct. 2019, pp. 21–26, doi: 10.1109/ICSC47195.2019.8950509.

[8] A. Noordin, M. A. M. Basri, and Z. Mohamed, "Sliding mode control for altitude and attitude stabilization of quadrotor UAV with external disturbance," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 7, no. 2, pp. 203–210, May 2019, doi: 10.52549/ijeei.v7i2.1149.

[9] E. A. Chater, H. Housny, and H. El Fadil, "Robust sliding mode control for quadrotor UAV," in *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, Dec. 2020, pp. 1–6, doi: 10.1109/ICECOCS50124.2020.9314584.

[10] N. I. Vitzilaios and N. C. Tsourveloudis, "An experimental test bed for small unmanned helicopters," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 5, pp. 769–794, May 2009, doi: 10.1007/s10846-008-9284-8.

[11] N. Maharani Raharja, E. Firmansyah, A. Imam Cahyadi, and I. Iswanto, "Hovering control of quadrotor based on fuzzy logic," *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 8, no. 1, pp. 492–504, Mar. 2017, doi: 10.11591/ijpeds.v8.i1.pp492-504.

[12] I. Iswanto, O. Wahyunggoro, and A. Imam Cahyadi, "Path planning based on fuzzy decision trees and potential field," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 1, pp. 212–222, Feb. 2016, doi: 10.11591/ijece.v6i1.8606.

[13] J. Muliadi and B. Kusumoputro, "Neural network control system of UAV altitude dynamics and its comparison with the PID control system," *Journal of Advanced Transportation*, vol. 2018, pp. 1–18, 2018, doi: 10.1155/2018/3823201.

[14] H. Housny, E. A. Chater, and H. El Fadil, "Multi closed-loop adaptive neuro-fuzzy inference system for quadrotor position control," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 5, pp. 526–535, 2020, doi: 10.25046/aj050565.

[15] H. Housny, E. A. Chater, and H. El Fadil, "Observer-based enhanced ANFIS control for a quadrotor UAV," *International Review on Modelling and Simulations (IREMOS)*, vol. 14, no. 1, Feb. 2021, doi: 10.15866/iremos.v14i1.18991.

[16] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, "Stability of small-scale UAV helicopters and quadrotors with added payload mass under PID control," *Autonomous Robots*, vol. 33, no. 1–2, pp. 129–142, Aug. 2012, doi: 10.1007/s10514-012-9280-5.

[17] A. Noordin, M. A. M. Basri, and Z. Mohamed, "Simulation and experimental study on PID control of a quadrotor MAV with perturbation," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 9, no. 5, pp. 1811–1818, Oct. 2020, doi: 10.11591/eei.v9i5.2158.

[18] H.-K. Tran and J.-S. Chiou, "PSO-based algorithm applied to quadcopter micro air vehicle controller design," *Micromachines*, vol. 7, no. 9, Sep. 2016, doi: 10.3390/mi7090168.

[19] S.-E.-I. Hasseni and L. Abdou, "Decentralized PID control by using GA optimization applied to a quadrotor," *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 12, no. 2, pp. 33–44, Jun. 2018, doi: 10.14313/JAMRIS_2-2018/9.

[20] T. K. Priyambodo, A. Dharmawan, O. A. Dhewa, and N. A. S. Putro, "Optimizing control based on fine tune PID using ant colony logic for vertical moving control of UAV system," *AIP Conference Proceedings*, 2016, doi: 10.1063/1.4958613.

[21] H. Housny, E. A. Chater, and H. El Fadil, "Fuzzy PID control tuning design using particle swarm optimization algorithm for a quadrotor," in *2019 5th International Conference on Optimization and Applications (ICOA)*, Apr. 2019, pp. 1–6, doi: 10.1109/ICOA.2019.8727702.

[22] A. Benbouali, F. Chabni, R. Taleb, and N. Mansour, "Flight parameters improvement for an unmanned aerial vehicle using a lookup table based fuzzy PID controller," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 23, no. 1, pp. 171–178, Jul. 2021, doi: 10.11591/ijeecs.v23.i1.pp171-178.

[23] S. Bari, S. S. Zehra Hamdani, H. U. Khan, M. ur Rehman, and H. Khan, "Artificial neural network based self-tuned PID controller for flight control of quadcopter," in *2019 International Conference on Engineering and Emerging Technologies (ICEET)*, Feb. 2019, pp. 1–5, doi: 10.1109/CEET1.2019.8711864.

[24] Y. Chen, Y. He, and M. Zhou, "Decentralized PID neural network control for a quadrotor helicopter subjected to wind disturbance," *Journal of Central South University*, vol. 22, no. 1, pp. 168–179, Jan. 2015, doi: 10.1007/s11771-015-2507-9.

[25] Y.-F. Teng, B. Hu, Z.-W. Liu, J. Huang, and Z.-H. Guan, "Adaptive neural network control for quadrotor unmanned aerial vehicles," in *2017 11th Asian Control Conference (ASCC)*, Dec. 2017, pp. 988–992, doi: 10.1109/ASCC.2017.8287305.

[26] N. M. Ibrahim, S. T. F. Al-Janabi, and B. Al-Khateeb, "Electricity-theft detection in smart grids based on deep learning," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 2285–2292, Aug. 2021, doi: 10.11591/eei.v10i4.2875.

[27] Y. Zhang, P. Phillips, S. Wang, G. Ji, J. Yang, and J. Wu, "Fruit classification by biogeography-based optimization and feedforward neural network," *Expert Systems*, vol. 33, no. 3, pp. 239–253, Jun. 2016, doi: 10.1111/exsy.12146.

[28] B.-S. Oh, K.-A. Toh, A. B. J. Teoh, and Z. Lin, "An analytic Gabor feedforward network for single-sample and pose-invariant face recognition," *IEEE Trans. on Image Processing*, vol. 27, no. 6, pp. 2791–2805, Jun. 2018, doi: 10.1109/TIP.2018.2809040.

[29] S. Edhah, S. Mohamed, A. Rehan, M. AlDhaheri, A. AlKhaja, and Y. Zweiri, "Deep learning based neural network controller for quad copter: application to hovering mode," in *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, Nov. 2019, pp. 1–5, doi: 10.1109/ICECTA48151.2019.8959776.

[30] H. Housny, E. Chater, and H. El Fadil, "Multi-closed-loop design for quadrotor path-tracking control," in *2019 8th International Conference on Systems and Control (ICSC)*, Oct. 2019, pp. 27–32, doi: 10.1109/ICSC47195.2019.8950659.

[31] D. Lee, J. Sim, K. Han, C. Kim, and G. O. Koh, "Angle-of-attack command longitudinal control for supersonic advanced trainer aircraft," *Int. J. of Aeronautical and Space Sciences*, vol. 22, no. 1, pp. 120–128, Feb. 2021, doi: 10.1007/s42405-020-00279-2.

[32] J. Chen, L. Wang, J. Fu, and Z. Yang, "Engineering comprehensive model of complex wind fields for flight simulation," *Aerospace*, vol. 8, no. 6, May 2021, doi: 10.3390/aerospace8060145.

[33] H. Ji, R. Chen, and P. Li, "Analysis of helicopter handling quality in turbulence with recursive Von Kármán model," *Journal of Aircraft*, vol. 54, no. 5, pp. 1631–1639, Sep. 2017, doi: 10.2514/1.C034189.

[34] A. A. Suratgar, M. B. Tavakoli, and A. Hoseinabadi, "Modified Levenberg-Marquardt method for neural networks training," *World Acad Sci Eng Technol*, vol. 6, no. 1, pp. 46–48, 2005.

[35] M. Brunato and R. Battiti, "A telescopic binary learning machine for training neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 665–677, Mar. 2017, doi: 10.1109/TNNLS.2016.2537300.

[36] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, pp. 387–408, Jan. 2018, doi: 10.1007/s00500-016-2474-6.

[37] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–38, 2015, doi: 10.1155/2015/931256.

[38] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," Lausanne, EPFL, 2007, 10.5075/epfl-thesis-3727.

[39] K. P. Valavanis, Ed., *Advances in unmanned aerial vehicles*. Dordrecht: Springer Netherlands, 2007.

BIOGRAPHIES OF AUTHORS

El Ayachi Chater received a B.S. degree in Electronics from the ENSET, Hassan II University, Mohammedia, Morocco, in 1992, and the Agrégation degree in Electrical Engineering from the ENSET, Mohammed V University, Rabat, Morocco, in 1999. He received an M.S. and a Ph.D. Degrees in Control Engineering from the EMI, Mohammed V University, Rabat, Morocco, in 2002 and 2013, respectively. From 2016, he was an assistant professor, Physics department, ENS, Mohammed V University, Rabat, Morocco. Then, from 2019, he has been teaching control systems and industrial computing, electrical engineering department, EST, Salé, Morocco. His research interest includes nonlinear and intelligent control of electromechanical and mechatronic systems. He has published several journal/conference papers on these topics. He can be contacted at email: elayachi.chater@est.um5.ac.ma.

Halima Housny received a B.S. degree in EEA from the FST, Hassan II University, Mohammedia, Morocco, in 1998. She received an M.S. degree in control engineering from the EMI, Mohammed V University, Rabat, Morocco, in 2003. Then, she received a Ph.D. degree in Electrical Engineering with ISA Lab., ENSA, Ibn Tofail university, Kenitra, Morocco, in 2021. Since 2003, she was teaching computer sciences in high schools, Salé, Morocco. Her research interest is the optimization and intelligent control strategies for unmanned aerial vehicles. She has already published several papers on this topic. She can be contacted at email: housny.hal@gmail.com.

Hassan El Fadil received a B.S. degree in Electronics, the Agrégation degree in Electrical Engineering, from the ENSET, Mohammed V University, Rabat, Morocco, in 1994 and 1999, respectively. He received an M.S. degree and a Ph.D. degree in Automatic Control from the EMI, Mohammed V University, Rabat, Morocco, in 2002 and 2008, respectively. He is currently Professor at the ENSA, Ibn Tofail University, Kenitra, Morocco. His research interest includes nonlinear and adaptive control, power converters and electric motors control, renewable energy, distributed energy resources, smart grid, and electric vehicles. He has published over 100 journals/conference papers on these topics. Hassan El Fadil is with ISA Lab., ENSA, Ibn Tofail University, Kenitra, Morocco. He can be contacted at email: elfadilhassan@yahoo.fr.