Neural Network and Fuzzy-logic-based Self-tuning PID Control for Quadcopter Path Tracking

Khadija EL HAMIDI[1]*, Mostafa MJAHED[2], Abdeljalil El KARI[1], Hassan AYAD[1]

[1] Applied Physics Department, Cadi Ayyad University, Marrakesh 40000, Morocco
khadija.elhamidi91@gmail.com (*Corresponding author*), jalil.elkari@gmail.com, ayad.ha@gmail.com

[2] Mathematics and Systems Department, Royal School of Aeronautics, Marrakesh 40000, Morocco
mjahed.mostafa@gmail.com

Abstract: The purpose of this research is to design adaptive control methods for addressing the stabilization and trajectory tracking problems in a quadcopter unmanned aerial vehicle (UAV). To accomplish these tasks, a comparative study of the Proportional Integral Derivative (PID) and PD controllers is performed. Intelligent algorithms (IAs) have been used to tune the conventional structure of PID/PD controllers. The proposed hybrid intelligent controllers consist of the neural network PID/PD (NNPID/PD) and the Optimized Fuzzy PID/PD based on the Particle Swarm Optimization (FPID/PD-PSO). Adaptive neural networks are deployed to schedule PID/PD gains, the improved back-propagation algorithm is used to update the weights of the neural network. Then, an effective control approach based on adaptive PID Fuzzy logic and Particle Swarm Optimization (PSO) algorithm has been applied. PSO algorithm is introduced to adjust the scaling factors for improving the convergence speed and production rate. Finally, in order to demonstrate the robustness of the proposed control methods, disturbances in the quadcopter system are added. The results so obtained demonstrate the effectiveness of the proposed control strategy.

Keywords: Neural Network, Fuzzy Logic, PSO, Control nonlinear system, Trajectory Tracking, PID/PD.

1. Introduction

The Quadcopter unmanned aerial vehicles (UAVs) becomes a promising option for applications in various areas. It can be utilized for military and civil purposes due to their many advantages such as aerial surveillance, search and rescue missions, fire detection. They can be also boarded without human during complex or risky missions.

The control techniques for UAV platforms have witnessed a rapidly expanding research in order to achieve not only autonomous hovering and orientation but also trajectory tracking (Salehfard et. al, 2018, Rendón & Martins, 2017). In this context, there is a need to develop a simplified method that can be used to design a controller. Several applied methods have been employed to solve the attitude stabilization and trajectory tracking problems (Labbadi & Cherkaoui, 2019, Zheng et al. 2014, Basri, 2018).

The proportional, integral and Derivative (PID/PD) controllers have been successfully used in the industry owing to their simplicity and suitable performance for both linear and nonlinear systems. The PID controllers are used for both quadcopter attitude and trajectory controls, the values of controlling parameters were tuned by particle swarm optimization (PSO) (Estevez, et al. 2016). Alongside a quadcopter is a Multiple-Input Multiple-Output (MIMO) system affected by various uncertainties such as nonlinear dynamics and external disturbances. Therefore, the linear controllers are effective only for the small ranges

of angles. To solve the stabilization problem of quadrotor, different nonlinear control learning algorithms have been developed. The fuzzy logic control (Harik et al., 2017, Mardan et al., 2017, Gül & Arıso, 2013) and the neural network controller (Çakir & Yüksel, 2017, Muliadi & Kusumoputro, 2018, Lei et al.,2014) are used to build and design an intelligent flight control module for a quadrotor UAV, and they are widely used for nonlinear plants due to their characteristic of self-learning, self-organizing and self-adapting (Belhadri et al., 2016). So, combining the characteristics of PID/PD controllers with intelligent algorithms represents a very promising technique for a nonlinear system. The designed neural network and Fuzzy PID control algorithms are tested using the full nonlinear mathematical model of the considered Quadcopter with disturbed and variable inputs (Bojja et al., 2019, Kuantama et al., 2017, Chen et al., 2015). As mentioned in previous researches, the neural networks and the fuzzy logic PID controllers can achieve good results in controlling an intelligent vehicle, and provide safe driving. These control techniques have proven to be robust against system parameter variations.

In the present work, two different nonlinear control approaches have been studied with the aim of allowing the Quadcopter to track complex trajectories and reject disturbances. The first control method proposed is an adaptive control algorithm based on neural network. An auto-tune PID/PD-like controller based on an online

Neural Networks (NN) is developed in this research for trajectory tracking. The improved back-propagation algorithm is used to optimize the weights of the neural network.

The second control law to be designed and implemented in the Simulink simulation environment is the Fuzzy logic-PID (FPID/PD) control law. The tuning of PID/PD gains is performed by a fuzzy logic method. However, such a gains scheduling method of a PID and fuzzy controllers has some drawbacks and limitations. The design of FLC (Fuzzy Logic Control) is based on the perfect choice of the limits and a perfect shaping of the membership functions and scaling factors (SFs). The Particle Swarm Optimization (PSO) algorithm-based approach is proposed to improve the effectiveness of the FPID/PD. Scaling factors are scheduled based on the improved variants of the conventional PSO algorithm.

This paper is structured as follows: the mathematical non-linear model of the studied Quadcopter UAV based on the Newton-Euler formalism is described in section 2. In section 3, the design of the flight adaptive intelligent controller for path tracking is established. In Section 4, to verify the performance of the proposed method, simulation results are presented. Finally, the main conclusions of this paper are given in section 5.

2. Quadcopter Model

The generic dynamics of the quadcopter are derived through Newton-Euler formulation **(Basri, 2018, El Hamidi et al.,2019)** and written as expressed in (1- 6).

$$\ddot{\varphi} = \frac{lU_2}{I_x} + \frac{(I_y - I_z)}{I_x}\left(-\dot{\theta}\sin\varphi + \dot{\psi}\cos\varphi\cos\theta\right)$$

$$\left(\dot{\theta}\cos\varphi + \dot{\psi}\sin\phi\cos\theta\right) - \left(\dot{\varphi}^2 - 2\dot{\varphi}\dot{\psi}\sin\theta^2\right)\frac{K_{fax}}{I_x}$$

$$-\left(\dot{\theta}\cos\varphi + \dot{\psi}\sin\varphi\cos\theta\right)\frac{J_r\left(\omega_1 - \omega_2 + \omega_3 - \omega_4\right)}{I_x} \quad (1)$$

$$\ddot{\theta} = \frac{lU_3}{I_y} + \frac{(I_z - I_x)}{I_y}\left(\dot{\psi}\cos\varphi\cos\theta - \dot{\theta}\sin\varphi\right)$$

$$\left(\dot{\varphi} - \dot{\psi}\sin\theta\right) - \left(\dot{\psi}\sin\theta - \dot{\varphi}\right)\frac{J_r\left(\omega_1 - \omega_2 + \omega_3 - \omega_4\right)}{I_y} - \frac{K_{fay}}{I_y} \quad (2)$$

$$\left(\dot{\theta}^2\cos\varphi^2 + 2\dot{\varphi}\dot{\psi}\cos\theta\sin\varphi\cos\varphi + \dot{\psi}^2\cos\theta^2\sin\varphi^2\right)$$

$$\ddot{\psi} = \frac{U_4}{I_z} + \frac{(I_x - I_y)}{I_z}\left(\dot{\psi}\sin\phi\cos\theta + \dot{\theta}\cos\varphi\right)\left(\dot{\varphi} - \dot{\psi}\sin\theta\right)$$

$$-\left(\dot{\psi}\sin\theta - \dot{\varphi}\right)\frac{J_r\left(\omega_1 - \omega_2 + \omega_3 - \omega_4\right)}{I_z} - \frac{K_{faz}}{I_z} \quad (3)$$

$$\left(\dot{\theta}^2\sin\varphi^2 - 2\dot{\varphi}\dot{\psi}\cos\theta\sin\varphi\cos\varphi + \dot{\psi}^2\cos\theta^2\cos\varphi^2\right)$$

$$\ddot{x} = \frac{\cos\varphi\cos\psi\sin\theta + \sin\varphi\sin\psi}{m}U_1 - \frac{K_{ftx}}{m}\dot{x} \quad (4)$$

$$\ddot{y} = \frac{\sin\psi\sin\theta\cos\varphi - \cos\psi\sin\varphi}{m}U_1 - \frac{K_{fty}}{m}\dot{y} \quad (5)$$

$$\ddot{z} = -g + \frac{\cos\varphi\cos\theta}{m}U_1 - \frac{K_{ftz}}{m}\dot{z} \quad (6)$$

$$\begin{cases} u_1 = b\left(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2\right) \\ u_2 = b\left(\omega_4^2 - \omega_2^2\right) \\ u_3 = b\left(\omega_3^2 - \omega_1^2\right) \\ u_4 = d\left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) \end{cases} \quad (7)$$

where I_x, I_y and I_z represent the moment inertia of the quadcopter around x, y and z axis; Jr is the rotor inertia; k_{ftx}, k_{fty}, k_{ftz} are the translation drags coefficients; k_{fax}, k_{fay} and k_{faz} are the frictions aerodynamics coefficients; ϕ, θ and ψ are the rotation around the roll, pitch and yaw axis, respectively. g is the gravitational acceleration, and l represents the Quadcopter half span. Equation (7) provides the mathematical expression of the input forces of the quadcopter.

3. Flight Control Synthesis

In this work, intelligent PID/PD controllers are used instead of the linear PID/PD controllers. So, the control strategy is designed by combining conventional and intelligent control techniques. The main approaches proposed in this research are neural networks, fuzzy logic control and PID/PD control.

Figure 1 shows the block diagram of the proposed control architecture scheme. Two control loops are used in this research: the outer-loop position which contains the translation position coordinates (x, y and z) and the inner-loop attitude which contains the three angles (φ, θ and ψ).
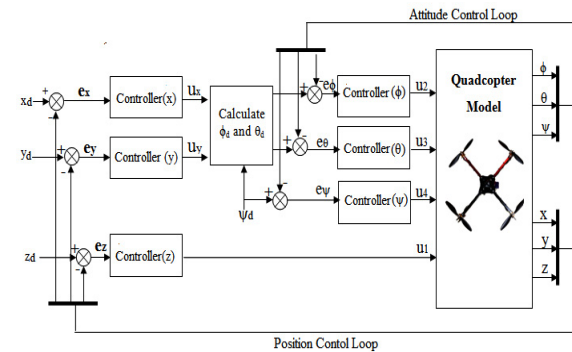

Figure 1. Control scheme

Note that from the dynamical model presented in (1 to 6), the quadcopter is constituted of six outputs which are controlled only by four

inputs. Unlike the altitude and orientation of the quadcopter, its x and y positions are not decoupled and cannot be directly controlled using any of the four control laws $u1$ to $u4$. On the other hand, the x and y positions can be controlled indirectly by the roll and pitch angles. The desired roll (ϕ_d) and pitch (θ_d) angles can be computed from the motion translation equations (4-5).

The following notations are introduced:

$$\begin{cases} u_x = (\cos\psi\,\sin\theta\,\cos\varphi + \sin\psi\,\sin\varphi)\,u_1 \\ u_y = (\sin\psi\,\sin\theta\,\cos\varphi - \cos\psi\,\sin\varphi)\,u_1 \end{cases} \quad (8)$$

The desired angles (Roll, Pitch) can be extracted from the following expressions:

$$\begin{cases} \phi_d = \arcsin\left(\dfrac{u_x\,s\psi_d - u_y\,c\psi_d}{\sqrt{u_x^2 + u_y^2 + u_1^2}}\right) \\ \\ \theta_d = \arctan\left(\dfrac{u_x\,c\psi_d + u_y\,s\psi_d}{\sqrt{u_x^2 + u_y^2 + u_1^2}}\right) \end{cases} \quad (9)$$

where ϕ_d, θ_d and ψ_d are the desired trajectories in roll, pitch and yaw respectively.

3.1 Neural Network PID/PD control Structure

Self-tuning (El Hamidi et al.,2018; El Hamidi et al., 2019) has received great attention for control problems since it provides a systematic and flexible approach for dealing with uncertainties, non-linearities and time-varying parameters. In order to achieve an optimal system performance, a novel intelligent PID/PD control strategy based on backpropagation neural network (BPNN) is designed for quadcopter control. Figure 2 presents the structure of the proposed gain scheduling neural network GS-NNPID/PD controller that includes two main parts: a conventional PID/PD controller employed for the quadcopter control and a BPNN neural network used for adjusting PID/PD parameters: k_p, k_i and k_d. The attitude vector X_1 and position vector $X2$ are defined in (10).
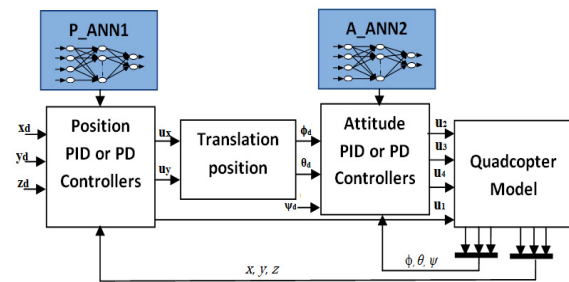
$$X = \begin{bmatrix} X_1 & X_2 \end{bmatrix} = \begin{bmatrix} \varphi & \theta & \psi & x & y & z \end{bmatrix} \quad (10)$$

The controller is designed using the incremental PID/PD control algorithm and multilayer neural network. The PID and PD control equations can be expressed in (11) and (12) respectively.

$$\begin{aligned} u_{PID}(k) &= u_{PID}(k-1) + k_p\left[e(k) - e(k-1)\right] + k_i e(k) \\ &+ k_d\left[e(k) - 2e(k-1) + e(k-2)\right] \end{aligned} \quad (11)$$

$$\begin{aligned} u_{PD}(k) &= u_{PD}(k-1) + k_p\left[e(k) - e(k-1)\right] \\ &+ k_d\left[e(k) - 2e(k-1) + e(k-2)\right] \end{aligned} \quad (12)$$

where $u_{PID}(k)$ and $u_{PD}(k)$ are the PID and PD control signal respectively, $e(k) = X_d - X$ represents the tracking error, $X_d = [X_{1d}\,X_{2d}] = [x_d\,y_d\,z_d\,\phi_d\,\theta_d\,\psi_d]$ denotes the desired reference, k_p, k_i and k_d are the proportional, integral and derivative gains respectively, and k is the sample time.

The gains of the PID/PD controller are adjusted according to the learning of the BPNN. The BPNN consists of an input layer composed of four inputs, a hidden layer made up of the hidden neurons, and an output layer comprised of six outputs, which represent the gains of the PID/PD parameters.

As shown in Figure 1, six independent PID/PD were implemented, one for each flight parameter.

The whole system is formed of two neural networks as presented in Figure 2. The position neural network (P_ANN1) is implemented to adjust the PID/PD gains of position outer loop. The second one is the attitude neural network (A_ANN2), which is exploited to regulate the gains of the three PID/PD in the attitude inner loop. In each controller, the GS-NNPID/PD is built by combining PID/PD and feedforward neural network as presented in Figure 3.
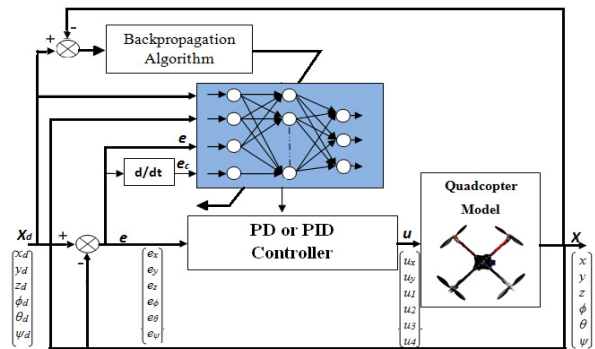

Figure 2. Block Diagram of an Auto-Tuned PID/PD with Artificial NN Control

Figure 3. Block diagram of the implemented GS NNPID/PD

The input layer has 4 neurons; the specific form is given by (13).

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{pmatrix} = \begin{pmatrix} X_{id} \\ X_i \\ e_i \\ e_{ci} \end{pmatrix}_{i=\{1;2\}} \tag{13}$$

where μ is the input vector, e is the error vector and ec is the changing rate of system error expressed in (14).

$$e_{ci}(k) = e_i(k) - e_i(k-1) \tag{14}$$

The Output of each neuron in the input layer is expressed as:

$$Out_i^{(1)} = x(i) \quad i = 1...4 \tag{15}$$

$$\begin{cases} Out_1^{(1)} = x(1) = X_{id} \\ Out_2^{(1)} = x(2) = X_i \\ Out_3^{(1)} = x(3) = e_i \\ Out_4^{(1)} = x(4) = e_{ci} \quad i = 1;2 \end{cases} \tag{16}$$

Hidden layer has $n=10$ neurons. The Input and output of each hidden layer neuron can be calculated respectively as follows:

$$in_j^{(2)}(k) = \sum_{i=1}^{4} v_{ij} \, Out_i^{(1)} \quad j = 1...n \tag{17}$$

$$Out_j^{(2)} = f\,(in_j^{(2)}(k)) \tag{18}$$

Output layer has m=6 neurons. The neurons placed on this layer correspond to the PID/PD gains. The input and output of each neuron in this layer are expressed respectively as follows:

$$in_l^{(3)}(k) = \sum_{j=1}^{n} w_{lj} \, Out_j^{(3)}(k) \quad l = 1...m \tag{19}$$

$$Out_l^{(3)} = g\,(in_l^{(3)}(k)) \tag{20}$$

where superscript (1), (2), (3) stands for the input layer, hidden layer and output layer respectively, v_{ij} is the weight connecting the input layer neurons to the hidden layer neurons, w_{jl} is the weight connecting the hidden layer neurons to the output layer neurons, f and g are the activation functions of the hidden and output layer, respectively. They are defined by (21) and (22).

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{21}$$

$$g(x) = \frac{1}{2}(1 + \tanh(x)) = \frac{e^x}{e^x + e^{-x}} \tag{22}$$

The backpropagation algorithm is based on the minimization of a sum-squared error (MSE) utilizing the optimization gradient descent method. MSE is used as the cost function which is a function of error defined as follows:

$$E(k) = \frac{1}{2}e(k)^2 \tag{23}$$

Generally, the adjustment of each weight of hidden-layer and output-layer can be expressed as follows:

$$\Delta w_{ij}(k) = -\eta \frac{\partial E(k)}{\partial \Delta w_{ij}} \tag{24}$$

However, in order to avoid the local minima which represent the best-known problem associated with back-propagation algorithm, a momentum term is added to the weight change in the proposed algorithm. This means that the weight changes this iteration by depending not only on the current error, but also on the previous changes. So, the connection weights at the output layer and the hidden layer are updated by the following equations:

$$\Delta w_{jl}(k) = -\eta \frac{\partial E(k)}{\partial \Delta w_{jl}} + \alpha \Delta w_{jl}(k-1) \tag{25}$$

$$\Delta v_{ij}(k) = -\eta \frac{\partial E(k)}{\partial \Delta v_{ij}} + \alpha \Delta v_{ij}(k-1) \tag{26}$$

where μ is the learning rate; α is the is momentum factor.

Since

$$\frac{\partial E(k)}{\partial \Delta w_{jl}} =$$

$$\frac{\partial E(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial \Delta u(k)} \cdot \frac{\partial \Delta u(k)}{\partial O_l^{(3)}(k)} \cdot \frac{\partial O_l^{(3)}(k)}{\partial in_l^{(3)}(k)} \cdot \frac{\partial in_l^{(3)}(k)}{\partial w_{jl}^{(3)}(k)} \tag{28}$$

$$\frac{\partial E(k)}{\partial y(k)} = e(k) \tag{29}$$

$$\frac{\partial in_l^{(3)}(k)}{\partial v_{jl}^{(3)}(k)} = out_j^{(2)}(k) \tag{30}$$

$$\frac{\partial Out_l^{(3)}(k)}{\partial in_l^{(3)}(k)} = g'(in_l(k)) \tag{31}$$

and then

$$\begin{cases} \dfrac{\partial \Delta u(k)}{\partial Out^{(3)}(k)} = e(k) - e(k-1) \;; if \; Out_l = k_p \\[2mm] \dfrac{\partial \Delta u(k)}{\partial Out^{(3)}(k)} = e(k) \;; if \; Out_l = k_i \\[2mm] \dfrac{\partial \Delta u(k)}{\partial Out^{(3)}(k)} = e(k) - 2e(k-1) + e(k-2) \;; if \; Out_l = k_d \end{cases} \tag{32}$$

The term $\partial y(k)/\partial \Delta u(k)$ is unknown, it is replaced by $sgn(\partial y(k)/\partial \Delta u(k))$. The above analysis can lead

to the adjustment of the weights of the output layer by using the following formulas:

$$\begin{cases} \Delta w_{jl}(k) = \alpha \Delta w_{jl}(k-1) + \eta \delta_l^{(3)} Out_j^{(2)}(k) \\ \delta_l^{(3)} = e(k)\,\text{sgn}\left(\dfrac{\partial y(k)}{\partial \Delta u(k)}\right)\dfrac{\partial \Delta u(k)}{\partial Out_l^{(3)}(k)}\,g'(in_l^{(3)}(k)) \end{cases} \quad (33)$$

$(l = 1, 2..m)$

The learning algorithm of the weighted hidden layer can also be led to:

$$\begin{cases} \Delta v_{jl}(k) = \alpha \Delta v_{jl}(k-1) + \eta \delta_j^{(2)} Out_i^{(1)}(k) \\ \delta_j^{(2)} = f'(in_j^{(2)}(k)) \sum\limits_{l=1}^{m} \delta_l^{(3)} \Delta w_{ij}(k) \end{cases} \quad (34)$$

$(j = 1, 2..n)$

The derivative functions of $f(x)$ and $g(x)$ are:

$$\begin{cases} f'(.) = (1 - f^2(x)) \\ g'(.) = 2g(x)(1 - g(x)) \end{cases} \quad (35)$$

3.2 Design of Optimized Fuzzy FPID/ PD-PSO

A combination between proportional integral derivative controller and a fuzzy system is applied in order to control the defined reference trajectory tracking of the quadcopter. The fuzzy logic system is developed to calculate the proportional integral derivative gains. An improved fuzzy PID/ PD controller with scaling factors based on the heuristic Particle Swarm Optimization (FPID/PD PSO) is considered to steer the system trajectories toward the desired dynamics.

The scaling factors of the proposed scheme are further tuned using the heuristic method PSO. This design is composed of the conventional PID/PD control system in conjunction with a set of fuzzy rules (knowledge base) and a fuzzy reasoning mechanism. Using the knowledge and fuzzy inference to regulate PID/PD gains, as a result, the PID/PD controller generates the control signal. The proposed structure of the optimized fuzzy FPID/PD-PSO controller is shown in Figure 4. The input variables of such a fuzzy supervisor are the error e and its derivative de as given in (36) and (37).

$$e(t) = rin(t) - yout(t) \quad (36)$$

$$de(t) = \frac{de(t)}{dt} \quad (37)$$

where $rin = \begin{bmatrix} \phi_d\,\theta_d\,\psi_d\,x_d\,y_d\,z_d \end{bmatrix}^T$ denotes the desired reference and $yout = \begin{bmatrix} \phi\,\theta\,\psi\,x\,y\,z \end{bmatrix}^T$ represents the output of the controlled quadcopter. The PID

and PD controller are described in equations (38) and (39) as:

$$u_{PID} = k_p e(t) + k_i \int e(t) + k_d \frac{de(t)}{dt} \quad (38)$$

$$u_{PD} = k_p e(t) + k_d \frac{de(t)}{dt} \quad (39)$$

The output variables of the optimized fuzzy tuning gains are obtained by using the fuzzy reasoning inference and optimization algorithm. These variables denote the optimal proportional integral and derivative gains k_{po}, k_{io} and k_{do} which are given respectively as follows:

$$\begin{cases} k_{po} = G_p k_p \\ k_{po} = G_i k_i \\ k_{po} = G_d k_d \end{cases} \quad (40)$$

where $k_p = [k_{p\phi}\,k_{p\theta}\,k_{\psi p}\,k_{px}\,k_{py}\,k_{pz}]^T$, $ki = [k_{i\phi}\,k_{i\theta}\,k_{i\psi}\,k_{ix}\,k_{iy}\,k_{iz}]^T$ and $k_d = [k_{d\phi}\,k_{d\theta}\,k_{d\psi}\,k_{dx}\,k_{dy}\,k_{dz}]^T$ denote the fuzzy inference reasoning outputs for proportional, integral and derivative actions of PID/PD controllers, respectively.

The parameters, $G_p = [G_{p\phi}\,G_{p\theta}\,G_{p\psi}\,G_{px}\,G_{py}\,G_{pz}]^T$, $G_i = [G_{i\phi}\,G_{i\theta}\,G_{i\psi}\,G_{ix}\,G_{iy}\,G_{iz}]^T$, $G_d = [G_{d\phi}\,G_{d\theta}\,G_{d\psi}\,G_{dx}\,G_{dy}\,G_{dz}]^T$, $ke = [ke_\phi\,ke_\theta\,ke_\psi\,ke_x\,ke_y\,ke_z]^T$ and $kec = [kec_\phi\,kec_\theta\,kec_\psi\,kec_x\,kec_y\,kec_z]^T$ are the positive input and output scaling factors, introduced to eliminate the classical predefined ranges on k_p, k_i and k_d parameters of the FPID/PD controllers.
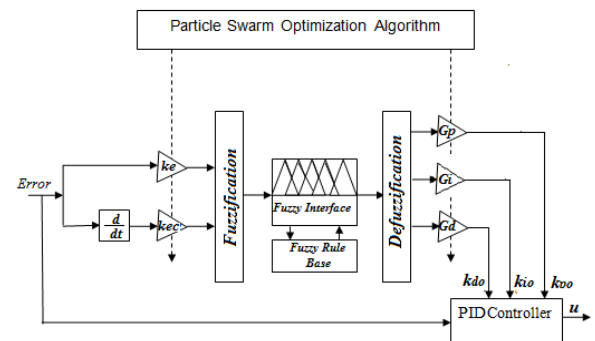

Figure 4. Structure of the proposed of adaptive optimized Fuzzy FPID_PSO Controller

A set of linguistic rules is used in the fuzzy reasoning inference block to determine the parameters kp, ki and kd. Seven fuzzy labels (Negative Big (*NB*), Negative Medium(*NM*), Negative Small (*NS*), Zero (*ZO*), Positive Small (*PS*), Positive Medium (*PM*)and Positive Big (*PB*)) are used for the fuzzy input variables and three fuzzy labels (Small (*S*), Medium (*M*) and Big (*B*)) for the fuzzy output variable. These

linguistic variables are listed in Table 1 and Table 2 which contain the 25 rules for the fuzzy-PID type controller. The membership functions for the input and output variables are defined with the triangular, uniformly and distributed shape.

Table 1. Fuzzy rules for k_p and k_i

		ERROR						
		NB	NM	NS	ZO	PS	PM	PB
ERROR RATE	NB	M	S	S	S	S	S	M
	NM	B	M	S	S	S	M	B
	NS	B	B	M	S	M	B	B
	ZO	B	B	B	M	B	B	B
	PS	B	B	M	S	M	B	B
	PM	B	M	S	S	S	M	B
	PB	M	S	S	S	S	S	M

Table 2. Fuzzy rules for k_d

		ERROR						
		NB	NM	NS	ZO	PS	PM	PB
ERROR RATE	NB	M	B	B	B	B	B	M
	NM	S	M	B	B	B	M	S
	NS	S	S	M	B	M	S	S
	ZO	S	S	S	M	S	S	S
	PS	S	S	M	B	M	S	S
	PM	S	M	B	B	B	M	S
	PB	M	B	B	B	B	B	M

For the FLC related to attitude, the error is normalized to the interval [-1 1], the error rate is confined within the range [-10 10] and the output is also normalized to the ranges [0 100], [0 1] and [0 30] for k_p, k_i and k_d respectively. For the quadcopter position control requirement, the domain of the displacement deviation e is set as [-5 5], and the domain of the de is [-10 10]. The domains of k_p, k_i and k_d are [0 200], [0 25] and [0 100] respectively. Once the rule base has been defined, the membership functions (MFs) for e, de, k_p, k_i and k_d need to be determined. Identical membership functions are used for all variables. By using the product-sum inference and the center of gravity defuzzification method, the overall outputs of each FLC can be obtained.

Generally, the fuzzy parameters are selected by trial and error. Sometimes, this is not adequate to provide the necessary control actions. In such cases, the static values of the scaling factors SFs and single MFs are not enough to achieve

the desired control action. A straightforward solution to this problem is to propose a novel strategy to design and fine-tune the input/output scaling factor parameters of the fuzzy gain scheduling PID (GS-FPID) controllers for the tracking control problem of a quadcopter UAV. The optimization theory, summarized in the PSO technique, presents a promising solution to this kind of non-linear control problem. The scaling factors of the fuzzy GS-FPID control structure used for the studied quadcopter UAV are scheduled based on the PSO algorithm. Such a proposed systematic PSO-based approach can further improve the effectiveness of the fuzzy GS-FPID structure and eliminate the tedious and repetitive trial and error process for its designing and tuning. As described above the optimization parameters of the GS-FPID controllers are the inputs and outputs scaling factors ke, kec, Gp, Gi and Gd. These design parameters present the decision variables of the following multi-objective optimization problem. The cost functions of the optimization problem are chosen according to each controlled variable as the integral absolute error (IAE) criteria are defined by using the generic form, as follows:

$$g_i(x) = \frac{1}{2}\int (rin_i(t) - yout_i(t))^2 dt \qquad (41)$$

Then the fuzzy controller outputs can be reformulted as :

$$u_{PID}(t) = G_p \times k_p e(t) + G_i \times k_i \int_0^t e(t)dt + G_d \times k_d \frac{de(t)}{dt} \qquad (42)$$

$$u_{PD}(t) = G_p \times k_p e(t) + G_d \times k_d \frac{de(t)}{dt} \qquad (43)$$

3.3 Settings of Particle Swarm Optimization (PSO)

Similar to most optimization techniques, PSO requires a fitness evaluation function relevant to the particle's position. Let *Pbest* represent the personal best position and *Gbest* global best position of the ith particle respectively. Each particle is initialized with a random position and velocity, and it is moved toward the global best and its own personal best by updating its velocity based on the following equation :

$$V_i(t) = w \times V_i(t-1) + c_1 \times rand_1(P_{best}(t) - X_i(t)) + c_2 \times rand_2 \times (G_{best}(t) - X_i(t)) \qquad (43)$$

The position of each particle is denoted by X_i and it is represented by a set of coordinates that represents a point in the research space.

$$X_i(t+1) = X_i(t) + V_i(t+1) \qquad (44)$$

where $V_i(t)$ is the velocity of particle i at iteration t, $rand_1$ and $rand_2$ are two random numbers uniformly distributed in the interval $[0,1]$; c_1 and c_2 are the acceleration constants, and w is the inertia weight factor.

Many values of the inertia weights are tested. It is noted that the value $w = 0.8$ allows a compromise between local and global exploration (El Gmili et al. ,2017). The parameters of PSO are set based on trial and error, so that the optimal value can be obtained: The size of population $N = 50$; Number of iteration= 20 ; $c_1 = 1.2$; $c_2 = 1.2$.

Then the fitness is selected according to the design requirements. The following fitness criterion is applied.

$$f = a_1 O_p + a_2 t_s + a_3 |e_{ss}| \qquad (45)$$

where a_1, a_2, a_3 are the weighting factors and O_p, t_s and e_{ss} are the overshooot, settling time, and steady state error respectively. We have set $a_1 = 0.8$, $a_2 = 0.9$, $a_3 = 0.8$.

4. Simulation Results and Analysis

To investigate the validity and efficiency of the proposed control methods, numerical simulations have been carried out. In order to compare the auto-tuned PID vs. PD, a statistical indicator was implemented. It makes it possible to determine which one has the best behavior when following a trajectory. The main control objective is to lead $[x\ y\ z\ \phi\ \theta\ \psi]$ to the desired reference $[1\ 1\ 1\ \phi_d\ \theta_d\ 1]$. The desired angles ϕ_d and θ_d are determined according to (23). The initial position and angle values are set to zero. The research objects are slightly changed so as to emphasize the comparisons with the following simulation tests.

4.1 Network based PD/PID Control

In this part of the work, more details of the PID/PD control and Neural Network tuning method for a quadcopter UAV have been introduced. The simulation results, which verify the effectiveness of the synthesis control scheme, are shown in Figures 5 and 6.
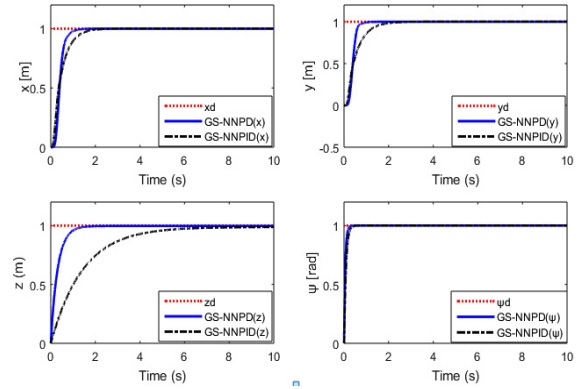

Figure 5. Quadcopter's position (x, y, z) and yaw (ψ) behaviors obtained by GS-NNPID and GS NNPD controllers
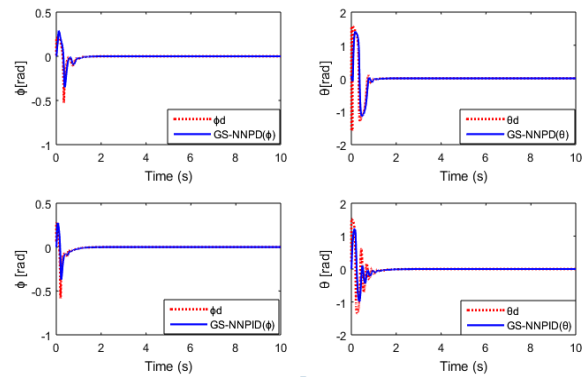

Figure 6. Quadcopter's attitude angle (ϕ, θ) behaviors obtained by GS-NNPID and GS-NNPD controllers

4.2 Fuzzy and PSO Based PD/PID Control

In this subsection, the efficiency of the fuzzy PID/PD and the optimized fuzzy PID/PD controller are analysed. The gains of the PID/PD controller are determined using the fuzzy logic method, while the PSO algorithm is introduced in order to compensate by increasing or decreasing kp, ki and kd gains, as appropriate. As it can be observed from Figures 5-8, the best performance are obtained by employing the GS-NNPD and FPID-PSO controllers. Tables 3, 4 and 5 present the statistical results of the proposed algorithms (GS-PDNN and PSO-FPID) in comparison with other approaches in terms of settling time, overshoot and mean square error (MSE). The designed GS-NNPD controller shows superior performance over the GS-NNPID, as illustrated in Figures 5 and 6.

As demonstrated in Tables 3 to 5, the GS-NNPD can enhance performance specifications if the proper integral function is not used. In stabilizing the angles and position, the GS-NNPD has shorter settling time, small MSE and without overshoot than the GS-NNPID. The responses obtained by using the FPD, FPID, optimized FPD-PSO and FPID-PSO controllers are provided in Figures 7 and 8. It is noted that the proposed FPID-PSO controller provides better system dynamic response, due to the optimized scaling factors by using the PSO algorithm.

From equations (1-6), it can be noticed that the model of quadcopter has already been integrated; therefore, the cost of placing an integrator in the controller has been an increase in the loop gain. Despite this, for the PSO based on FLC tuned PID gains an integrator is desirable since it eliminates the steady-state errors as illustrated in Figures 7 and 8.
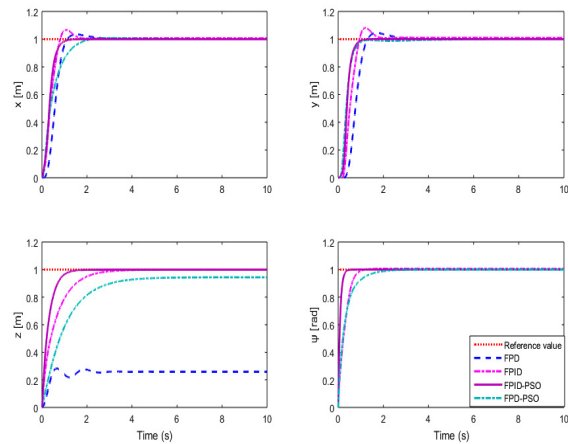

Figure 7. Quadcopter's position (x,y,z) and yaw (ψ) behaviors obtained by FPID, FPD, FPID-PSO and FPD-PSO controllers.

Table 3. Statistical results of proposed algorithms (GS-NNPD and GS-NNPID)

Type of Controller/ characteristics	GS-NNPD			GS-NNPID		
	ts(sec)	O (%)	MSE	ts(sec)	O (%)	MSE
Position(x)	0.94	0	0.0333	1.36	0	0.0299
Position(y)	0.81	0	0.0334	1.5	0	0.0350
Position(z)	0.64	0	0.0155	4.7	0	0.0735
Yaw (ψ)	0.153	0	0.0028	0.16	0	0.0029
Roll (ϕ)	--	0	0.0008	--	0	0.0009
Pitch(θ)	--	0	0.0327	--	0	0.0359

Table 4. Statistical result of proposed algorithms (FPD and FPID)

Type of Controller/ characteristics	FPD			FPID		
	ts(sec)	O (%)	MSE	ts(sec)	O (%)	MSE
Position(x)	0.9865	3.4159	0.0460	1.2283	6.1182	0.0278
Position(y)	1.1795	4.4788	0.0652	1.4105	7.2642	0.0438
Position(z)	2.0459	9.3707	0.5624	1.9974	0	0.028
Yaw (ψ)	1.2130	0	0.0155	0.7525	0.0773	0.0155
Roll (ϕ)	--	--	0.0011	--	--	0.0001
Pitch(θ)	--	--	0.0144	--	--	0.0041

Table 5. Statistical result of proposed algorithms (FPD-PSO and FPID-PSO)

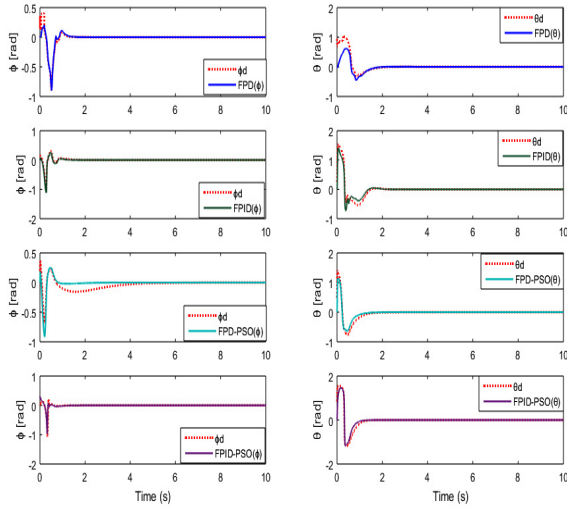| Type of Controller/ characteristics	FPD-PSO			FPID-PSO		
	ts(sec)	O (%)	MSE	ts(sec)	O (%)	MSE
Position(x)	0.7171	1.7184	0.0029	0.7714	0	0.0247
Position(y)	0.8035	0.3817	0.0101	0.5139	0	0.0328
Position(z)	1.2886	4.1832	0.0247	0.9142	0	0.0155
Yaw (ψ)	0.5867	0	0.0044	0.2225	0	0.0046
Roll (ϕ)	--	--	0.0065	--	--	0.0010
Pitch(θ)	--	--	0.0055	--	--	0.0058

Figure 8. Quadcopter's attitude (ϕ, θ) behaviors obtained by FPID, FPD, FPID-PSO and FPD-PSO controllers

4.3 Trajectory Tracking

To analyze the performance of the intelligent control methods, extensive simulation tests have been performed. It can be clearly seen that while all the state variables converge to their reference values as desired in a minimum settling time. It has been demonstrated that the synthesis control methods based on the GS-NNPD and optimized FPID-PSO are more reliable and effective approaches when performing the tracking control for the quadcopter UAV.

4.3.1 Circle Trajectory Tracking

First, consider that the quadcopter tracks a circle trajectory given by (46).

$$\begin{cases} \omega = 2\pi / 20 \\ x_d = 2\cos(\omega t) - 1 \\ y_d = 2\sin(\omega t) \\ z_d = \cos(\omega t) \\ \psi_d = 1\,rad \end{cases} \tag{46}$$

Table 6 shows the parameters of GS-NNPD controller and FPID-PSO scaling factors, with the initial conditions set to zero.

In this trajectory, the quadcopter makes a complete circle. In the same time, the orientation of the desired trajectory is *0 rad* for roll and pitch angles and *1 rad* for yaw angle.
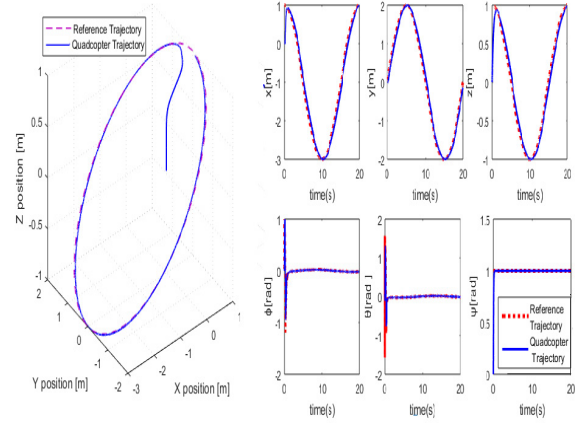

Figure 9. Circle trajectory, position and attitude of quadcopter in the closed-loop using GS-NNPD
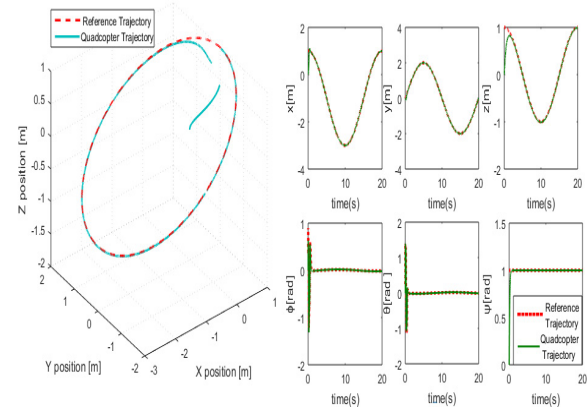

Figure 10. Circle trajectory, position and attitude of quadrotor in the closed-loop using FPID-PSO

Figures 9-10 show the tracking of the desired trajectory with precision at the end of the movement of all variables the altitude *z*, the translation motions *x* and *y*, and the yaw angle. A proper selection of the input and output SFs is necessary for the satisfactory operation of the

Table 6: PD parameters and SFs for circle trajectory using GS-NNPD and FPID-PSO

	GS-NNPD			FPID-PSO			
	kp	*kd*	*ke*	*kde*	*Gp*	*Gi*	*Gd*
Position(x)	75.8745	19.3646	0.165	0.113	5.044	0.361	3.752
Position(y)	76.7839	17.4088	0.268	0.066	4.261	0.711	2.871
Position(z)	1.8437e+03	562.8856	0.375	1.460	2.771	1.583	1.560
Yaw (ψ)	102.5617	5.3561	0.270	0.127	0.612	0.143	0.175
Roll (ϕ)	96.4076	4.1119	0.324	0.055	2.576	0.111	0.205
Pitch(θ)	97.5106	4.2620	0.355	0.101	1.302	0.935	0.336

FPID-PSO controllers. Accordingly, they are tuned by using PSO algorithm.

4.3.2 Spiral Trajectory Tracking

For a better investigation of the effectiveness of the proposed controllers, a spiral trajectory is considered as follows:

$$\begin{cases} \omega = 2\pi / 6 \\ x_d = \cos(\omega t) \\ y_d = \sin(\omega t) \\ z_d = 3 + \dfrac{t}{4} \\ \psi_d = 1 rad \end{cases}$$

(47)

Applying the same controllers, the 3D trajectory, position tracking and attitude stabilization are depicted in Figures 11 and 12 respectively. Table 7 presents the control parameters found for the second trajectory. The obtained results show also how desired attitude angles alter while the quadrotor UAV tracks the desired trajectory. As it can be observed, the tracking of variables x, y, z illustrated that both controllers drive the system towards the prescribed trajectory. The proposed methods have a higher level of accuracy compared to those presented in (Salehfard, 2018) and (Xiong & Zheng, 2014).
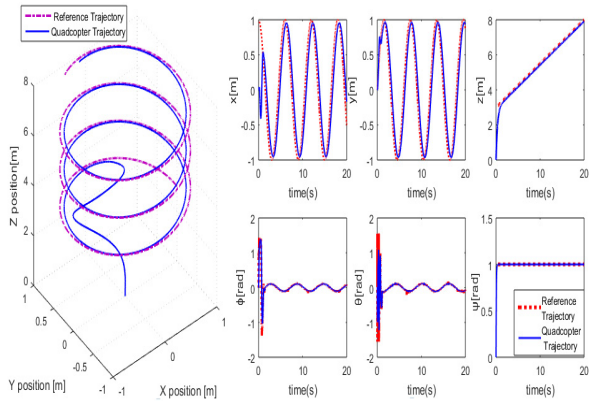

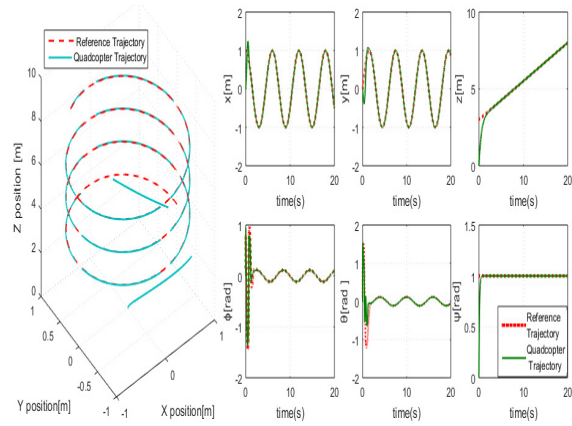**Figure 11.** Spiral trajectory, position and attitude of quadrotor in the closed-loop using GS-NNPD

Figure 12. Spiral trajectory, position and attitude of quadrotor in the closed-loop using FPID-PSO

4.4 Robustness against Disturbances

White Gaussian noise disturbance is introduced in the measured controller variables x, y and z.
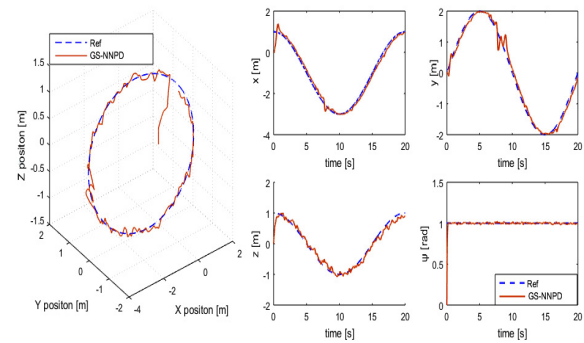

Figure 13. Circle trajectory, position and yaw angle (ψ) of quadcopter under disturbances using GS-NNPD

Figures 13-18 illustrate the position, attitude, and 3D flight trajectory of quadcopter UAV under noisy measurements.

The effectiveness and performance of the proposed scheme are validated on different trajectories against disturbance. The following simulation results demonstrate good trajectory tracking for

Table 7: PD parameters and SFs for spirale trajectory using GS-NNPD and FPID-PSO

	GS-NNPD		FPID-PSO				
	k_p	k_d	ke	kde	G_p	G_i	G_d
Position (x)	2.3749e+03	787.7987	0.392	0.172	4.400	0.595	1.584
Position(y)	41.2843	14.0504	0.074	0.094	2.337	0.846	1.484
Position(z)	90.9440	25.9208	0.163	0.604	4.255	0.367	4.454
Yaw (ψ)	119.3270	7.3105	0.135	0.114	0.714	0.126	
Roll (ϕ)	74.7289	5.9024	0.288	0.127	1.895	0.015	0.460
Pitch(θ)	118.9664	3.2114	0.368	0.044	2.640	0.091	0.647

x, y, and z displacements as well as for attitude (ϕ, θ,ψ) stabilization. Therefore, it is observed that, when the disturbance affects the quadcopter, the GS-NNPD and FPID-PSO controllers correct these undesirable effects.

So,it is noticeable that the performance shows not only the stability robustness against disturbance but also the fast response and excellent tracking capacity.
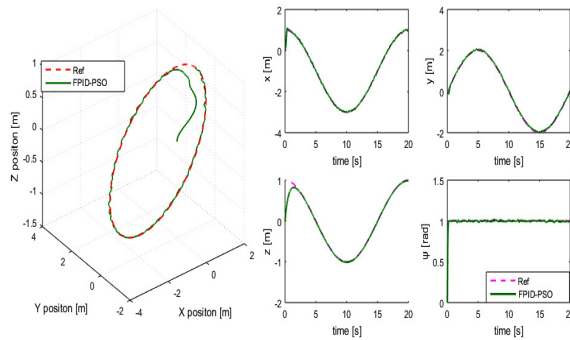

Figure 14. Circle trajectory, position and yaw angle (ψ) of quadcopter under disturbances using FPID-PSO
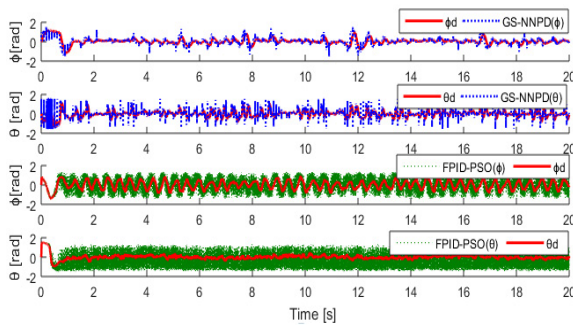

Figure 15. Quadcopter's attitude angle (ϕ and θ) stabilization under disturbances using GS-NNPD and FPID-PSO (circle trajectory)
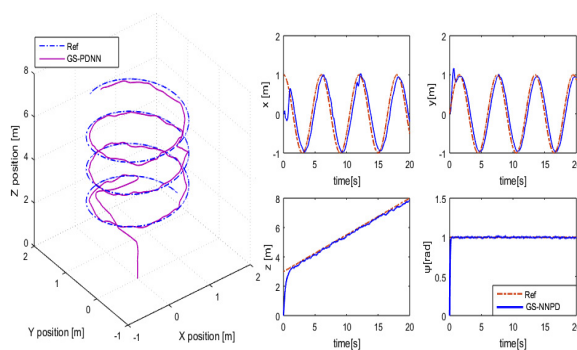

Figure 16. Spiral trajectory, position and attitude of quadcopter using GS-NNPD under disturbances
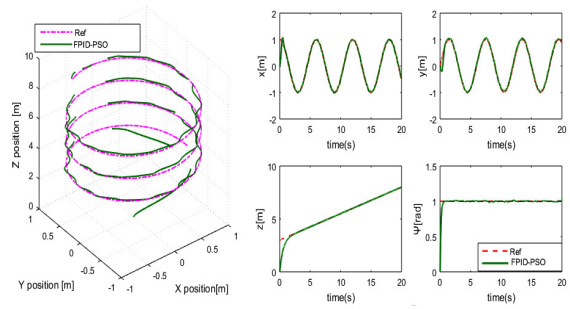

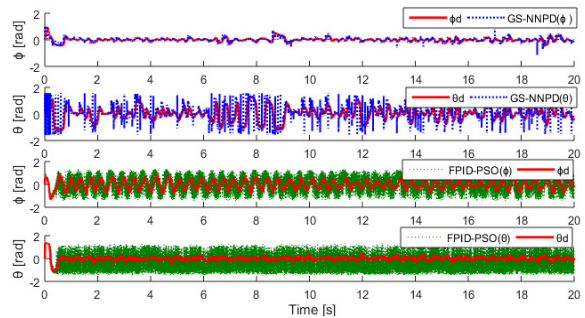**Figure 17.** Spiral trajectory, position and attitude of quadcopter using FPID-PSO under disturbances

Figure 18. Quadcopter's attitude angle (ϕ and θ) under disturbances using GS-NNPD and FPID-PSO (spiral trajectory)

5. Conclusion

This work presents the design and development of a PID/PD controller based on intelligent techniques in order to adress the problem of attitude dynamics stabilization and position tracking control for a quadcopter. Firstly, an adaptive neural network gain scheduling PID/PD controller (GS-NNPID/PD) has been developed. An improved backpropagation algorithm has been used to update the weight of this network. Secondly, the proposed control system which combines fuzzy logic and PID/PD controller has been implemented. The fuzzy logic approach is derived to schedule the control parameters adaptively according to the fuzzy rules. The PSO algorithm has been employed to determine the scaling factors of the FPID/PD-PSO controller. The validity of the proposed control algorithm is proven using different computer simulations. A comparison study based on statistical analysis has been introduced to evaluate the performance of each controller. It is concluded that the GS-NNPD and FPID PSO permit faster response, better performance, and better robustness.

REFERENCES

1. Basri, M. (2018). Design and application of an adaptive backstepping sliding mode controller for a six DOF quadrotor aerial robot, *Robotica*, *36*(11), 1701-1727.

2. Bojja, P., Raja, P. & Sumit. K. (2019). Design and Development of Robust Control Strategy using Soft Computing Techniques for Non-Linear System, *Journal of Engineering Technology*, *8*(1), 223-236.

3. Çakir, O. & Yüksel, T. (2017). Neural Network Control for Quadrotors, *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, *31*(1), 191-200.

4. Chen, Y., He, Y. & Zhou, M (2014). Designing Fuzzy PID Controller for Quadcopter, *International Journal of Advanced Research in Computer Science and Technology*, *2*(4), 221-227.

5. El Gmili, N., Mjahed, M., El Kari, A. & Ayad. H. (2017). An Improved Particle Swarm Optimization (IPSO) Approach for Identification and Control of Stable and Unstable Systems, *International Review of Automatic Control (IREACO), 10*(3), 229-239.

6. El Hamidi, K., Mjahed, M., El Kari, A. & Ayad, H. (2018). Neural and Fuzzy Based Nonlinear Flight Control for an Unmanned Quadcopter, *International Review of Automatic Control (IREACO)*, *11*(3), DOI: /10.15866/ireaco.v11i3.14055 1974-6059.

7. El Hamidi, K., Mjahed, M., El Kari, A. & Ayad, H. (2019). Quadcopter attitude and altitude tracking by using improved PD controllers, *International Journal of Nonlinear Dynamics and Control*, *1*(3), 287-303.

8. Estevez, J., Lopez-Guede, J. M. & Graa, M. (2016). Particle Swarm Optimization Quadrotor Control for Cooperative Aerial Transportation of Deformable Linear Objects, *Cybernetics and Systems*, *47*(1-2), 4-16.

9. Gül, G. & Arısoy, A. (2013). Fuzzy logic controller design for quadrotor in real time, *Journal of Aeronautics and Space Technologies*, *6*(2), 59-67.

10. Harik, E. H. C., Gurin, F., Guinand, F., Breth, J. F, Pelvillain, H. & Pard, J.-Y. (2017). Fuzzy logic controller for predictive vision-based target tracking with an unmanned aerial vehicle, *Advanced Robotics*, *31*(7), 368-381.

11. Kuantama, E., Vesselenyi, T., Dzitac, S. & Tarca, R. (2017). PID and Fuzzy-PID Control Model for Quadcopter Attitude with Disturbance Parameter, *International Journal of Computers Communications and Control*, *12*(4), 519-532.

12. Lei, X., Ge, S. & Fang, J. (2014). Adaptive neural network control of small unmanned aerial rotorcraft*, Journal of Intelligent and Robotic Systems*, *75*(2), 331-341.

13. Mardan, M., Esfandiari, M. & Sepehri, N. (2017). Attitude and position controller design and implementation, for a quadrotor, *International Journal of Advanced Robotic Systems*, *14*(3), 1-11.

14. Muliadi, J. & Kusumoputro, B. (2018). Neural Network Control System of UAV Altitude Dynamics and Its Comparison with the PID Control System, *Journal of Advanced Transportation*, *Volume 2018*(2-2), 1-18. Article ID: 3823201.

15. Rendón, M. A. & Martins, F. F. (2017). Path Following Control Tuning for an Autonomous Unmanned Quadrotor Using Particle Swarm Optimization, *IFAC-Papers On Line*, *50*(1), 325-330.

16. Ryoo, Y. J. (2017). An Autonomous Control of Fuzzy-PD Controller for Quadcopter, *International Journal of Fuzzy Logic and Intelligent Systems*, *17*(2), 107-113.

17. Salehfard, S, Abdollahi, T, Xiong, C. H & Ai, Y. H. (2018). An Optimized Fuzzy-Padé Controller Applied to Attitude Stabilization of a Quadrotor, *International Journal of Control, Automation and Systems*, *16*(3), 1425-1434.

18. Xiong, J. J. & Zheng, E. H. (2014). Position and attitude tracking control for a quadrotor UAV, *ISA Transactions*, *53*(3), 725-731.

19. Zheng, E. H & Xiong, J. J. & Luo, J. L. (2014). Second order sliding mode control for a quadrotor UAV, *ISA Transaction*, *53*(4), 1350-1356.