frontiers in Neurorobotics

HYPOTHESIS AND THEORY
published: 18 January 2021
doi: 10.3389/fnbot.2020.619350

Online Tuning of PID Controller Using a Multilayer Fuzzy Neural Network Design for Quadcopter Attitude Tracking Control

Daewon Park[1,2†], *Tien-Loc Le*[1,3†], *Nguyen Vu Quynh*[3], *Ngo Kim Long*[3] *and Sung Kyung Hong*[1*]

[1] *Faculty of Mechanical and Aerospace, Sejong University, Seoul, South Korea,* [2] *Department of Convergence Engineering for Intelligent Drone, Sejong University, Seoul, South Korea,* [3] *Faculty of Mechatronics and Electronics, Lac Hong University, Bien Hoa, Vietnam*

This study presents an online tuning proportional-integral-derivative (PID) controller using a multilayer fuzzy neural network design for quadcopter attitude control. PID controllers are simple but effective control methods. However, finding the suitable gain of a model-based controller is relatively complicated and time-consuming because it depends on external disturbances and the dynamic modeling of plants. Therefore, the development of a method for online tuning of quadcopter PID parameters may save time and effort, and better control performance can be achieved. In our controller design, a multilayer structure was provided to improve the learning ability and flexibility of a fuzzy neural network. Adaptation laws to update network parameters online were derived using the gradient descent method. Also, a Lyapunov analysis was provided to guarantee system stability. Finally, simulations concerning quadcopter attitude control were performed using a Gazebo robotics simulator in addition to a robot operating system (ROS), and their results were demonstrated.

Keywords: quadcopter attitude, fuzzy neural network, proportional-integral-derivative, attitude tracking control, fuzzy PID

OPEN ACCESS

Edited by:
Jose De Jesus Rubio, Instituto Politécnico Nacional (IPN), Mexico

Reviewed by:
Jiancheng Lyu, Qualcomm, United States
Yu Cao, Huazhong University of Science and Technology, China
Di Wu, Chongqing Institute of Green and Intelligent Technology (CAS), China

***Correspondence:**
Sung Kyung Hong
skhong@sejong.ac.kr

†These authors share first authorship

Received: *21 October 2020*
Accepted: *14 December 2020*
Published: *18 January 2021*

Citation:
Park D, Le T-L, Quynh NV, Long NK and Hong SK (2021) Online Tuning of PID Controller Using a Multilayer Fuzzy Neural Network Design for Quadcopter Attitude Tracking Control. Front. Neurorobot. 14:619350. doi: 10.3389/fnbot.2020.619350

INTRODUCTION

In the 4th industrial revolution era, the application of multi-copters has significantly expanded, and it has thus attracted the interest of many researchers specialized in multi-copter control engineering. Multi-copters need to maintain accurate attitudes to ensure stable flight, so the development of accurate and stable controllers for multi-copters is essential. The most popular controller used for multi-copters is the cascaded proportional-proportional integral derivative controller (PPID) (Hernandez and Frias, 2016; Salas et al., 2019; Santos et al., 2019; Xuan-Mung and Hong, 2019a,b). Since PID is a linear controller, it is generally hard to use to achieve the highest control performance for non-linear control systems (Sarabakha et al., 2017). Furthermore, in the design of a PID controller, it is necessary to obtain the exact mathematical model of the control system and to optimize the gain value to achieve the desired performance. However, this work requires complexity calculations and an accurate modeling plant.

Gain tuning for PID controllers is time-consuming, as it depends on the knowledge of dynamic plants and the experience of expert operators (Kim et al., 2007). Mathematical models are different,

as they depend on the size and appearance of multi-copters. Even if it is the same aircraft, the center of gravity can be different (Kurak and Hodzic, 2018). Therefore, the gain values are always flexible, and it may take longer to find suitable values. However, such tasks must be completed correctly, as an incorrect calculation can lead to terrible problems, such as falling or bumping into a people or objects. In past decades, many studies have been conducted to solve such problems (time-consuming, costly, etc.; Fatan et al., 2013; Kuantama et al., 2017; Noordin et al., 2017; Rouhani et al., 2017; Sumardi and Riyadi, 2017; Prayitno et al., 2018; Thanh and Hong, 2018; Chen et al., 2019; Rabah et al., 2019; Soriano et al., 2020). The Ziegler Nichols method is a well-known for tuning PID parameters (Azman et al., 2017). However, it produces oscillatory responses and yields overshoot problems (Zahir et al., 2018). Moreover, it cannot be applied in the online-tuning of parameters, and the control performance can be further improved.

In recognizing the above-mentioned limitation of traditional PID controllers, many researchers have provided methods for auto-tuning PID parameters (Amoura et al., 2016; De Keyser et al., 2016; Concha et al., 2017; Live et al., 2017; Mendes et al., 2017; Wang, 2017; Bernardes et al., 2019). In which, using the fuzzy neural network to tune PID parameter controllers has attracted the attention of many researchers. In 2018, Davanipour et al. proposed a self-tuning PID controller based on a fuzzy wavelet neural network (Davanipour et al., 2018). In 2019, Tripathy et al. introduced a fuzzy PID controller for load frequency control using spider monkey optimization (Tripathy et al., 2019). In 2017, Wang et al. provided an improved fuzzy PID controller design using predictive functional control structure (Wang et al., 2017). These studies could reduce the effort required for tuning the required parameters for PID controllers. However, most of their methods were complicated, but the control performance could be further improved.

In recent years, many studies have investigated the fuzzy neural network (FNN), which is a combination of fuzzy systems and neural networks. Therefore, combined FNN systems have the advantages of both fuzzy systems and neural networks, such as human-like reasoning and learning capabilities. Using the neural network structure, FNN parameters can be trained online to achieve suitable values. According to the above discussions, this study presents a method for auto-tuning PID parameters using multilayer fuzzy neural network (PID-MFNN) to control the attitude the quadcopters. Compared with the previous studies (Wang et al., 2017; Davanipour et al., 2018; Tripathy et al., 2019), our proposed PID-MFNN has some advantages, such as no need for offline training. The major differences and improvements among this study and related works are shown in **Table 1**. Also, the fuzzy weights and fuzzy membership functions can be updated online, and the multilayer structure is applied to improve the control performance. The main contributions of this study are summarized as follows: (1) the successful design of a multilayer structure was provided to improve the learning ability and flexibility of a FNN; (2) the adaptation laws for updating the parameters of a controller online were derived by using the gradient descent method; (3) the simulation results of controlling the attitude of a quadcopter were provided to illustrate the effectiveness of the control design method.

The rest of this paper is organized as follows. The quadcopter dynamics model and quadcopter controller are given in section Problem Formulation. The PID-MFNN structure and its learning parameter are described in section PID-MFNN Structure and Parameter Learning. The simulation results of controlling the quadcopter attitude are presented in section Simulation Results. Finally, the conclusions are given in section Conclusion.

TABLE 1 | The major differences and improvements among this study and related works.

	PID-MFNN (our proposed method)	**FWNN-PID (Davanipour et al., 2018)**	**SMO-PID (Tripathy et al., 2019)**	**PFPID (Wang et al., 2017)**
Method	Using multilayer membership function neural network to online adjust the PID controller's parameters	Using fuzzy wavelet neural network to model the system for approximation the non-linear function. Then, the PID controller's parameters are designed based on the obtained model	Using the spider monkey optimization algorithm to optimize the PID controller's parameters	Using fuzzy system combined with PID to predictive functional model and control the process. The PID controller's parameters are tuned online by fuzzy inference
Membership function	Provided the adaptation laws for online updating the membership functions	Using fixed membership functions	Using fixed membership functions	Using predefined triangular membership functions and predefined sigmoid membership functions
Fuzzy weights	Provided the adaptation laws for online adjusting the fuzzy weights	Provided the adaptation laws for online adjusting the fuzzy weights	Using fixed fuzzy weights	Using predefined fuzzy rules based on technical knowledge and engineering design
Adaptation laws	Using the steepest descent gradient approach and a backpropagation algorithm	Using the steepest descent gradient approach and a backpropagation algorithm	Not use	Not use

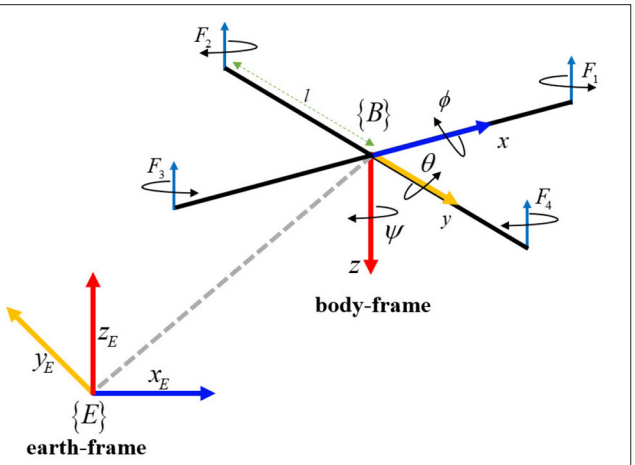

FIGURE 1 | Quadcopter schematic and its reference frames.

PROBLEM FORMULATION

Quadcopter Dynamics Model

The quadcopter schematic and its reference frames are shown in **Figure 1**. In which, the earth-frame and quadcopter body frame are denoted by $E\left(x_E,\ y_E,\ z_E\right)$ and $B\left(x,\ y,\ z\right)$. The dynamic equations of the quadcopter are given as:

$$\begin{cases} \ddot{x} = \frac{1}{M}\left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right)U_1 \\ \ddot{y} = \frac{1}{M}\left(\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi\right)U_1 \\ \ddot{z} = g - \frac{1}{M}\left(\cos\phi\cos\theta\right)U_1 \\ \ddot{\phi} = \left(\frac{J_y - J_z}{J_x}\right)\dot{\theta}\dot{\psi} + \frac{l}{J_x}U_2 \\ \ddot{\theta} = \left(\frac{J_z - J_x}{J_y}\right)\dot{\phi}\dot{\psi} + \frac{l}{J_y}U_3 \\ \ddot{\psi} = \left(\frac{J_x - J_y}{J_z}\right)\dot{\phi}\dot{\theta} + \frac{1}{J_z}U_4 \end{cases} \tag{1}$$

where θ, ϕ, and ψ stand for the three Euler angles: roll, pitch, and yaw, respectively; J_x, J_y and J_z stand for the moment of inertia for the x, y, and z axes, respectively; l denotes the length of the quadcopter's arm; U_1 denotes the total thrust on the body in the z-axis; U_2, U_3, and U_4 denote the roll, pitch, and yaw torques, respectively, which are given as

$$\begin{cases} U_1 = F_1 + F_3 + F_2 + F_4 \\ U_2 = F_4 - F_2 \\ U_3 = F_3 - F_1 \\ U_4 = C_d\left(F_1 + F_3 - F_2 - F_4\right) \end{cases} \tag{2}$$

where $F_i = C_t\Omega_i^2$ denotes the thrust of the i-th motor; Ω_i denotes the speed of the i-th motor; C_t and C_d stand for the thrust and drag coefficients, respectively.

Quadcopter Controller

The quadcopter controller scheme is given in **Figure 2**. There are two kinds of controllers that need to be considered in quadcopter control: position controller and attitude controller. The PPID attitude controller is given in **Figure 3**. The output of the outer-loop controller is given by

$$\begin{aligned} \dot{\theta}_{ref} &= G_\theta e_\theta \\ \dot{\phi}_{ref} &= G_\phi e_\phi \\ \dot{\psi}_{ref} &= G_\psi e_\psi \end{aligned} \tag{3}$$

where G_θ, G_ϕ, and G_ψ denote the proportional gain for the roll, pitch, and yaw outer-loop controller, respectively; e_θ, e_ϕ, and e_ψ are the angle errors of the roll, pitch, and yaw, respectively.

$$\begin{aligned} e_\theta &= \theta_{ref} - \theta \\ e_\phi &= \phi_{ref} - \phi \\ e_\psi &= \psi_{ref} - \psi \end{aligned} \tag{4}$$

where θ_{ref}, ϕ_{ref}, and ψ_{ref} are the angle references of the roll, pitch, and yaw, respectively.

The output of the PID inner-loop controller is given by

$$\begin{aligned} u_\theta &= K_p^\theta e_{\dot{\theta}} + K_i^\theta \int_0^t e_{\dot{\theta}} + K_d^\theta \dot{e}_{\dot{\theta}} \\ u_\phi &= K_p^\phi e_{\dot{\phi}} + K_i^\phi \int_0^t e_{\dot{\phi}} + K_d^\phi \dot{e}_{\dot{\phi}} \\ u_\psi &= K_p^\psi e_{\dot{\psi}} + K_i^\psi \int_0^t e_{\dot{\psi}} + K_d^\psi \dot{e}_{\dot{\psi}} \end{aligned} \tag{5}$$

where K_p^θ, K_i^θ, K_d^θ, K_p^ϕ, K_i^ϕ, K_d^ϕ, K_p^ψ, K_i^ψ, and K_d^ψ, are the PID gains for the roll, pitch, and yaw inner-loop controller; $e_{\dot{\theta}}$, $e_{\dot{\phi}}$, and $e_{\dot{\psi}}$ denote the angular velocity errors of the roll, pitch, and yaw, respectively.

$$\begin{aligned} e_{\dot{\theta}} &= \dot{\theta}_{ref} - \dot{\theta} \\ e_{\dot{\phi}} &= \dot{\phi}_{ref} - \dot{\phi} \\ e_{\dot{\psi}} &= \dot{\psi}_{ref} - \dot{\psi} \end{aligned} \tag{6}$$

To obtain suitable values for the PID gains in the outer-loop and inner-loop controller, the PID-MFNN was designed in the following section to adjust the gains for the PPID controller online.

PID-MFNN STRUCTURE AND PARAMETER LEARNING

PID-MFNN Structure

The fuzzy rule forms for the proposed PID-MFNN controller are:

$$\begin{aligned} &l^{th}\ \text{rule}: \ IF\ i_1\ is\ \mu_{1jk}\ and\ i_2\ is\ \mu_{2jk}, ..., \ and\ i_{n_i}\ is\ \mu_{n_ijk} \\ &\qquad Then\ out = w_{klo} \\ &i = 1, 2, ..., n_i;\ \ j = 1, 2, ..., n_j; \\ &k = 1, 2, ..., n_k;\ \ l = 1, 2, ..., n_l;\ \ o = 1, 2, ..., n_o \end{aligned} \tag{7}$$

where i_i is the fuzzy input; μ_{ijk} is the membership function (MF) for the input i-th, block j-th, and layer k-th; w_{klo} is the fuzzy weight for the layer k-th, rule l-th, and output o-th.

The auto-tuning algorithm structure of the PID-MFNN controller is shown in **Figure 4** in which, the MFNN structure is applied to adjust the PPID controller gains online. As shown in **Figure 5**, the MFNN structure includes five spaces as follows:

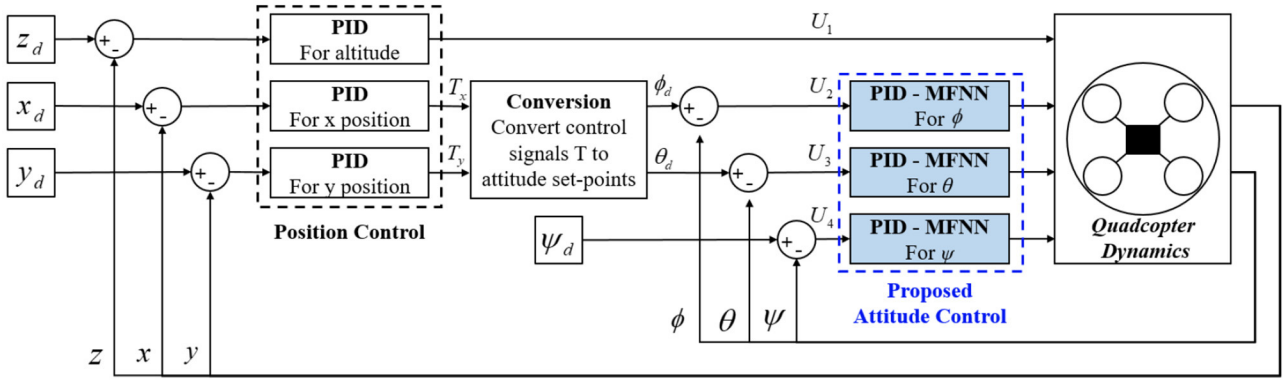

FIGURE 2 | Quadcopter controller scheme.

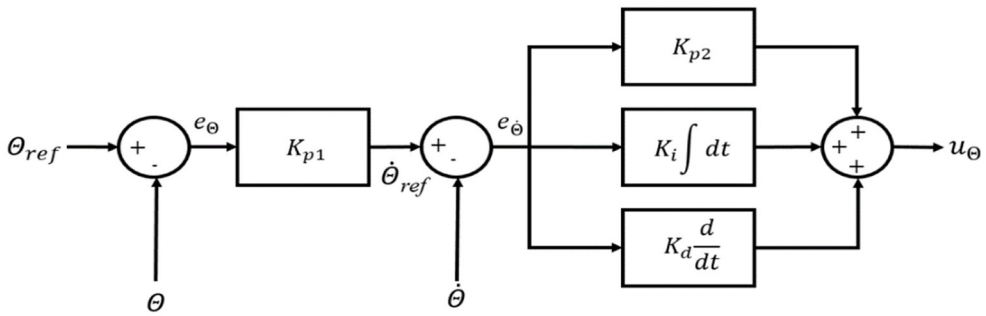

FIGURE 3 | Control scheme of the quadcopter attitude controller (Θ is replaced for ϕ, θ, or ψ).

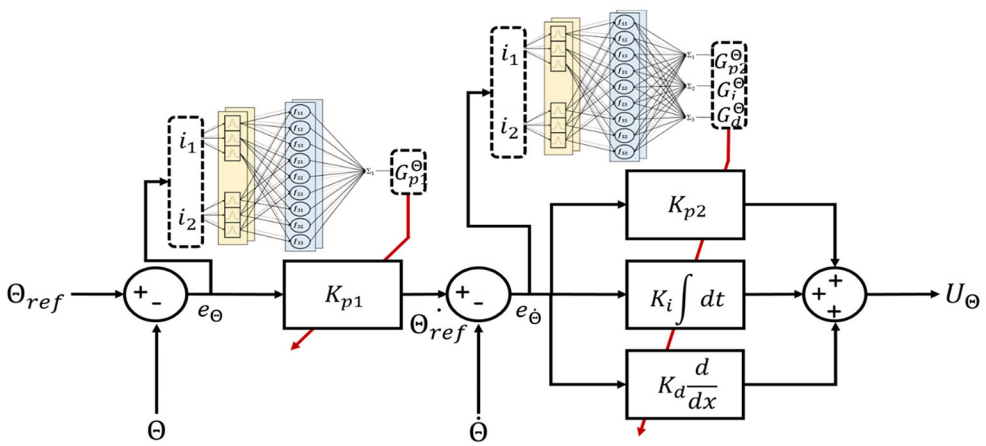

FIGURE 4 | Auto-tuning algorithm structure of the PID-MFNN controller.

1) The input space: the fuzzy input vector, $I = [i_1, i_2, ..., i_{n_i}]$, is prepared in this space. Each node corresponds to one input and is directly transferred to the next space.
2) The membership function space: the membership grades are calculated in this space using the fuzzy inputs and the Gaussian membership functions (GMFs). As shown in **Figure 6**, the multilayer fuzzy membership function is provided to improve the learning ability and flexibility of the FNN. The outputs of this space are computed as:

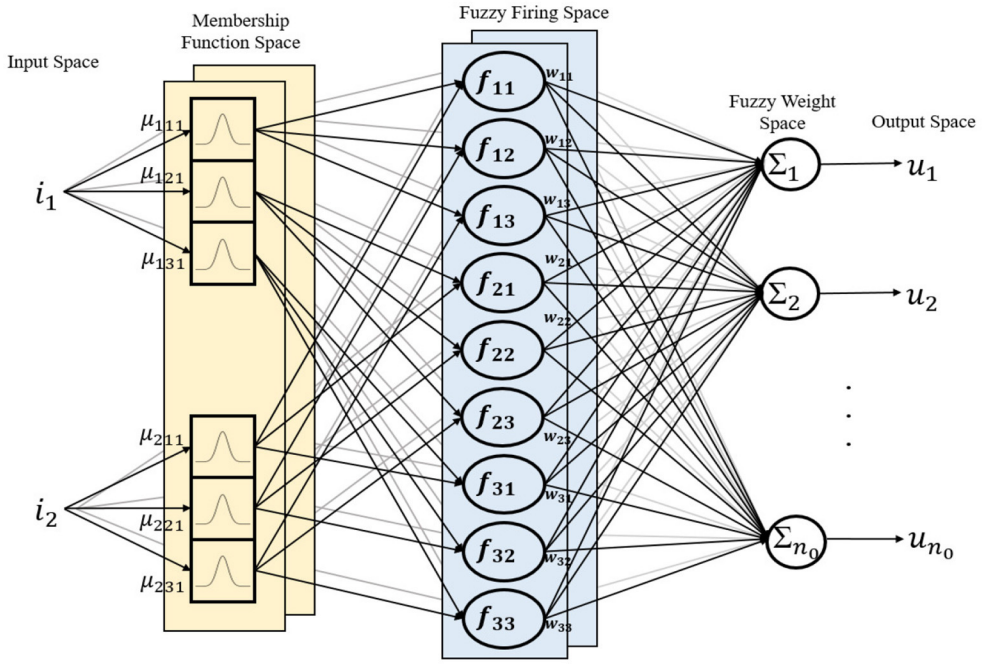

FIGURE 5 | Multilayer fuzzy neural network structure.

$$\mu_{ijk} = \exp\left\{\frac{-(i_i - m_{ijk})^2}{v_{ijk}^2}\right\} \tag{8}$$

where m_{ijk} and v_{ijk} are the mean and variance of the GMFs.

3) The fuzzy firing space: the fuzzy firing strength is calculated in this space using the membership grades as follows:

$$f_{kl} = \prod_{i=1}^{n_i} \mu_{ijk} \tag{9}$$

where f_{kl} is the fuzzy firing strength for the l-th rule in the k-th layer.

Then, the fuzzy firing vector is presented as

$$\begin{bmatrix} f_{1l} \\ f_{2l} \\ \vdots \\ f_{n_k l} \end{bmatrix} = \begin{bmatrix} f_{11}, f_{12}, ..., f_{1n_l} \\ f_{21}, f_{22}, ..., f_{2n_l} \\ \vdots \ \vdots \ \vdots \\ f_{n_k 1}, f_{n_k 2}, ..., f_{n_k n_l} \end{bmatrix} \tag{10}$$

4) The fuzzy weight space: in this space, the fuzzy weights w_{klo} are provided to connect the fuzzy firing space and output space. The fuzzy weight vector for the o-th output is presented as

$$\begin{bmatrix} w_{1lo} \\ w_{2lo} \\ \vdots \\ w_{n_k lo} \end{bmatrix} = \begin{bmatrix} w_{11o}, w_{12o}, ..., w_{1n_l o} \\ w_{21o}, w_{22o}, ..., w_{2n_l o} \\ \vdots \ \vdots \ \vdots \\ w_{n_k 1o}, w_{n_k 2o}, ..., w_{n_k n_l o} \end{bmatrix} \tag{11}$$

5) The output space: the final outputs of this space are the PID gains for the outer-loop and inner-loop controllers. Defuzzification is performed using the product operation as follows:

$$u_o = \sum_{k=1}^{n_k} \left(\frac{\sum_{l=1}^{n_l} (f_{kl} w_{klo})}{\sum_{l=1}^{n_l} f_{kl}} \right) \tag{12}$$

Using the proposed MFNN to adjust the PID gains online, the desired performance of the control network can be achieved. The adaptation laws for updating the parameters of the proposed PID-MFNN controller online are explained in the following section.

Parameter Learning

First, considering the outer-loop controller, Equation (3) can be rewritten, where the gains G_θ, G_ϕ, and G_ψ are obtained using the output of the proposed MFNN in Equation (10).

$$\dot{\Theta}_{ref} = G_\Theta e_\Theta \tag{13}$$

$$G_\Theta = \sum_{k=1}^{n_k} \left(\frac{\sum_{l=1}^{n_l} (f_{kl} w_{kl\Theta}^g)}{\sum_{l=1}^{n_l} f_{kl}} \right) \tag{14}$$

where Θ is replaced by (θ, ϕ, ψ).

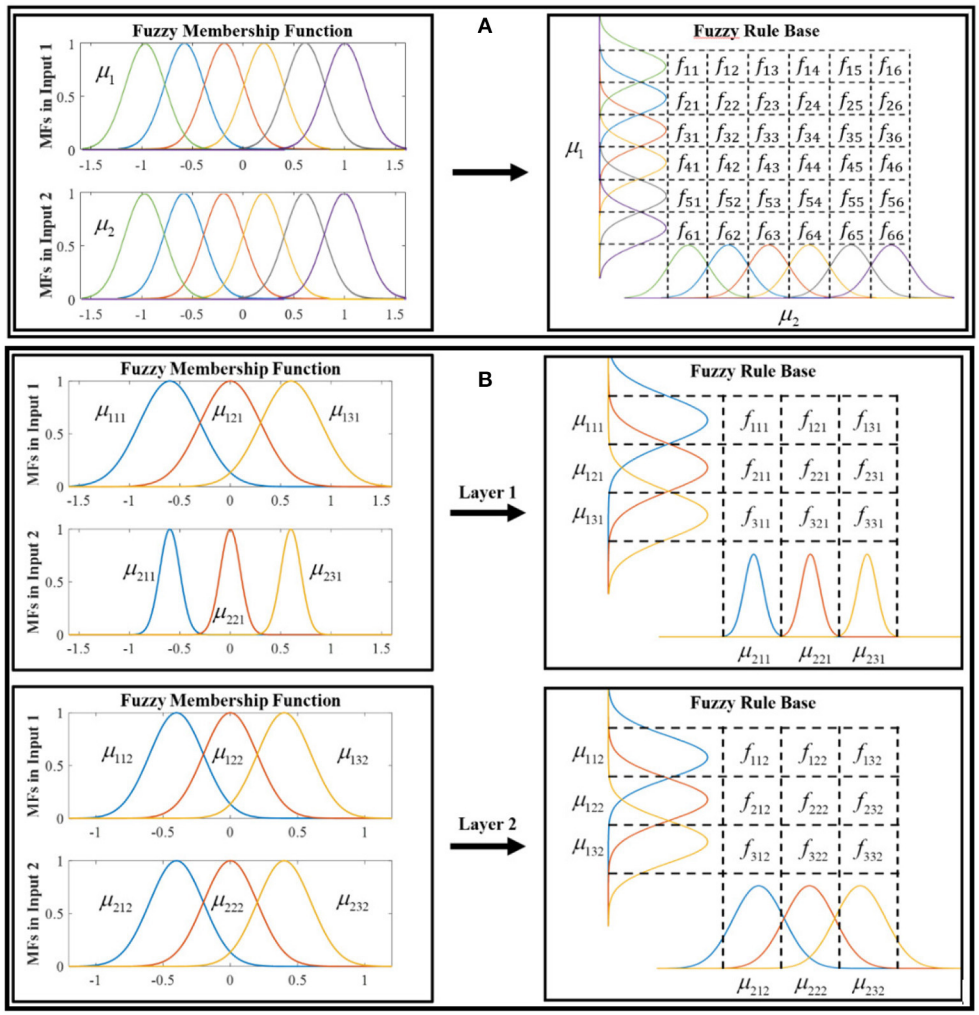

FIGURE 6 | Fuzzy membership function and rule base: **(A)** the conventional fuzzy system and **(B)** the proposed multilayer fuzzy neural network with two layers.

By rewriting the Equation (6),

$$e_{\dot{\Theta}} = \dot{\Theta}_{ref} - \dot{\Theta} \tag{15}$$

A Lyapunov function is defined as $E_1(t) = \frac{1}{2}\left(e_{\dot{\Theta}}(t)\right)^2$, using the gradient descent approach and chain rule, and the parameter update laws for PID-MFNN can be obtained as follows:

$$\begin{aligned}
\hat{w}^g_{kl\Theta}(t+1) &= \hat{w}^g_{kl\Theta}(t) - \eta^g_w \frac{\partial E_1}{\partial \hat{w}^g_{kl\Theta}} \\
&= \hat{w}^g_{kl\Theta}(t) - \eta^g_w \frac{\partial E_1}{\partial e_{\dot{\Theta}}} \frac{\partial e_{\dot{\Theta}}}{\partial \dot{\Theta}_{ref}} \frac{\partial \dot{\Theta}_{ref}}{\partial G_\Theta} \frac{\partial G_\Theta}{\partial \hat{w}^g_{kl\Theta}} \\
&= \hat{w}^g_{kl\Theta}(t) - \eta^g_w e_{\dot{\Theta}} e_\Theta \frac{f_{kl}}{\sum_{l=1}^{n_l} f_{kl}}
\end{aligned} \tag{16}$$

$$\begin{aligned}
\hat{m}^g_{ijk}(t+1) &= \hat{m}^g_{ijk}(t) - \eta^g_m \frac{\partial E_1}{\partial \hat{m}^g_{ijk}} \\
&= \hat{m}^g_{ijk}(t) - \eta^g_m \frac{\partial E_1}{\partial e_{\dot{\Theta}}} \frac{\partial e_{\dot{\Theta}}}{\partial \dot{\Theta}_{ref}} \frac{\partial \dot{\Theta}_{ref}}{\partial G_\Theta} \frac{\partial G_\Theta}{\partial f_{kl}} \frac{\partial f_{kl}}{\partial \mu_{ijk}} \frac{\partial \mu_{ijk}}{\partial \hat{m}^g_{ijk}} \\
&= \hat{m}^g_{ijk}(t) - \eta^g_m e_{\dot{\Theta}} e_\Theta \frac{\left(\hat{w}^g_{kl\Theta} - G_\Theta\right)}{\sum_{l=1}^{n_l} f_{kl}} f_{kl} \frac{\left(i_i - \hat{m}^g_{ijk}\right)}{\left(\hat{v}^g_{ijk}\right)^2}
\end{aligned} \tag{17}$$

$$\begin{aligned}
\hat{v}^g_{ijk}(t+1) &= \hat{v}^g_{ijk}(t) - \eta^g_v \frac{\partial E_1}{\partial \hat{v}^g_{ijk}} \\
&= \hat{v}^g_{ijk}(t) - \eta^g_v \frac{\partial E_1}{\partial e_{\dot{\Theta}}} \frac{\partial e_{\dot{\Theta}}}{\partial \dot{\Theta}_{ref}} \frac{\partial \dot{\Theta}_{ref}}{\partial G_\Theta} \frac{\partial G_\Theta}{\partial f_{kl}} \frac{\partial f_{kl}}{\partial \mu_{ijk}} \frac{\partial \mu_{ijk}}{\partial \hat{v}^g_{ijk}} \\
&= \hat{v}^g_{ijk}(t) - \eta^g_v e_{\dot{\Theta}} e_\Theta \frac{\left(\hat{w}^g_{kl\Theta} - G_\Theta\right)}{\sum_{l=1}^{n_l} f_{kl}} f_{kl} \frac{\left(i_i - \hat{m}^g_{ijk}\right)^2}{\left(\hat{v}^g_{ijk}\right)^3}
\end{aligned} \tag{18}$$

where $\hat{w}^g_{kl\Theta}$, $\hat{m}^g_{ijk}$, and $\hat{v}^g_{ijk}$ are the connecting weight, mean, and variance of the outer-loop MFNN controller; η^g_w, η^g_m, and η^g_v are the positive learning rates.

Then, considering the inner-loop controller, Equation (5) can be rewritten, where the gains K^θ_p, K^θ_i, K^θ_d, K^ϕ_p, K^ϕ_i, K^ϕ_d, K^ψ_p, K^ψ_i, and K^ψ_d are obtained using the outputs of the proposed MFNN.

$$u^\Theta = u^\Theta_p + u^\Theta_i + u^\Theta_d = K^\Theta_p G^\Theta_p + K^\Theta_i G^\Theta_i + K^\Theta_d G^\Theta_d \quad (19)$$

where $G^\Theta_p = e_{\dot{\Theta}}$; $G^\Theta_i = \int_0^t e_{\dot{\Theta}}$; $G^\Theta_d = \dot{e}_{\dot{\Theta}}$

$$K^\Theta_p = \sum_{k=1}^{n_k}\left(\frac{\sum_{l=1}^{n_l}\left(f_{kl}w^p_{kl\Theta}\right)}{\sum_{l=1}^{n_l} f_{kl}}\right); K^\Theta_i = \sum_{k=1}^{n_k}\left(\frac{\sum_{l=1}^{n_l}\left(f_{kl}w^i_{kl\Theta}\right)}{\sum_{l=1}^{n_l} f_{kl}}\right);$$
$$K^\Theta_d = \sum_{k=1}^{n_k}\left(\frac{\sum_{l=1}^{n_l}\left(f_{kl}w^d_{kl\Theta}\right)}{\sum_{l=1}^{n_l} f_{kl}}\right) \quad (20)$$

Then

$$K^\Theta_\alpha = \sum_{k=1}^{n_k}\left(\frac{\sum_{l=1}^{n_l}\left(f_{kl}w^\alpha_{kl\Theta}\right)}{\sum_{l=1}^{n_l} f_{kl}}\right); u^\Theta_\alpha = K^\Theta_\alpha G^\Theta_\alpha \quad (21)$$

where α is replaced by (p, i, d).

By rewriting the Equation (4),

$$e_\Theta = \Theta_{ref} - \Theta \quad (22)$$

A Lyapunov function can be defined as $E_2(t) = \frac{1}{2}(e_\Theta(t))^2$, using the gradient descent approach and chain rule, and the parameter update laws for PID-MFNN can be obtained as:

$$\begin{aligned}\hat{w}^\alpha_{kl\Theta}(t+1) &= \hat{w}^\alpha_{kl\Theta}(t) - \eta^\alpha_w \frac{\partial E_2}{\partial \hat{w}^\alpha_{kl\Theta}}\\ &= \hat{w}^\alpha_{kl\Theta}(t) - \eta^\alpha_w \frac{\partial E_2}{\partial e_\Theta}\frac{\partial e_\Theta}{\partial \Theta}\frac{\partial \Theta}{\partial u^\Theta_\alpha}\frac{\partial u^\Theta_\alpha}{\partial K^\Theta_\alpha}\frac{\partial K^\Theta_\alpha}{\partial \hat{w}^\alpha_{kl\Theta}}\\ &= \hat{w}^\alpha_{kl\Theta}(t) + \eta^\alpha_w e_\Theta T_\Theta G^\Theta_\alpha \frac{f_{kl}}{\sum_{l=1}^{n_l} f_{kl}} \end{aligned} \quad (23)$$

$$\begin{aligned}\hat{m}^\alpha_{ijk}(t+1) &= \hat{m}^\alpha_{ijk}(t) - \eta^\alpha_m \frac{\partial E_2}{\partial \hat{m}^\alpha_{ijk}}\\ &= \hat{m}^\alpha_{ijk}(t) - \eta^\alpha_m \frac{\partial E_2}{\partial e_\Theta}\frac{\partial e_\Theta}{\partial \Theta}\frac{\partial \Theta}{\partial u^\Theta_\alpha}\frac{\partial u^\Theta_\alpha}{\partial K^\Theta_\alpha}\frac{\partial K^\Theta_\alpha}{\partial f_{kl}}\frac{\partial f_{kl}}{\partial \mu_{ijk}}\frac{\partial \mu_{ijk}}{\partial \hat{m}^\alpha_{ijk}}\\ &= \hat{m}^\alpha_{ijk}(t) + \eta^\alpha_m e_\Theta T_\Theta G^\Theta_\alpha \frac{\left(\hat{w}^\alpha_{kl\Theta} - K^\Theta_\alpha\right)}{\sum_{l=1}^{n_l} f_{kl}} f_{kl}\frac{\left(i_i - \hat{m}^\alpha_{ijk}\right)}{\left(\hat{v}^\alpha_{ijk}\right)^2} \end{aligned} \quad (24)$$

$$\begin{aligned}\hat{v}^\alpha_{ijk}(t+1) &= \hat{v}^\alpha_{ijk}(t) - \eta^\alpha_v \frac{\partial E_2}{\partial \hat{v}^\alpha_{ijk}}\\ &= \hat{v}^\alpha_{ijk}(t) - \eta^\alpha_v \frac{\partial E_2}{\partial e_\Theta}\frac{\partial e_\Theta}{\partial \Theta}\frac{\partial \Theta}{\partial u^\Theta_\alpha}\frac{\partial u^\Theta_\alpha}{\partial K^\Theta_\alpha}\frac{\partial K^\Theta_\alpha}{\partial f_{kl}}\frac{\partial f_{kl}}{\partial \mu_{ijk}}\frac{\partial \mu_{ijk}}{\partial \hat{v}^\alpha_{ijk}}\\ &= \hat{v}^\alpha_{ijk}(t) + \eta^\alpha_v e_\Theta T_\Theta G^\Theta_\alpha \frac{\left(\hat{w}^\alpha_{kl\Theta} - K^\Theta_\alpha\right)}{\sum_{l=1}^{n_l} f_{kl}} f_{kl}\frac{\left(i_i - \hat{m}^\alpha_{ijk}\right)^2}{\left(\hat{v}^\alpha_{ijk}\right)^3} \end{aligned} \quad (25)$$

where $\hat{w}^\alpha_{kl\Theta}$, $\hat{m}^\alpha_{ijk}$, and $\hat{v}^\alpha_{ijk}$ are the connecting weight, mean, and variance of the inner-loop MFNN controller; T_Θ can be T_ϕ, T_θ, or T_Ψ, which are the quadcopter transfer functions for the yaw, pitch and roll angles; η^α_w, η^α_m, and η^α_v are the positive learning rates.

Using the proposed adaptation laws in (16–18) and (23–25), the parameters of the proposed PID-MFNN can be obtained. Then, the suitable gains for the outer-loop and inner-loop PPID controller can be achieved.

The Convergence Analysis

The Lyapunov cost function is defined as

$$V(t) = E_1 = \frac{1}{2}\left(e_{\dot{\Theta}}(t)\right)^2 \quad (26)$$

Then,

$$\Delta V(t) = V(t+1) - V(t) = \frac{1}{2}\left[\left(e_{\dot{\Theta}}(t+1)\right)^2 - \left(e_{\dot{\Theta}}(t)\right)^2\right] \quad (27)$$

By applying the Taylor expansion,

$$e_{\dot{\Theta}}(t+1) = e_{\dot{\Theta}}(t) + \Delta e_{\dot{\Theta}}(t) \cong e_{\dot{\Theta}}(t) + \left[\frac{\partial e_{\dot{\Theta}}(t)}{\partial \hat{w}^g_{kl\Theta}}\right]\Delta \hat{w}^g_{kl\Theta} \quad (28)$$

From (16),

$$\frac{\partial e_{\dot{\Theta}}(t)}{\partial \hat{w}^g_{kl\Theta}} = e_\Theta \frac{f_{kl}}{\sum_{l=1}^{n_l} f_{kl}} = \zeta \quad (29)$$

By rewriting (28) using (29) and (16),

$$e_{\dot{\Theta}}(t+1) = e_{\dot{\Theta}}(t) - \zeta\left(\eta^g_w e_{\dot{\Theta}}(t)\zeta\right) = e_{\dot{\Theta}}(t)\left[1 - \eta^g_w \zeta^2\right] \quad (30)$$

By rewriting (27) using (30),

$$\begin{aligned}\Delta V(t) &= \frac{1}{2}\left(e_{\dot{\Theta}}(t)\right)^2\left[\left(1 - \eta^g_w\zeta^2\right)^2 - 1\right]\\ &= \frac{1}{2}\left(e_{\dot{\Theta}}(t)\right)^2\left[\left(\eta^g_w\zeta^2\right)^2 - 2\eta^g_w\zeta^2\right]\\ &= \frac{1}{2}\eta^g_w\left(e_{\dot{\Theta}}(t)\right)^2\zeta^2\left(\eta^g_w\zeta^2 - 2\right)\end{aligned} \quad (31)$$

From (31), if η_w^g is chosen to satisfy $0 < \eta_w^g < \frac{2}{\zeta^2}$, then $\Delta V(t) < 0$. Consequently, the convergence of the adaptation law is guaranteed by the Lyapunov theorem. Similarly, the convergence of the adaptation laws for η_w^α, η_m^α, and η_v^α can be obtained.

The computational complexity of our approaches using big O notation is given as:

+ Big O notation for PID-CFNN:

$$Big - O = O\left(T * \left[\max\left(n_1, n_2\right) + n_1 * n_2\right]\right) \tag{32}$$

+ Big O notation for PID-MFNN:

$$Big - O = O\left(T * n_k\left[\max\left(n_{j_1}, n_{j_2}\right) + \left(n_{j_1} * n_{j_2}\right)\right]\right) \tag{33}$$

where:

T: is the simulation time.
n_{j_1}: the number of membership functions in input 1.
n_{j_2}: the number of membership functions in input 2.
n_k: the number of layers.

In our proposed network, we choose $n_{j_1} = n_{j_2} = 3$ and $n_k = 2$.

SIMULATION RESULTS

This section presents the results of the quadcopter attitude control using the proposed PID-MFNN controller, which is performed using the Gazebo robotics simulator and ROS. The Gazebo simulation is a multi-robot simulator for outdoor environments. It supports the interface of the quadcopter's firmware, Pixhawk 4 (PX4). Therefore, it is easy to implement our algorithm on the quadcopter and to receive or transmit necessary data. The basic commands using MAVlink are also supported. Inside PX4, many terrains, sensors, and quadcopter types are provided using JavaScript. In this study, the 3D Robotics IRIS quadcopter was used without additional sensors, which were simulated on the basic terrain as a real world environment for verifying our designed control system. The wind disturbances of 1.0 m/s are considered as external disturbances.

The quadcopter parameters used in this model were chosen as shown in **Table 2**.

The goal of the control system is to control the attitude of the quadcopter following the reference set-point trajectory:

$$\begin{aligned} \theta_{ref} &= 5 * square\left(0.2\pi t\right) \\ \phi_{ref} &= 5 * square\left(0.2\pi t\right) \\ \psi_{ref} &= 90 + 5 * square\left(0.2\pi t\right) \end{aligned} \tag{34}$$

$$\begin{aligned} \theta_{ref} &= 5 * sin\left(0.2\pi t\right) \\ \phi_{ref} &= 5 * sin\left(0.2\pi t\right) \\ \psi_{ref} &= 90 + 5 * sin\left(0.2\pi t\right) \end{aligned} \tag{35}$$

The Gazebo simulation interface is shown in **Figure 7**. The tracking performances of the quadcopter attitude under square wave and sine wave commands are given in **Figures 8**, **11**, respectively. The black line presents the reference set-point trajectory. The red and blue lines present the quadcopter attitude output using the PID-MFNN controller and the PID conventional fuzzy controller (PID-CFNN), respectively. The control signals under the square wave and sine wave commands are shown in **Figures 9**, **12**, respectively. The tracking errors under the square wave and sine wave commands are given in **Figures 10**, **13**, respectively. The comparison results concerning the root mean square error (RMSE) between the proposed PID-MFNN controller and PID-CFNN controller are given in **Tables 3**, **4**. From **Figures 8**–**13**, it can be seen that the proposed PID-MFNN controller has better control performance with a faster convergence speed and smaller tracking errors compared with the conventional fuzzy controller, which demonstrates the effectiveness of our proposed control method. The RMSE was used to evaluate the effectiveness of the control performance as follows:

$$RMSE = \sqrt{\frac{1}{n_s}\sum_{s=1}^{n_s}\left(\left(e_\Theta^s\right)^2\right)}, \tag{36}$$

where n_s is the number of samples.

The parameter settings of the proposed PID-MFNN controller were chosen as $n_i = 2$, $n_j = 3$, and $n_k = 2$; the initial GMF for the fuzzy rules are given as $m_{ij1} = [-0.6, 0, 0.6]$, $v_{ij1} = [0.3, 03, 0.3]$, $m_{ij2} = [-0.4, 0, 0.4]$, and $v_{ij2} = [0.2, 02, 0.2]$; the

TABLE 2 | The 3DR's IRIS parameters inside the Gazebo simulation used in the experiment.

Parameter	Value	Unit
m	1.5	kg
l	0.275	m
h	0.11	m
g	9.81	m/s^2
J_x	0.0347563	$kg \cdot m^2$
J_y	0.0458929	$kg \cdot m^2$
J_z	0.0977	$kg \cdot m^2$
C_t	8.54858e-06	$N \cdot s^2$
C_d	0.000806428	$N \cdot m \cdot s^2$

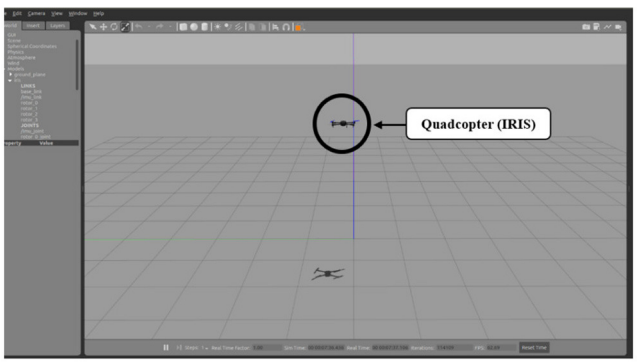

FIGURE 7 | Gazebo simulation interface.

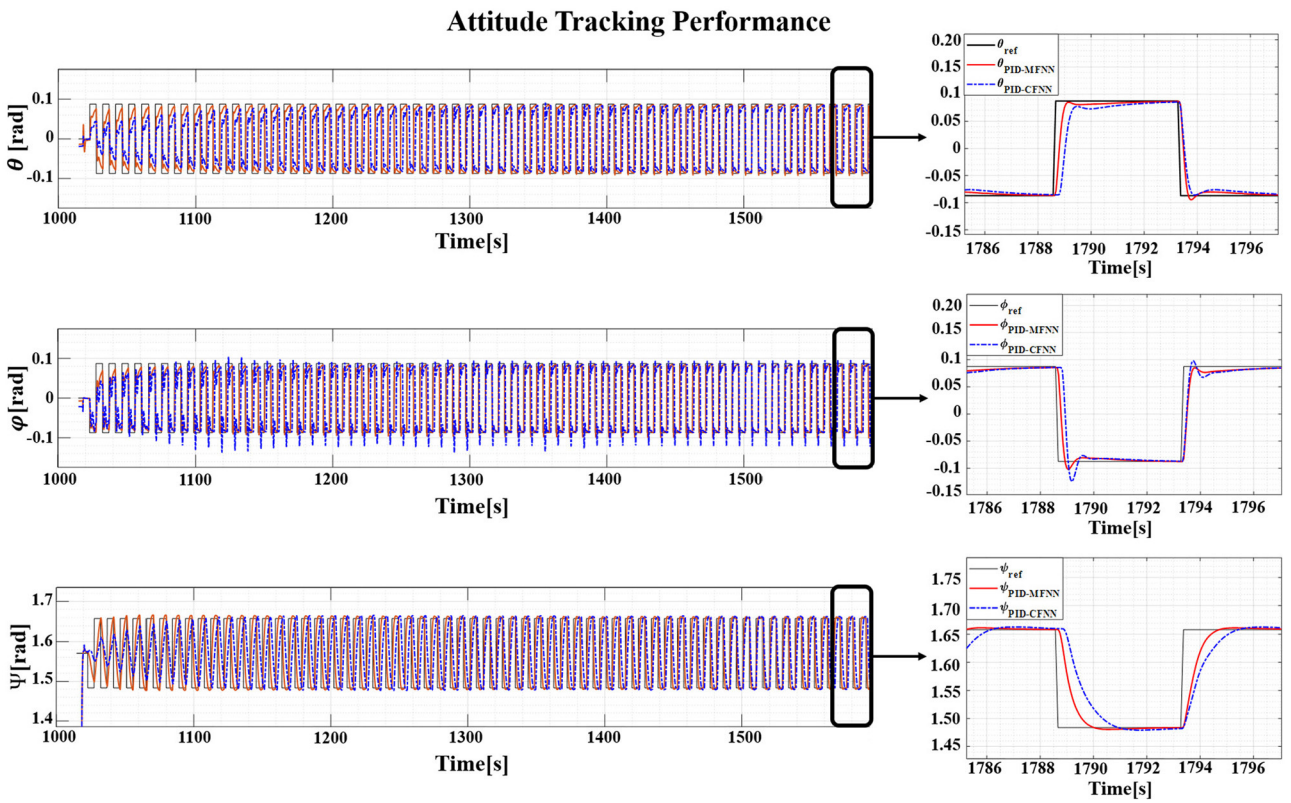

FIGURE 8 | The tracking performance for the quadcopter attitude control for the square wave commands.

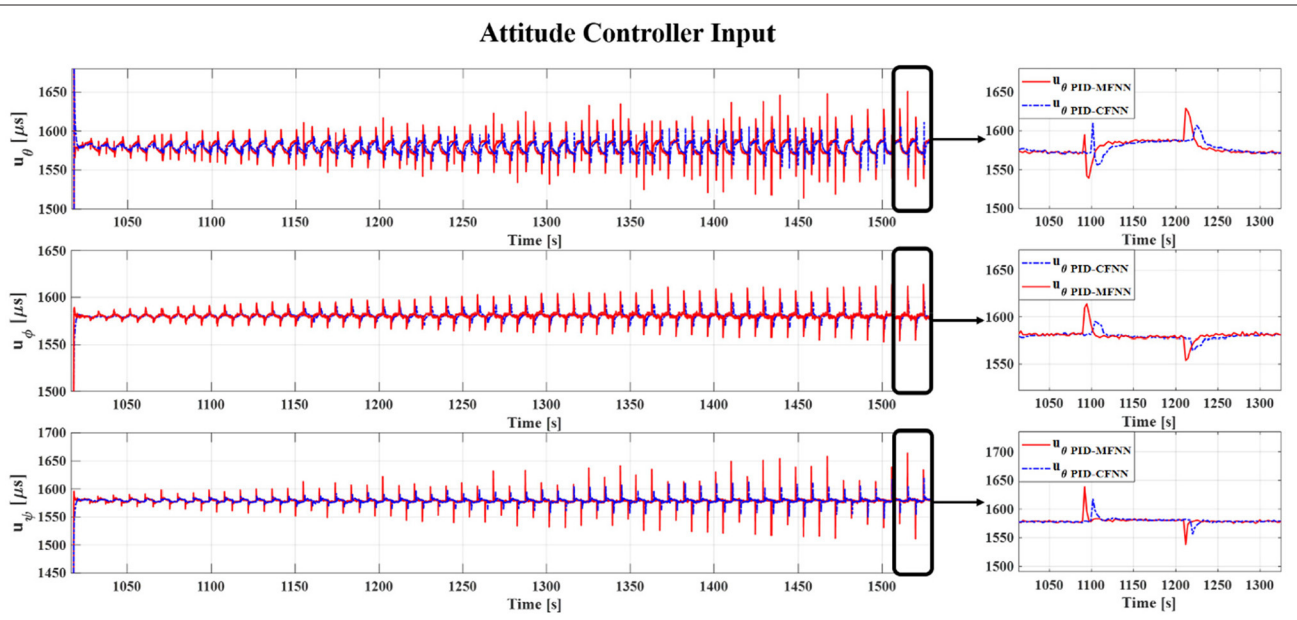

FIGURE 9 | The control signals for the quadcopter attitude control for the square wave commands.

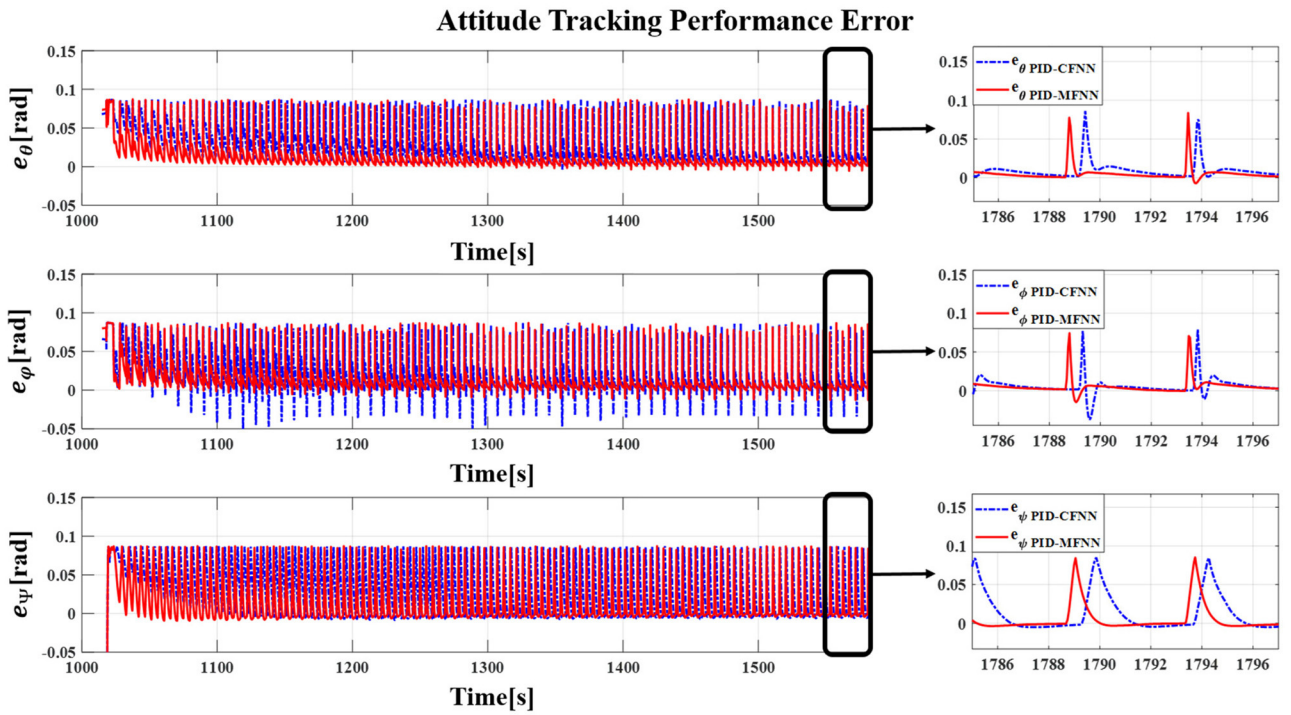

FIGURE 10 | The tracking errors for the quadcopter attitude control for the square wave commands.

TABLE 3 | RMSE comparison for the quadcopter attitude control for the square wave commands.

Controller	Computation time (ms)	Roll	Pitch	Yaw
PID-MFNN	0.11444	0.01896	0.01902	0.07713
PID-CFNN	0.11093	0.02698	0.02177	0.07967

TABLE 4 | RMSE comparison for the quadcopter attitude control for the sine wave commands.

Controller	Computation time (ms)	Roll	Pitch	Yaw
PID-MFNN	0.11444	0.0106	0.0106	0.1170
PID-CFNN	0.11093	0.0168	0.0154	0.1820

initial fuzzy weight is $w_{klo} = 0.2$; the learning rates were chosen as $\eta_w^g = \eta_w^\alpha = \eta_m^\alpha = \eta_v^\alpha = 0.005$; the sampling time was 0.01 s. The same parameters were chosen for PID-CFNN, but only the number of layers was $n_k = 1$, and all GMFs were put in one layer.

The simulation results in **Figures 8**, **10** show that the proposed PID-MFNN controller has better performance than the PID-CFNN controller in terms of the rising time, settling time, and overshoot. The multilayer structure in our proposed controller, which contains a multilayer MF, provides it with a better scope regarding errors and disturbances compared with the single MF structure in PID-CFNN. The proposed PID-MFNN had a little longer computation time than the PID-CFNN due to the processing time of the multilayer structure. However, this did not affect the control performance, and the proposed PID-MFNN controller could achieve a smaller RMSE in both simulation cases. Apply some methods such as latent analysis in Wu et al. (2019) and Wu et al. (2020) to reduce the computational cost for the proposed multilayer fuzzy neural network will be our future work.

Remark 1. The conventional PID usually hard to achieve high control performance for controlling non-linear systems due to its parameters are usually tuned for local points based on a linearization method (Mohan and Sinha, 2008; Cetin and Iplikci, 2015). Therefore, it is not suitable for controlling highly complex and non-linear systems, such as quadcopters. However, in our proposed PID-MFNN method, we designed adaptation laws for updating the MFNN network parameters online, so the output of the MFNN network can auto-tune the PID gains. Therefore, our proposed PID-MFNN is suitable for quadcopters as well as other non-linear control systems.

Remark 2. By separating the member functions into a multilayer structure, the proposed method can better cover the changes in inputs. Moreover, with the proposed multilayer structure, the learning ability and flexibility of the network can be further improved.

Remark 3. The difficulties and limits in this study are described as follows. When applying our method in an experiment with real quadcopters, suitable initial parameters needed to be chosen. Otherwise, control could be lost. In our study, the simulation results were obtained using a Gazebo robot simulator, and it was easy to choose these parameters using the

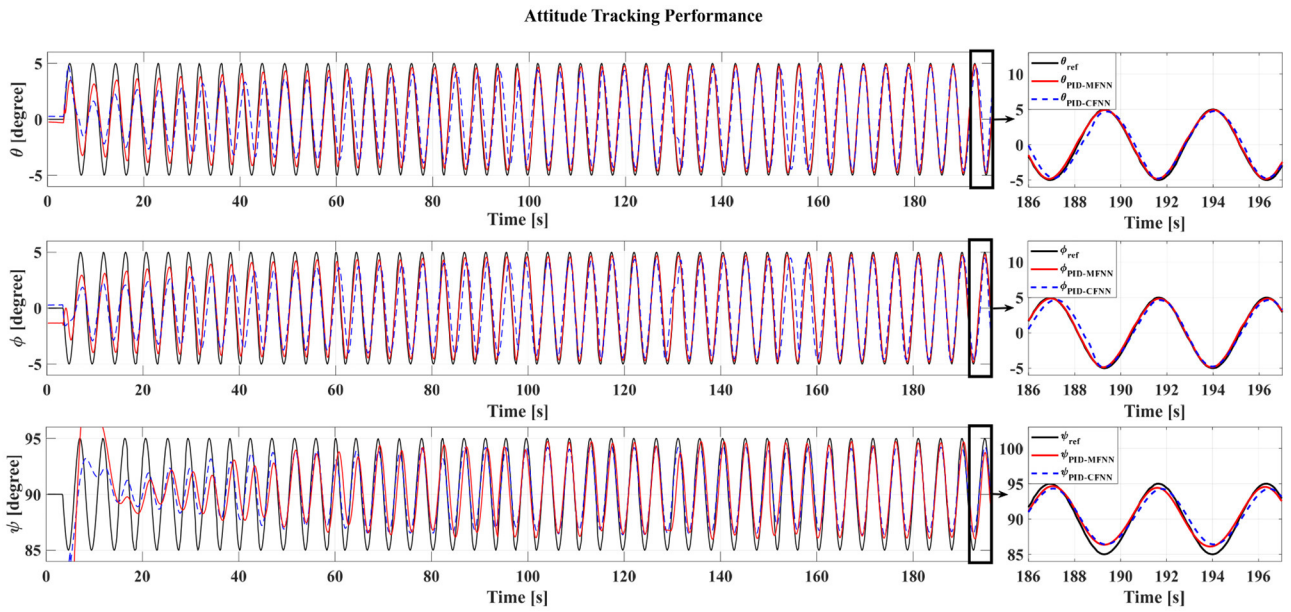

FIGURE 11 | The quadcopter attitude control tracking performance for the sine wave commands.

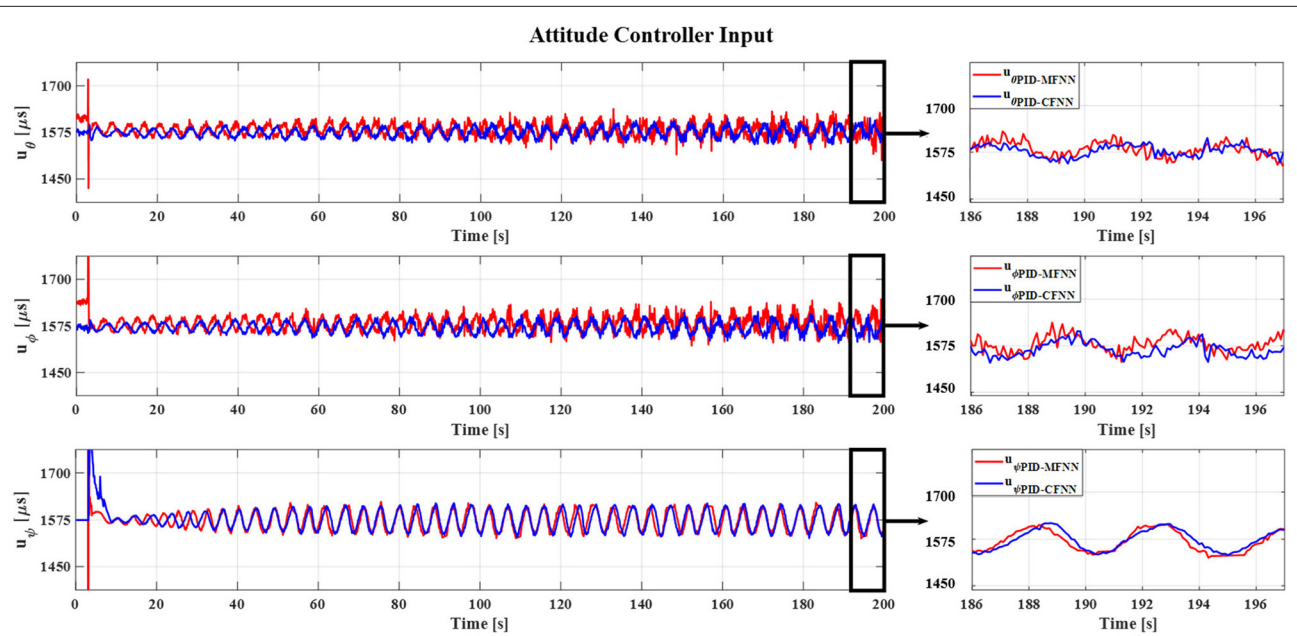

FIGURE 12 | The quadcopter attitude controller input for the sine wave commands.

trial-and-error method. In our future work, we plan to apply some optimal algorithms to realize suitable initial parameters.

CONCLUSION

In this study, auto-tuning PID parameters using MFNN were successfully designed for controlling the attitude of quadcopters. The proposed method provided an effective method for obtaining suitable gains for PID controllers without the need for a mathematical system model or complicated calculations. In addition, a new multilayer structure was provided to improve the learning ability and flexibility of the used FNN. The parameters of the proposed network could be updated online using adaptation

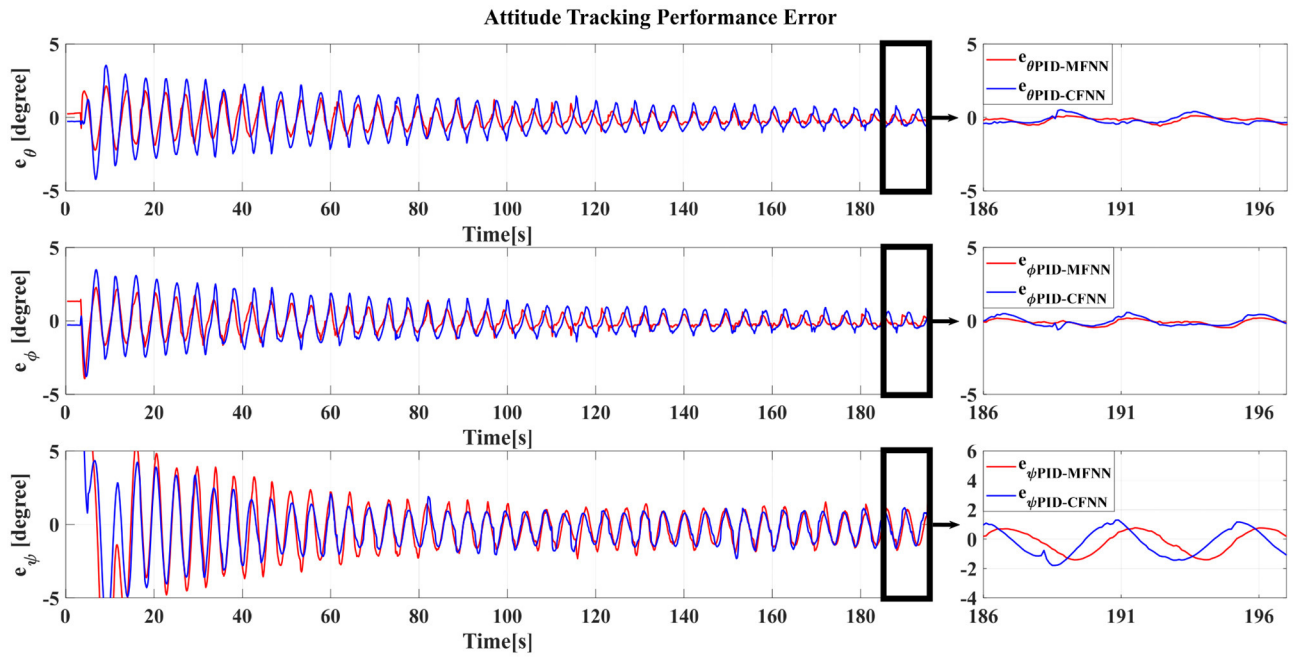

FIGURE 13 | The quadcopter attitude control tracking performance error for the sine wave commands.

laws. Finally, the effectiveness of the proposed method was illustrated through the results of the conducted numerical simulations of quadcopter attitude control using a Gazebo robotics simulator and ROS. Our designed controller can also apply to a real quadcopter, and in our future work, we plan to apply some optimal algorithms to achieve suitable initial parameters.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2020R1A6A1A03038540) and by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (N0002431, the Competency Development Program for Industry Specialist).

REFERENCES

Amoura, K., Mansouri, R., Bettayeb, M., and Al-Saggaf, U. M. (2016). Closed-loop step response for tuning PID-fractional-order-filter controllers. *ISA Trans.* 64, 247–257. doi: 10.1016/j.isatra.2016.04.017

Azman, A. A., Rahiman, M. H. F., Mohammad, N. N., Marzaki, M. H., Taib, M. N., and Ali, M. F. (2017). "Modeling and comparative study of PID Ziegler Nichols (ZN) and Cohen-Coon (CC) tuning method for multi-tube aluminum sulphate water filter (MTAS)," in 2017 *IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS)* (Sabah: IEEE), 25–30. doi: 10.1109/I2CACIS.2017.8239027

Bernardes, N. D., Castro, F. A., Cuadros, M. A., Salarolli, P. F., Almeida, G. M., and Munaro, C. J. (2019). Fuzzy logic in auto-tuning of fractional PID and backstepping tracking control of a differential mobile robot. *J. Intell. Fuzzy Syst.* 37, 4951–4964. doi: 10.3233/JIFS-181431

Cetin, M., and Iplikci, S. (2015). A novel auto-tuning PID control mechanism for nonlinear systems. *ISA Trans.* 58, 292–308. doi: 10.1016/j.isatra.2015.05.017

Chen, Y., Zhang, G., Zhuang, Y., and Hu, H. (2019). Autonomous flight control for multi-rotor UAVs flying at low altitude. *IEEE Access* 7, 42614–42625. doi: 10.1109/ACCESS.2019.2908205

Concha, A., Varadharaj, E., Hernandez-Rivera, N., and Gadi, S. (2017). "A novel implementation technique for genetic algorithm based auto-tuning PID controller," in 2017 *IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)* (Chennai: IEEE), 1403–1408. doi: 10.1109/ICPCSI.2017.8391942

Davanipour, M., Javanmardi, H., and Goodarzi, N. (2018). Chaotic self-tuning PID controller based on fuzzy wavelet neural network model. *Iran. J. Sci. Technol. Trans. Electr. Eng.* 42, 357–366. doi: 10.1007/s40998-018-0069-1

De Keyser, R., Muresan, C. I., and Ionescu, C. M. (2016). A novel auto-tuning method for fractional order PI/PD controllers. *ISA Trans.* 62, 268–275. doi: 10.1016/j.isatra.2016.01.021

Fatan, M., Sefidgari, B. L., and Barenji, A. V. (2013). "An adaptive neuro pid for controlling the altitude of quadcopter robot," in 2013 18*th International*

Conference on Methods & Models in Automation & Robotics (MMAR) (Międzyzdroje: IEEE), 662–665. doi: 10.1109/MMAR.2013.6669989

Hernandez, Y. L., and Frias, O. O. G. (2016). Design and control of a four-rotary-wing aircraft. *IEEE Latin Am. Trans.* 14, 4433–4438. doi: 10.1109/TLA.2016.7795811

Kim, T.-H., Maruta, I., and Sugie, T. (2007). "Particle swarm optimization based robust PID controller tuning scheme," in 2007 46*th IEEE Conference on Decision and Control* (New Orleans, LA: IEEE), 200–205.

Kuantama, E., Vesselenyi, T., Dzitac, S., and Tarca, R. (2017). PID and Fuzzy-PID control model for quadcopter attitude with disturbance parameter. *Int. J. Comput. Commun. Control* 12, 519–532. doi: 10.15837/ijccc.2017.4.2962

Kurak, S., and Hodzic, M. (2018). Control and estimation of a quadcopter dynamical model. *Period. Eng. Nat. Sci.* 6, 63–75. doi: 10.21533/pen.v6i1.164

Live, H., Wang, H., Zhu, X., Shen, Z., and Chen, J. (2017). "Simulation research of fuzzy auto-tuning PID controller based on matlab," in 2017 *International Conference on Computer Technology, Electronics and Communication (ICCTEC)* (Dalian: IEEE), 180–183. doi: 10.1109/ICCTEC.2017.00047

Mendes, J., Osório, L., and Araújo, R. (2017). Self-tuning PID controllers in pursuit of plug and play capacity. *Control Eng. Pract.* 69, 73–84. doi: 10.1016/j.conengprac.2017.09.006

Mohan, B., and Sinha, A. (2008). Analytical structures for fuzzy PID controllers? *IEEE Trans. Fuzzy Syst.* 16, 52–60. doi: 10.1109/TFUZZ.2007.894974

Noordin, A., Basri, M. M., Mohamed, Z., and Abidin, A. Z. (2017). Modelling and PSO fine-tuned PID control of quadrotor UAV. *Int. J. Adv. Sci. Eng. Inform. Technol.* 7, 1367–1373. doi: 10.18517/ijaseit.7.4.3141

Prayitno, A., Indrawati, V., and Trusulaw, I. I. (2018). Fuzzy gain scheduling PID control for position of the AR. Drone. *Int. J. Electr. Comput. Eng.* 8, 1939–1946. doi: 10.11591/ijece.v8i4.pp1939-1946

Rabah, M., Rohan, A., Mohamed, S. A., and Kim, S.-H. (2019). Autonomous moving target-tracking for a UAV quadcopter based on fuzzy-PI. *IEEE Access* 7, 38407–38419. doi: 10.1109/ACCESS.2019.2906345

Rouhani, H., Same, M., Masani, K., Li, Y. Q., and Popovic, M. R. (2017). PID controller design for FES applied to ankle muscles in neuroprosthesis for standing balance. *Front. Neurosci.* 11:347. doi: 10.3389/fnins.2017.00347

Salas, F. G., Orrante-Sakanassi, J., Juarez-Del-Toro, R., and Parada, R. P. (2019). A stable proportional-proportional integral tracking controller with self-organizing fuzzy-tuned gains for parallel robots. *Int. J. Adv. Rob. Syst.* 16:1729881418819956. doi: 10.1177/1729881418819956

Santos, M., Honório, L., Costa, E., Silva, M., Vidal, V., Neto, A. S., et al. (2019). "Detection time analysis of propulsion system fault effects in a hexacopter," in 2019 20*th International Carpathian Control Conference (ICCC)* (Kraków-Wieliczka: IEEE), 1–6. doi: 10.1109/CarpathianCC.2019.8765990

Sarabakha, A., Fu, C., Kayacan, E., and Kumbasar, T. (2017). Type-2 fuzzy logic controllers made even simpler: from design to deployment for UAVs. *IEEE Trans. Industr. Electron.* 65, 5069–5077. doi: 10.1109/TIE.2017.2767546

Soriano, L. A., Zamora, E., Vazquez-Nicolas, J., Hernández, G., Barraza, J. A., and Balderas, D. (2020). PD control compensation based on the reinforcement learning applied to a robot manipulator. *Front. Neurorobot.* 14:78. doi: 10.3389/fnbot.2020.577749

Sumardi, M., and Riyadi, M. A. (2017). "Particle swarm optimization (PSO)-based self tuning proportional integral derivative (PID) for bearing navigation control system on quadcopter," in *2017 4th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)* (Semarang), 181–186

Thanh, H. L. N. N., and Hong, S. K. (2018). Quadcopter robust adaptive second order sliding mode control based on PID sliding surface. *IEEE Access* 6, 66850–66860. doi: 10.1109/ACCESS.2018.2877795

Tripathy, D., Barik, A. K., Choudhury, N. B. D., and Sahu, B. K. (2019). "Performance comparison of SMO-based fuzzy PID controller for load frequency control," in *Soft Computing for Problem Solving*, eds J. C. Bansal, K. N. Das, A. Nagar, K. Dee, and A. K. Ojha (Khordha: Springer), 879–892. doi: 10.1007/978-981-13-1595-4_70

Wang, L. (2017). Automatic tuning of PID controllers using frequency sampling filters. *IET Control Theory Appl.* 11, 985–995. doi: 10.1049/iet-cta.2016.1284

Wang, Y., Jin, Q., and Zhang, R. (2017). Improved fuzzy PID controller design using predictive functional control structure. *ISA Trans.* 71, 354–363. doi: 10.1016/j.isatra.2017.09.005

Wu, D., Luo, X., Shang, M., He, Y., Wang, G., and Wu, X. (2020). A data-characteristic-aware latent factor model for web services QoS prediction. *IEEE Trans. Knowl. Data Eng.* doi: 10.1109/TKDE.2020.3014302. [Epub ahead of print].

Wu, D., Luo, X., Shang, M., He, Y., Wang, G., and Zhou, M. (2019). A Deep latent factor model for high-dimensional and sparse matrices in recommender systems. *IEEE Trans. Syst. Man Cybern. Syst.* doi: 10.1109/TSMC.2019.2931393. [Epub ahead of print].

Xuan-Mung, N., and Hong, S. K. (2019a). Improved altitude control algorithm for quadcopter unmanned aerial vehicles. *Appl. Sci.* 9:2122. doi: 10.3390/app9102122

Xuan-Mung, N., and Hong, S. K. (2019b). Robust adaptive formation control of quadcopters based on a leader-follower approach. *Int. J. Adv. Rob. Syst.* 16:1729881419862733. doi: 10.1177/1729881419862733

Zahir, A., Alhady, S., Othman, W., and Ahmad, M. (2018). "Genetic algorithm optimization of PID controller for brushed DC motor," in *Intelligent Manufacturing & Mechatronics*, ed M. H. A. Hassan (Pekan: Springer), 427–437. doi: 10.1007/978-981-10-8788-2_38

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.