drones

MDPI

Article

Model-Free RBF Neural Network Intelligent-PID Control Applying Adaptive Robust Term for Quadrotor System

Sung-Jae Kim [1] and Jin-Ho Suh [2,*]

[1] The Industrial Science Technology Research Center, Pukyong National University, Busan 48513, Republic of Korea; bbman7020@gmail.com

[2] Major of Mechanical System Engineering, Pukyong National University, Busan 48513, Republic of Korea

* Correspondence: suhgang@pknu.ac.kr

Abstract: This paper proposes a quadrotor system control scheme using an intelligent–proportional–integral–differential control (I-PID)-based controller augmented with a radial basis neural network (RBF neural network) and the proposed adaptive robust term. The I-PID controller, similar to the widely utilized PID controller in quadrotor systems, demonstrates notable robustness. To enhance this robustness further, the time-delay estimation error was compensated with an RBF neural network. Additionally, an adaptive robust term was proposed to address the shortcomings of the neural network system, thereby constructing a more robust controller. This supplementary control input integrated an adaptation term to address significant signal changes and was amalgamated with a reverse saturation filter to remove unnecessary control input during a steady state. The adaptive law of the proposed controller was designed based on Lyapunov stability to satisfy control system stability. To verify the control system, simulations were conducted on a quadrotor system maneuvering along a spiral path in a disturbed environment. The simulation results demonstrate that the proposed controller achieves high tracking performance across all six axes. Therefore, the controller proposed in this paper can be configured similarly to the previous PID controller and shows satisfactory performance.

Keywords: quadrotor; attitude control; I-PID control; time-delayed control; RBF neural network; prescribed performance

Citation: Kim, S.-J.; Suh, J.-H. Model-Free RBF Neural Network Intelligent-PID Control Applying Adaptive Robust Term for Quadrotor System. *Drones* **2024**, *8*, 179. https://doi.org/10.3390/drones8050179

Academic Editor: Xiangwei Bu

Received: 28 March 2024
Revised: 26 April 2024
Accepted: 29 April 2024
Published: 1 May 2024

1. Introduction

Quadrotors have symmetric structures and high maneuverability, enabling hovering and demonstrating good flight performance. Currently, these characteristics find application across various industries, leveraging their capabilities [1–3]. Moreover, quadrotors are garnering significant attention not only for practical applications but also from researchers. Despite their favorable behavior, quadrotor systems present significant control challenges due to strong nonlinear coupling dynamics, model uncertainties, unmeasurable disturbances, and under-actuated problems. Consequently, these complexities pose formidable challenges for control engineers, prompting extensive research efforts [3–17].

In [7], dynamics were estimated using FCRNN (Fully Connected Recurrent Neural Network) and controlled using ISMC (Integral Sliding Mode Control). In [8], trajectory tracking was achieved by applying RGSMC (Rapid Global Sliding Mode Control) using a fast decay function. Attitude stabilization was attained in [13] through a nominal state-feedback controller and a robust compensator, while [14] utilized the PD2 feedback structure for the same purpose. Furthermore, Ref. [15] accomplished robust attitude control against disturbances through a switching model predictive attitude controller, showcasing promising results. Ref. [16] introduced a dynamic event-triggered control strategy that maximized the efficiency of maritime search operations through the collaboration of USVs and UAVs. The strategy enhanced system stability by efficiently managing key control inputs and robustly resisting external disturbances. Ref. [17] presented a control algorithm designed to efficiently perform path-following tasks through the cooperation of USVs and UAVs. Aiming to minimize complex design requirements and command transmissions, it utilizes a dynamic event-triggered mechanism to reduce data transmission and maximize system efficiency.

Their research shows excellent results. However, their control methods use model information. Since system identification of a control system is difficult, it can be hard to apply to a quadrotor system. Therefore, in one study, a quadrotor was controlled using a PID controller, a representative, model-free control method. It can be applied to various quadrotor systems with a simple control method and can lead to satisfactory performance. Although this control method has the advantage of being easy to apply, it also has the disadvantage of reducing control performance due to external forces and system uncertainty. Thus, this paper proposes a control method that combines an I-PID (Intelligent-PID) controller and an RBF (Radial Basis Function) neural network.

I-PID control is a model-free nonlinear control method [18–20], which offers advantages for application in complex systems such as robots and quadrotors. This method relies on previous control inputs and outputs rather than modeling information. Therefore, this method is one of the promising control techniques that can replace traditional PID, and there are currently results of its application to various types of plants [18–20]. Moreover, this controller exhibits similarities to time-delay estimation errors and a lack of robustness in the transient state [20–22]. In particular, time-delay estimation error is a problem that must be solved as it reduces the robustness of the controller and deteriorates control performance. To mitigate this issue, we employed an artificial neural network for compensation.

Artificial neural networks are powerful tools for solving control problems such as nonlinearity and disturbance approximation. In addition, a neural network does not require complex mathematical analysis such as disturbance observers and can be configured independently of the system [23–26]. These features can be implemented without the model information required in this paper and are easy to apply to the controller. However, neural networks have some disadvantages when online learning. Firstly, neural networks require a lot of computation. This can increase computation time and reduce the control rate. Secondly, neural networks are difficult to mathematically analyze. This issue combines with control systems to make stability analysis difficult. Thirdly, due to initial value problems, neural networks can make control systems unstable. This paper addresses this issue by employing an RBF neural network and an adaptive robust term for compensation.

The RBF neural network has a simple structure consisting of one layer each. The advantage of this neural network is that it is mathematically analyzed, and the computational amount is smaller than for other neural networks. Therefore, as in this paper, it is evaluated as a neural network suitable for mechatronics system control [27–34]. The activation function uses a Gaussian function. This function follows a probability distribution and has excellent nonlinear approximation ability when used with neural networks [27–31]. However, a challenge arises as Gaussian functions are not differentiable. In this paper, we design a Lyapunov-based update law to consider control system stability without using differentiation in the activation function.

To overcome some of the shortcomings of neural networks, we used adaptive robust terms. An adaptive robust term originates from a robust term. A robust term consists of constant gain and error sign functions. This ability of the robust term nicely suppresses the shortcomings that occur during the online learning of neural networks [29–35]. Although this may reduce the estimation performance of the network, it is a practical way to make control systems containing neural networks more robust. However, robust terms also have some disadvantages. A robust term uses constant control gain and cannot respond to large signal changes. Additionally, by generating unnecessary control inputs under a steady state, there is a risk of worsening control performance in a steady state and causing chattering. Therefore, we propose an adaptive robust term that can dynamically adapt to signal changes by adjusting the control gain and that utilizes a reverse saturation filter that can remove unnecessary control inputs in the steady state. In conclusion, this paper proposes I-PID control, which can be configured similarly to PID and is more robust to disturbances. This study used the RBF neural network to compensate for the time-delay estimation error, which is a drawback of I-PID. Afterwards, we proposed an adaptive robust term to overcome the shortcomings of neural network control and design a more robust controller.

The contributions of this paper are as follows:

- We performed mathematical modeling to control a quadrotor system and designed a controller that combines an I-PID controller and an RBF neural network.
- To make the control system more robust, we designed an adaptive robust term that includes a reverse saturation filter.
- For the proposed controller, we designed update laws based on Lyapunov stability.
- Stability was rigorously proven by investigating the control boundness of the whole control system.
- The performance of the proposed controller was proven through simulation.

This paper is structured as follows: Section 2 introduces the quadrotor system and its dynamics. Section 3 shows the design of the I-PID control with the RBF neural network and the design of the adaptive robust term; it also proves the whole control system's stability. Section 4 compares and verifies the performance of the proposed controller through computer simulation. Section 5 discusses the conclusions and future work.

2. Dynamic Model of the Quadrotor System

The quadrotor system moved by regulating the angular speed of four rotors that were equidistant from the center of mass. Due to these control methods, the quadrotor achieved six-degrees-of-freedom movement, including vertical take-off and landing (VTOL) and hovering. However, the quadrotor system needed to maintain its position using the four rotors that controlled axial movement. In other words, the quadrotor had an underactuated problem. Quadrotor dynamic models including these issues have been investigated in previous studies [4–15,36–39]. We used a quadrotor dynamic model with reference to previous studies.

Figure 1 is a coordinate system of the quadrotor considered in this paper. The quadrotor shows the position and orientation in an inertial coordinate system with 6 DOF $(x, y, z, \phi, \theta, \psi)$.

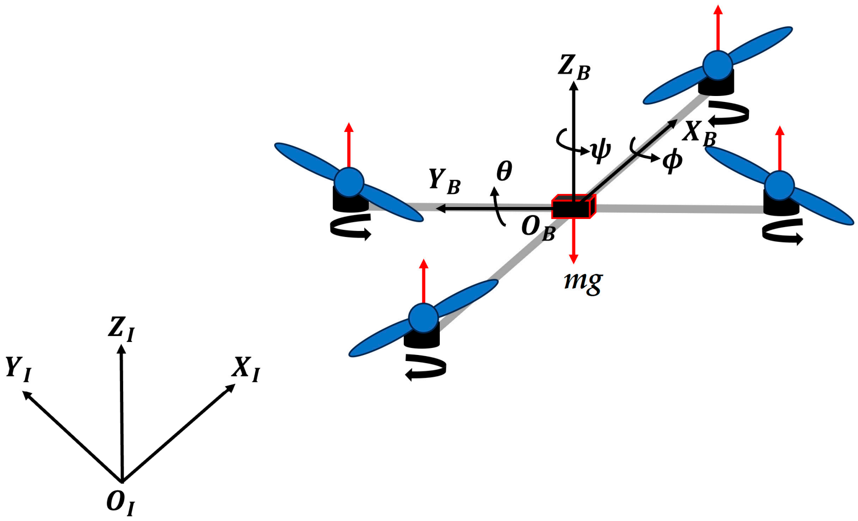

Figure 1. Coordinate system of quadrotor.

The linear velocity in the object coordinate system relative to the inertial coordinate system was defined as Equation (2) using Equation (1).

$$R = \begin{bmatrix} cos\,\psi\,cos\,\theta & cos\,\psi\,sin\,\theta\,sin\,\phi - sin\,\psi\,cos\,\phi & cos\,\psi\,sin\,\theta\,cos\,\phi + sin\,\psi\,sin\,\phi \\ sin\,\psi\,cos\,\theta & sin\,\psi\,sin\,\theta\,sin\,\phi + cos\,\psi\,cos\,\phi & sin\,\psi\,sin\,\theta\,cos\,\phi - cos\,\psi\,sin\,\phi \\ -sin\,\theta & cos\,\theta\,sin\,\phi & cos\,\theta\,cos\,\phi \end{bmatrix} \tag{1}$$

$$V_O = R^T V_I \tag{2}$$

where the object coordinate system velocity $V_O = [\dot{u} \quad \dot{v} \quad \dot{w}]$ and the inertial coordinate system velocity $V_I = [\dot{x} \quad \dot{y} \quad \dot{z}]$.

The angular velocity in the object coordinate system relative to the inertial coordinate system was defined as Equation (4) using Equation (3).

$$\hat{R} = \begin{bmatrix} 1 & 0 & -sin\,\theta \\ 0 & cos\,\phi & cos\,\theta\,sin\,\phi \\ 0 & -sin\,\phi & cos\,\theta\,cos\,\phi \end{bmatrix} \tag{3}$$

$$\omega_O = \hat{R}^T \omega_I \tag{4}$$

where the object coordinate system angular velocity $\omega_O = [\dot{p} \quad \dot{q} \quad \dot{r}]$ and the inertial coordinate system angular velocity $\omega_I = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]$. Due to the characteristics of the quadrotor, θ, and ϕ needed to remain around zero to maintain the quadrotor attitude. Therefore, this could be expressed as $cos\theta \cong cos\phi \cong 1$ and $sin\theta \cong sin\phi \cong 0$. Finally, using Equations (2) and (4), the dynamic model of the quadrotor could be expressed as Equation (5).

$$\begin{cases} \ddot{\phi} = \dfrac{1}{I_x}\left[\dot{\theta}\dot{\psi}\left(I_y - I_z\right) - J_r\bar{\Omega}\dot{\theta} + U_\phi + D_\phi\right] \\ \ddot{\theta} = \dfrac{1}{I_y}\left[\dot{\phi}\dot{\psi}(I_z - I_x) + J_r\bar{\Omega}\dot{\phi} + U_\theta + D_\theta\right] \\ \ddot{\psi} = \dfrac{1}{I_z}\left[\dot{\theta}\dot{\phi}\left(I_x - I_y\right) + U_\phi + D_\psi\right] \\ \ddot{x} = \dfrac{1}{m}[(cos\,\phi\,sin\,\theta\,sin\,\psi + sin\,\phi\,sin\,\psi)U_z + D_x] \\ \ddot{y} = \dfrac{1}{m}\left[(cos\,\phi\,sin\,\theta\,sin\,\psi - sin\,\phi\,cos\,\psi)U_z + D_y\right] \\ \ddot{z} = \dfrac{1}{m}[(cos\,\phi\,cos\,\theta)U_z - mg + D_z] \end{cases} \tag{5}$$

where m is the mass of the quadrotor, I_i is the inertia moment of the quadrotor, J_r is the inertia moment of the rotors, U_i is the control input, g is the gravitational acceleration, $\overline{\Omega}$ is the total rotor velocity, and D_i is the lumped disturbance including aerodynamic friction, drag force, and unmeasurable disturbance.

3. Quadrotor Controller Design

The proposed controller was composed of three parts: an I-PID controller, an RBF neural network for compensated time-delay estimation error, and an adaptive robust term for increased robustness of the control system.

3.1. I-PID Control

To construct the I-PID controller, we assumed a nonlinear second-order system such as Equation (6).

$$M\ddot{X}(t) = F(t) + u(t) \tag{6}$$

where M is an inertia matrix, $F(t)$ is nonlinear plant dynamics, and $u(t)$ is the control input. In Equation (6), if the sampling time h was fast, the system model could be approximated as Equation (7).

$$F(t) \simeq M\ddot{X}(t) - u(t-h) \tag{7}$$

The control input of the I-PID controller to be used in this study was designed as in Equation (8) using Equation (7).

$$e(t) = X_d - X$$
$$u(t) \simeq \underbrace{\left(u(t-h) - \bar{M}\ddot{X}\right)}_{\text{Linearization of the Plant}} + \underbrace{\bar{M}\left(\ddot{X}_d + K_P e(t) + K_I \int e(\tau)d\tau + K_D \dot{e}(t)\right)}_{\text{Desired error dynamics}} \tag{8}$$

To construct the controller in this study, where $X = [z\ \phi\ \theta\ \psi]^T$ system state, X_d was the desired system state; K_P, K_I, and K_D were control gains; and $\bar{M}$ was a constant inertia estimation matrix. In other I-PID-related studies [16–18], this is expressed as $1/\alpha$. In this paper, it is expressed as $\bar{M}$. This controller was similar to time-delay control. Therefore, this controller consisted of two parts: linearization of the plant and desired error dynamics.

$$\ddot{e} + K_P e(t) + K_I \int e(\tau)d\tau + K_D \dot{e}(t) = 0 \tag{9}$$

In the linearization of the plant, information from previous time steps was used to approximate and compensate for the system model containing lumped disturbances. Control performance achieved closed-loop error dynamics, as shown in Equation (9), through desired error dynamics. This controller could determine the performance of a closed-loop system through error dynamics.

$$\ddot{e} + K_P e(t) + K_I \int e(\tau)d\tau + K_D \dot{e}(t) = \Delta H \tag{10}$$

However, in real systems, time-delay estimation errors ΔH such as those in Equation (10) occur due to uncertainties such as disturbances and time delays. This error may reduce control performance; therefore, we compensated for this through the RBF neural network.

3.2. RBF Neural Network

To compensate for the time-delay estimation error, this study used an RBF neural network. It is very difficult to mathematically define time-delay estimation error. However, RBF neural networks are independent of control systems and allow for the estimation of time-delay estimation errors without complex mathematical analysis. In this study, we estimated the time-delay estimation error using the same 2-5-1 network structure as in previous research cases [29,30].

Figure 2 is the network structure used in this study, where $e = \left[e_z\ e_\phi\ e_\theta\ e_\psi\right]^T$, $\dot{e} = \left[\dot{e}_z\ \dot{e}_\phi\ \dot{e}_\theta\ \dot{e}_\psi\right]^T$, h_i is the activation function output, and w_i is weight value. The activation function was a Gaussian function and was expressed as Equation (11).

$$h_{ij} = exp(-\left\|x_i - c_j\right\|^2/2{b_j}^2) \tag{11}$$

where x_i is a Gaussian function input, c_j represents the coordinate value of the center point of the Gaussian function, and b_j represents the width of the Gaussian function. The output of this neural network could be expressed as Equation (12).

$$\Delta\hat{H} = \widehat{W}^T H(E) \tag{12}$$

where $\widehat{W} = [w_1 \cdots w_5]$ is the weight vector, $H = [h_1 \cdots h_5]^T$ is the activation function output vector, and E is the input vector.

$$\Delta\bar{H} = \bar{W}^T H(E) + \varepsilon \tag{13}$$

Equation (13) is the optimal time-delay estimation error: $\overline{W}$ is the optimal weight vector and ε is the neural network estimation error. To design the weight update law, we utilized Equation (10) and considered Equation (14), which was an error state space model.

$$\dot{E}_i = AE_i + B \cdot \Delta\widetilde{H}_i \tag{14}$$

where A is the desired error dynamics control gain matrix and P is the positive definite constant matrix that satisfied Equation (15).

$$A^T P + PA = -Q \tag{15}$$

Matrix Q is explicitly defined as a positive definite matrix.

$$\dot{\hat{W}} = \gamma(E_i^T PBH(E) - \sigma\widetilde{W}) \tag{16}$$

Equation (16) presents a weight update law formulated on the principles of Lyapunov stability without differentiation of the activation function. In this equation, $\widetilde{W} = \widehat{W} - \overline{W}$ represents the weight error vector and both γ and σ are positive control gains.

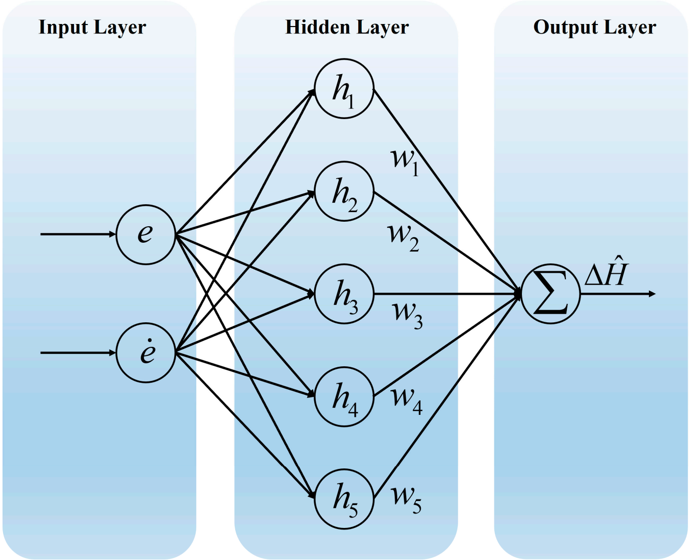

Figure 2. RBF neural network structure.

3.3. Adaptive Robust Term

Robust terms are used to increase the robustness of control systems using neural networks [29–35]. A robust term consists of a constant control gain and an error sign function similar to the switching term of sliding mode control. This control technique is effective in improving control performance by compensating for estimation errors in controllers using neural networks. However, it cannot respond to large signal changes and, even if it has reached a steady state, it can affect control performance by generating unnecessary control input. We propose an adaptive robust term that combines a robust term with a filter that can adaptively generate signals and remove unnecessary inputs in the steady state.

To design the adaptive robust term, we redefined the optimal time-delay estimation error as Equation (17).

$$\Delta\bar{H} = \bar{W}^T H(E) + \bar{\varepsilon} \tag{17}$$

$\bar{\varepsilon}$ denotes the optimal neural network estimation error. The adaptive robust term proposed in this paper compensated for neural network estimation error and increased the robustness of the control system.

$$s = \mu_1 E + \mu_2 \dot{E} \tag{18}$$

Equation (18) is the error surface for designing the adaptive term; μ is control gain. We approximated the nonlinearity of the neural network estimation error as the product of a nonlinear term and an error surface. Therefore, optimal neural network estimation error could be regarded as $\bar{\varepsilon} = \bar{\rho} \cdot s$. The adaptive robust term proposed in this paper is Equation (19), which was defined as the approximate neural network estimation error $\hat{\varepsilon}$.

$$\hat{\varepsilon} = \hat{\rho} \cdot s \tag{19}$$

$\bar{\rho}$ is an optimal nonlinear term; $\hat{\rho}$ is an approximated nonlinear term.

$$\dot{\hat{\rho}} = \kappa(E_i^T PBs - \beta\tilde{\rho}) \tag{20}$$

where $\tilde{\rho} = \hat{\rho} - \bar{\rho}$ and β is positive control gain. Equation (20) is a nonlinear term update law designed based on Lyapunov stability.

3.4. Proposed Controller Design

We designed the controller by integrating the previously proposed I-PID controller, RBF neural network, and adaptive robust term.

$$\begin{aligned} u(t) \simeq & \underbrace{\left(u(t-h) - \bar{M}\ddot{X}\right)}_{\text{Linearization of the Plant}} + \underbrace{\bar{M}\left(\ddot{X}_d + K_P e(t) + K_I \int e(\tau)d\tau + K_D \dot{e}(t)\right)}_{\text{Desired error dynamics}} \\ & + \underbrace{\bar{M} \cdot \Delta\hat{H}}_{\text{RBF neural network}} + \underbrace{\bar{M} \cdot \hat{\varepsilon}}_{\text{Adaptive robust term}} \end{aligned} \tag{21}$$

Equation (21) is the control input including RBF neural network output and the adaptive robust term. Using Equation (21), we could obtain closed-loop error dynamics such as Equation (22).

$$\dot{E}_i = AE_i + B \cdot N \tag{22}$$

$N = \widetilde{W}H(E_i) + \tilde{\rho}s$. Next, we considered the Lyapunov function candidate that could represent the control system, such as Equation (23).

$$V = E_i^T PE_i/2 + \widetilde{W}^T\widetilde{W}/2\gamma + \tilde{\rho}^T\tilde{\rho}/2\kappa \tag{23}$$

Its time derivative was Equation (24).

$$\begin{aligned} \dot{V} &= \dot{E}_i^T PE_i/2 + E_i^T P\dot{E}/2 + \widetilde{W}^T\dot{\widehat{W}}/\gamma + \tilde{\rho}^T\dot{\hat{\rho}}/\kappa \\ &= (AE_i + B \cdot N)PE_i/2 + E_i^T P(AE_i + B \cdot N)/2 + \widetilde{W}^T\dot{\widehat{W}}/\gamma + \tilde{\rho}^T\dot{\hat{\rho}}/\kappa \\ &= E_i^T(A^T P + PA)E_i/2 + E_i^T PN + \widetilde{W}^T\dot{\widehat{W}}/\gamma + \tilde{\rho}^T\dot{\hat{\rho}}/\kappa \\ &= -E_i^T QE_i/2 - E_i^T P\widetilde{W}^T BH(E) + \widetilde{W}^T\dot{\widehat{W}}/\gamma - E_i^T PB\tilde{\rho}^T s + \tilde{\rho}^T\dot{\hat{\rho}}/\kappa \\ &= -E_i^T QE_i/2 - \widetilde{W}^T(E_i^T PBH(E) - \dot{\widehat{W}}/\gamma) - \tilde{\rho}^T\left(E_i^T PBs - \dot{\hat{\rho}}/\kappa\right) \end{aligned} \tag{24}$$

Using Equations (16) and (17), we could obtain Equation (25), which satisfied the Lyapunov stability condition.

$$\dot{V} = -E_i^T QE_i/2 - \sigma\widetilde{W}^T\widetilde{W} - \beta\tilde{\rho}^T\tilde{\rho} \leq 0 \tag{25}$$

The proposed control method satisfied the Lyapunov stability condition. However, adaptive robust terms can cause unnecessary control input in a steady state. This situation may adversely affect control performance. Therefore, the reverse saturation filter was used in the adaptive robust term to prevent this.

$$sat_{rev}(u) = \begin{cases} -|u| & \text{if:} \quad u < -N \\ 0 & \text{if:} -N \leq u \leq N \\ +|u| & \text{if:} \quad u > N \end{cases} \tag{26}$$

The reverse saturation filter was defined as Equation (26), and the adaptive robust term operated for errors above the threshold value and removed unnecessary control input for errors below the threshold value. The proposed adaptive robust term applying this filter could be defined as $\hat{\varepsilon} \approx sat_{rev}(\hat{\varepsilon})$.

The optimal weights and optimal nonlinear terms were unknown. Therefore, the weight error and nonlinear term error in the weight update rule and nonlinear term update rule could not be known. For this reason, $\widetilde{W} = \widehat{W} - \overline{W}$ and $\tilde{\rho} = \hat{\rho} - \bar{\rho}$ could not be included in the update law. We replaced $\widehat{W}$ and $\hat{\rho}$. These values were associated with $\widetilde{W}$ and $\tilde{\rho}$ and would not result in large values since they were coupled with the I-PID controller. The modified update law used is shown in Equations (27) and (28).

$$\dot{\widehat{W}} = \gamma(E_i^T PBH(E) - \sigma\widehat{W}) \tag{27}$$

$$\dot{\hat{\rho}} = \kappa(E_i^T PBs - \beta\hat{\rho}) \tag{28}$$

Finally, the proposed control input is shown in Equation (29).

$$\begin{aligned} u(t) = \underbrace{\left(u(t-h) - \bar{M}\ddot{X}\right)}_{\text{Linearization of the Plant}} + \underbrace{\bar{M}\left(\ddot{X}_d + K_P e(t) + K_I \int e(\tau)d\tau + K_D\dot{e}(t)\right)}_{\text{Desired error dynamics}} \\ + \underbrace{\bar{M}\cdot\Delta\widehat{H}}_{\text{RBF neural network}} + \underbrace{\bar{M}\cdot\hat{\varepsilon}}_{\text{Adaptive robust term with filter}} \end{aligned} \tag{29}$$

3.5. Stability Analysis

This study used a modified neural network update law to generate the proposed control inputs and applied a reverse saturation filter to the adaptive robust term to remove unnecessary control inputs in the steady state. In this chapter, we prove that the proposed controller satisfies stability. The proposed controller was composed of three parts: an I-PID controller, an RBF neural network for compensated time-delay estimation error, and an adaptive robust term for increased robustness of the control system. To remove unnecessary control inputs, we used a reverse saturation filter. For this filter, the presence or absence of an adaptive robust term was determined based on the threshold value. Therefore, the stability proof was divided into two parts: above and below the threshold value.

First, we considered when the system error was larger than the threshold value. To investigate control system stability, this study defined the Lyapunov function candidate as in Equation (30).

$$V(t) = E_i^T PE_i/2 + \widetilde{W}^T\widetilde{W}/2\gamma + \tilde{\rho}^T\tilde{\rho}/2\kappa \tag{30}$$

The Equation (30) time derivative was Equation (31).

$$\dot{V}(t) = \dot{E}_i^T PE_i/2 + E_i^T P\dot{E}/2 + \widetilde{W}^T\dot{\widehat{W}}/\gamma + \tilde{\rho}^T\dot{\hat{\rho}}/\kappa \tag{31}$$

By applying the proposed control input from Equation (29), we derived Equation (32).

$$\begin{aligned} V(\dot{t}) &= E_i^T(A^T P + PA)E_i/2 - \sigma\left(\widetilde{W}^T\widetilde{W} + \widetilde{W}^T\widehat{W} + \widetilde{W}^T\overline{W}\right)/2 - \beta(\tilde{\rho}^T\tilde{\rho} + \tilde{\rho}^T\hat{\rho} + \tilde{\rho}^T\bar{\rho}) \\ &\le E_i^T P(A^T + A + \lambda_m I - \lambda_m I)E_i/2 - \sigma\left(\widetilde{W}^T\widetilde{W} - \overline{W}^T\overline{W}\right)/2 - \beta(\tilde{\rho}^T\tilde{\rho} - \bar{\rho}^T\bar{\rho})/2 \\ &= -\lambda_m E_i^T PE_i/2 + E_i^T P(A^T + A + \lambda_m I)E_i/2 - \sigma\left(\widetilde{W}^T\widetilde{W} - \overline{W}^T\overline{W}\right)/2 \\ &\qquad - \beta(\tilde{\rho}^T\tilde{\rho} - \bar{\rho}^T\bar{\rho})/2 \end{aligned} \tag{32}$$

If λ_m satisfied $\lambda_m > 0$, I was a 3×3 unit matrix. The control gain was equal to $\sigma \ge \lambda_m/\gamma$ and $\beta \ge \lambda_m/\kappa$, and we could obtain Equation (33).

$$\begin{aligned} \dot{V}(t) &\le -\lambda_m E_i^T PE_i/2 + E_i^T P\lambda_m IE_i/2 - \sigma\left(\widetilde{W}^T\widetilde{W} - \overline{W}^T\overline{W}\right)/2 - \beta\left(\tilde{\rho}^T\tilde{\rho} - \bar{\rho}^T\bar{\rho}\right)/2 \\ &\le -\lambda_m(E_i^T PE_i/2 + \widetilde{W}^T\widetilde{W}/2\gamma + \tilde{\rho}^T\tilde{\rho}/\kappa) + E_i^T P\lambda_m IE_i/2 + \lambda_m\overline{W}^T\overline{W}/2 + \lambda_m\bar{\rho}^T\bar{\rho}/2 \\ &\le -\lambda_m V + \Phi \end{aligned} \tag{33}$$

Defining $\Phi = E_i^T P\lambda_m IE_i/2 + \lambda_m\overline{W}^T\overline{W}/2 + \lambda_m\bar{\rho}^T\bar{\rho}/2$ and utilizing Equation (33), we could derive Equation (34).

$$V(t) \le (V(t_0) - \Phi/\lambda_m)e^{-\lambda_m(t_N - t_0)} + \Phi/\lambda_m \tag{34}$$

t_0 represents the initial time and t_N denotes the time when the system error reaches a threshold value. Through Equations (3)–(34), we could verify that the proposed control system was bounded when it was larger than the threshold value.

Next, we considered when the system error was lower than the threshold value. To verify stability, we considered the Lyapunov function candidate as in Equation (35).

$$V(t) = E_i^T PE_i/2 + \widetilde{W}^T\widetilde{W}/2\gamma \tag{35}$$

The Equation (35) time derivative was Equation (36).

$$\dot{V}(t) = \dot{E}_i^T P E_i/2 + E_i^T P \dot{E}/2 + \widetilde{W}^T \dot{\widehat{W}}/\gamma \tag{36}$$

Using the proposed control input Equation (29) without the adaptive robust term, we obtained Equation (37).

$$\begin{aligned} V(\dot{t}) &= E_i^T(A^T P + PA)E_i/2 - \sigma\big(\widetilde{W}^T\widetilde{W} + \widetilde{W}^T\widehat{W} + \widetilde{W}^T\overline{W}\big)/2 + E_i^T PB\varepsilon \\ &\le E_i^T P(A^T + A + \lambda_m I - \lambda_m I)E_i/2 - \sigma\big(\widetilde{W}^T\widetilde{W} - \overline{W}^T\overline{W}\big)/2 \\ &= -\lambda_m E_i^T P E_i/2 + E_i^T P(A^T + A + \lambda_m I)E_i/2 - \sigma\big(\widetilde{W}^T\widetilde{W} - \overline{W}^T\overline{W}\big)/2 \end{aligned} \tag{37}$$

When below the threshold value, it was assumed that ε approximated zero. If λ_m satisfied $\lambda_m > 0$, I was a 3 × 3 unit matrix. The control gain was equal to $\sigma \ge \lambda_m/\gamma$, and we could obtain Equation (38).

$$\begin{aligned} \dot{V}(t) &\le -\lambda_m E_i^T P E_i/2 + E_i^T P\lambda_m I E_i/2 - \sigma\left(\widetilde{W}^T\widetilde{W} - \overline{W}^T\overline{W}\right)/2 \\ &\le -\lambda_m(E_i^T P E_i/2 + \widetilde{W}^T\widetilde{W}/2\gamma) + E_i^T P\lambda_m I E_i/2 + \lambda_m\overline{W}^T\overline{W}/2 \\ &\le -\lambda_m V + \breve{\Phi} \end{aligned} \tag{38}$$

Defining $\breve{\Phi} = E_i^T P\lambda_m I E_i/2 + \lambda_m\overline{W}^T\overline{W}/2$ and utilizing Equation (37), we could derive Equation (39).

$$V(t) \le (V(t_N) - \breve{\Phi}/\lambda_m)e^{-\lambda_m(t-t_N)} + \breve{\Phi}/\lambda_m \tag{39}$$

According to Equation (39), it was bounded even below the threshold value. Using the reverse saturation filter, this system was not smooth and had two bounds. However, as both cases were bounded, the whole control system was confirmed to be asymptotically stable [29,30,40–42].

4. Simulation

4.1. Simulation Setup

Computer simulations were performed to verify the performance of the proposed controller. The quadrotor parameters [43] used in the simulation are listed in Table 1.

Table 1. The quadrotor parameters.

Parameter	Description	Value
m	Mass of quadrotor	0.5 [kg]
I_x	Moment of inertia about X_B	0.0023 [kg · m^2]
I_y	Moment of inertia about Y_B	0.0023 [kg · m^2]
I_z	Moment of inertia about Z_B	0.0051 [kg · m^2]
l	Distance between center of the quadrotor and the propeller	0.17 [m]
b	Thrust factor	0.00018 [N · m^2]
J_r	Total moment of inertia of motor	0.000065 [kg · m^2]
g	Acceleration of gravity	9.81 [m/s^2]

The control parameters are shown in Table 2, and the sampling time was 0.001 s. All control parameters were determined through iterative simulations. The parameter determination process was a step-by-step tuning process starting from I-PID. The RBF neural network initial weight $W = [0.1, 0.01, 0.1, 0.01, 0.1]$, c = $[-1, 0.5, 0, 0.5, 1]$, and $b = 15$.

Table 2. The proposed controller parameters.

Parameter	Value

k_P	8
k_I	0.002
k_D	1.6
$\bar{M}$	0.02
P	$\begin{bmatrix} 28.5 & 0.5 \\ 0.5 & 28.5 \end{bmatrix}$
γ	0.7
σ	0.2
μ_1	0.5
μ_2	0.2
κ	0.01
β	0.001
N	0.1

The relationship between the control input and the angular velocity of the rotor was defined as Equation (40).

$$\begin{bmatrix} u_z \\ u_\phi \\ u_\theta \\ u_\psi \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ -lb & 0 & lb & 0 \\ 0 & -lb & 0 & lb \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{40}$$

where d is the drag factor and $\bar{\Omega} = \omega_1 - \omega_2 + \omega_3 - \omega_4$. The path equation set to follow a continuous helical path in the simulation is given in Equation (41). In this simulation, effects such as disturbances, model uncertainty, and noise were considered lumped disturbances. Therefore, the lumped disturbance used a complex disturbance including disturbance proportional to the velocity term, a sinusoidal signal, and a constant bias, and was defined as Equation (42) and applied to the quadrotor.

$$\begin{gathered} x_r = (1 - e^{-0.5t}) \cdot \sin(t) \\ y_r = (1 - e^{-0.5t}) \cdot \cos(t) \\ z_r = 0.1t \\ \psi_r = 1 \end{gathered} \tag{41}$$

$$\begin{gathered} D_x = k_{drag} \cdot \dot{x} + 2\sin(1.5t) + 0.8 \\ D_y = k_{drag} \cdot \dot{y} + 4\sin(2t) + 0.7 \\ D_z = k_{drag} \cdot \dot{z} + 3\sin(3t) + 1.5 \\ D_\phi = k_{drag} \cdot \dot{\phi} + 3\sin(2t) + 0.9 \\ D_\theta = k_{drag} \cdot \dot{\theta} + 3\sin(3t) + 0.7 \\ D_\psi = k_{drag} \cdot \dot{\psi} + 3\sin(1.5t) + 0.8 \end{gathered} \tag{42}$$

k_{drag} was set at 0.9, defining the drag coefficient. The threshold value for input saturation was 10.

Quadrotor systems inherently pose an underactuated problem, which complicates position control using conventional controllers. Typically, quadrotor systems resolve this issue by employing coupling equations that enable movement along the x and y axes through control inputs ϕ_d and θ_d. There are research examples for this coupling equation [8–10,15,16,43]. One method to achieve this is through the relationship with dynamics and another method is to design a controller. This study utilized a coupling equation for ϕ_r and θ_r, as presented in Equation (43), to avoid additional controller design.

$$sin\phi = \frac{\ddot{x}sin\psi - \ddot{y}cos\psi}{v_z}, \qquad tan\theta = \frac{\ddot{x}cos\psi + \ddot{y}sin\psi}{\ddot{z} + g} \tag{43}$$

The quadrotor, θ, and ϕ needed to remain around zero to maintain the quadrotor attitude. Additionally, to track the x and y axes, we utilized the error signal and the coupling equation, such as Equation (44), similar to the existing quadrotor PD control system.

$$\phi_d = \frac{x_e sin\psi - y_e cos\psi}{v_e}, \qquad \theta_d = \frac{x_e cos\psi + y_e sin\psi}{z_e + g} \tag{44}$$

The whole quadrotor control diagram for simulation is shown in Figure 3. The quadrotor dynamics part contains thruster information and input saturation.

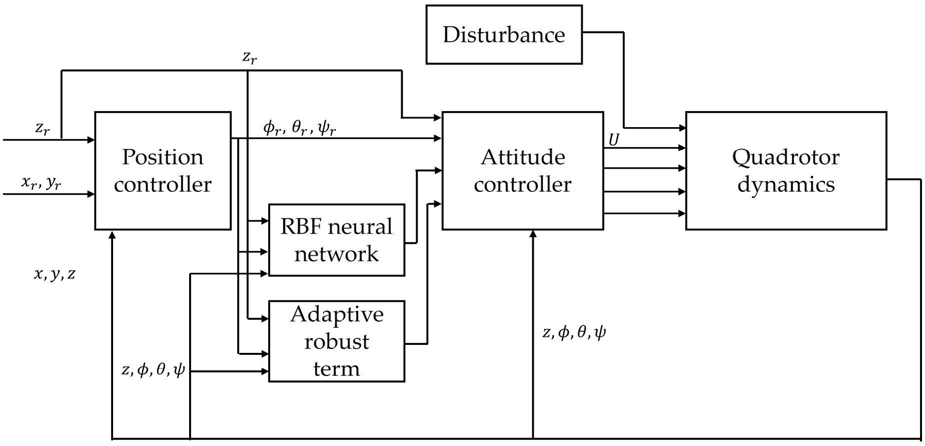

Figure 3. Quadrotor control system structure.

4.2. Simulation Results

This chapter presents the simulation results using the parameters and control system covered in Section 4.1 The initial value of the quadrotor was $x = 0.3, y = 0.2, z = 0, \phi = 0.05, \theta = 0.035, \psi = 0$.

Figure 4 illustrates the tracking performance of both the proposed controller and the PID control when following the reference trajectory. As indicated by the results, both controllers were capable of maintaining effective tracking performance, even in environments with disturbances. These outcomes confirm that both the traditional PID control system and the coupling equations used effectively managed the control tasks. However, the PID control was slightly wider in the x and y axes in the reference trajectory than the proposed controller. This means that the proposed controller showed better tracking performance compared to the PID control. Therefore, the next section discusses the position and posture tracking performance of the proposed controller and PID control.

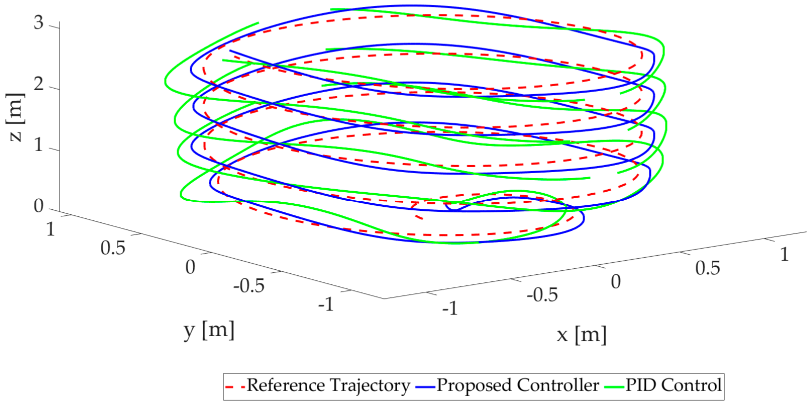

Figure 4. Comparison of 3D trajectory tracking performance using the proposed controller and PID control.

4.2.1. Position Control Simulation Results

This chapter presents the position control simulation results using the proposed controller. Figure 5 illustrates the position tracking performance of the proposed controller, while Figure 6 displays the performance of the PID control. The results depicted in Figures 5 and 6 demonstrate good tracking performance across all three axes. First, we see that the tracking performance for the X-axis of the proposed controller converged to the reference trajectory at 1.145 s. The maximum error that occurred was 0.05, which shows that the error was well suppressed. On the other hand, with the PID control, the X-axis converged to the reference trajectory at 1.532 s. The maximum error that occurred was 0.125. Next, the Y-axis tracking performance of the proposed controller showed that it converged to the reference trajectory in 1.449 s. The maximum error was 0.221, which was larger than for the X-axis error but smaller overall, showing satisfactory performance. The performance of the PID control was also satisfactory, converging to the reference trajectory in 1.844 s. The maximum error that occurred was 0.236. Z-axis tracking performance was different from the x-axis and y-axis. Based on the maximum error amplitude occurring below 0.2, the tracking performance of the proposed controller was found to reach near the reference trajectory in 1.5 s. The error was 0.117, and it can be seen that it operated around the reference trajectory. PID control also operated based on the reference trajectory and converged in 6.316 s, with an error of 0.161. A detailed comparison can be seen in the error comparison in Figure 7. As can be seen from the position control results, the x-axis tracking performance was noticeably improved, and the impact on the y and z axes was slightly improved, showing that all three axes could be improved.

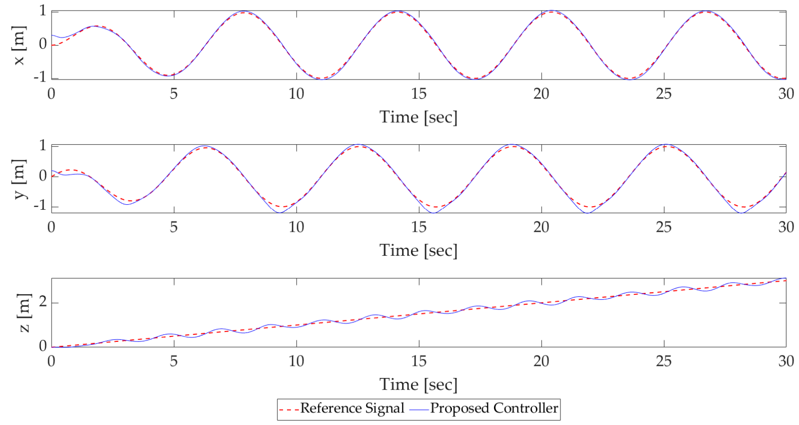

Figure 5. Position tracking performance of the proposed controller.

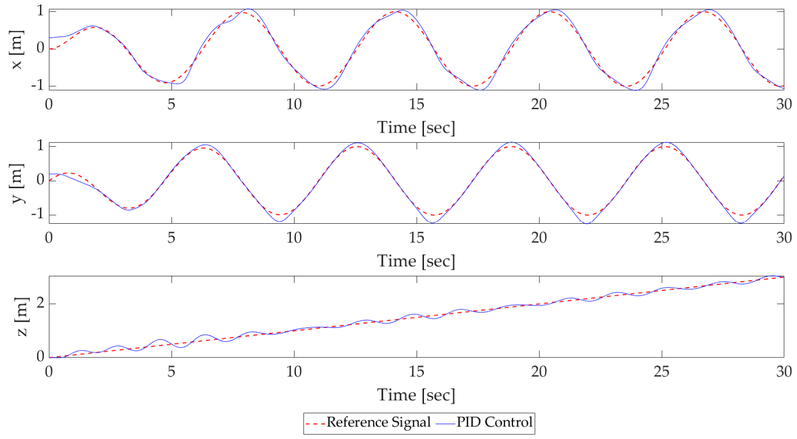

Figure 6. Position tracking performance of the PID control.

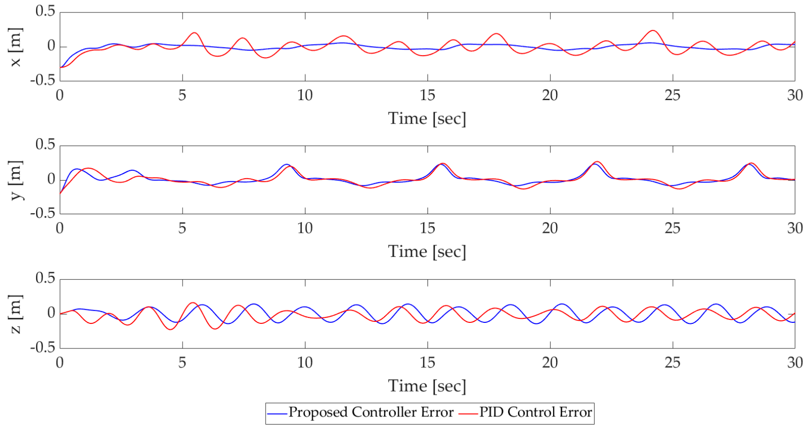

Figure 7. Comparison of the position error performance of the proposed controller and PID control.

4.2.2. Attitude Control Simulation Results

This chapter presents the attitude control simulation results for both the proposed controller and the traditional PID control. The reference signal ϕ_d, θ_d was generated by the position PD controller. Consequently, the expression of the reference signal varied based on the tracking performance of each controller. Figures 8 and 9 illustrate the attitude tracking performance of the proposed controller and the PID control, respectively, while Figure 10 compares the attitude error performance between the two controllers. At ϕ, the proposed controller began to converge in 1.9 s and showed excellent tracking performance, with a maximum error of 0.008. On the other hand, although the PID control could respond to changes in signals, it did not have excellent tracking performance. The maximum error was 0.173. At θ, it converged from about 2.6 s. The maximum error occurred at 0.25. After 2.6 s, the tracking performance error rapidly decreased and converged. In the case of the PID control, the convergence speed was 2 s faster than that of the proposed controller. However, compared to the proposed controller, the tracking performance was insufficient, and a large error occurred during operation, with a maximum error of 0.142.

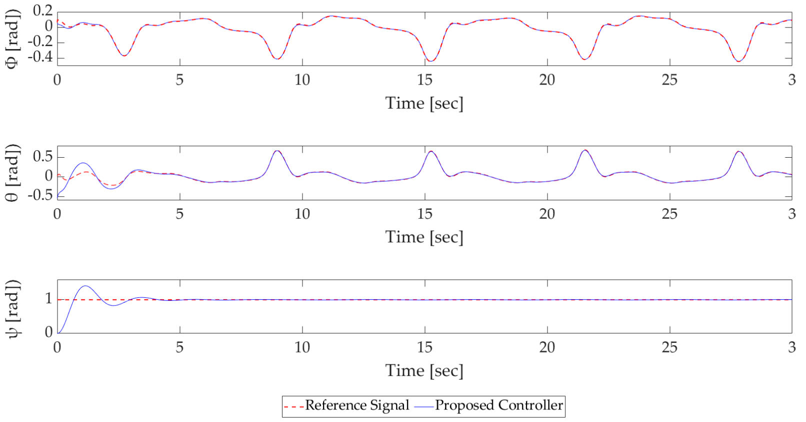

Figure 8. Attitude tracking performance of the proposed controller.

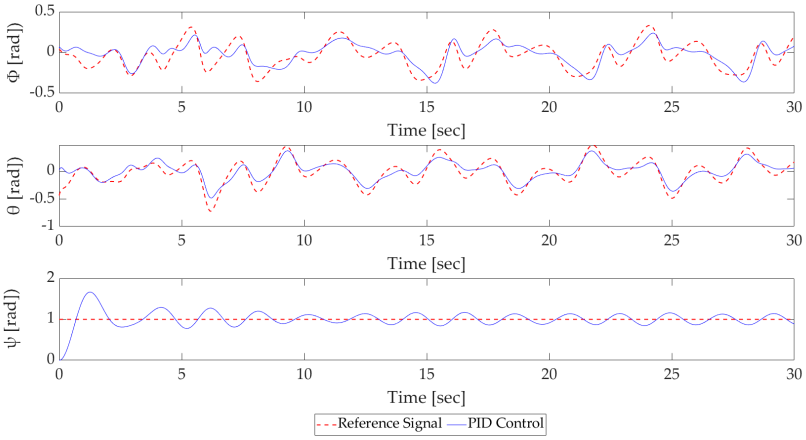

Figure 9. Attitude tracking performance of the PID controller.

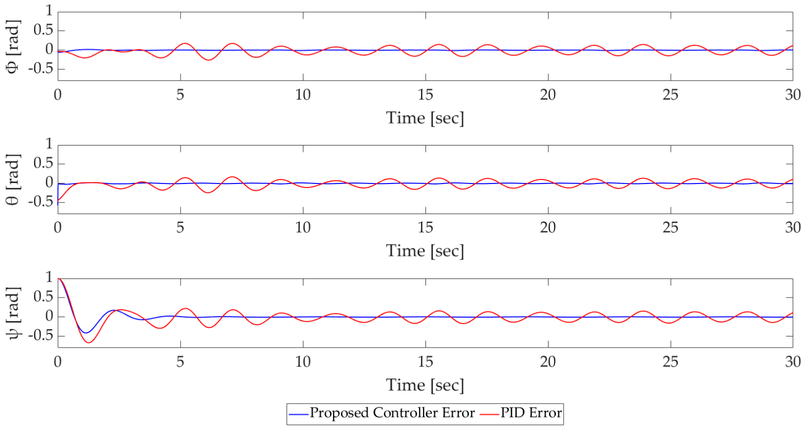

Figure 10. Comparison of the attitude error performance of the proposed controller and PID control.

ψ exhibited similar behavior characteristics to a second-order system. The maximum %overshoot was 39% and converged to a steady state around 2.7 s. After convergence, it had high tracking performance. For the PID control, the maximum %overshoot was 66% and ranged between 0.088 and 0.117 from around 8.5 s.

Figure 11 shows the control input comparison results when the proposed controller and the quadrotor system using PID control tracked the reference trajectory. The control input aspect was similar for both controllers. In particular, as shown in U_θ, the proposed controller could generate large inputs at the beginning of an operation that required a lot of tracking performance or when control inputs were required. Additionally, when entering a steady state, it could be confirmed that phenomena such as chattering caused by unnecessary control input did not occur due to the reverse saturation filter.

Consequently, the proposed control system enabled the design of a control system based on a model-free technique similar to the conventional PID control, and it was confirmed that it could enhance tracking performance.

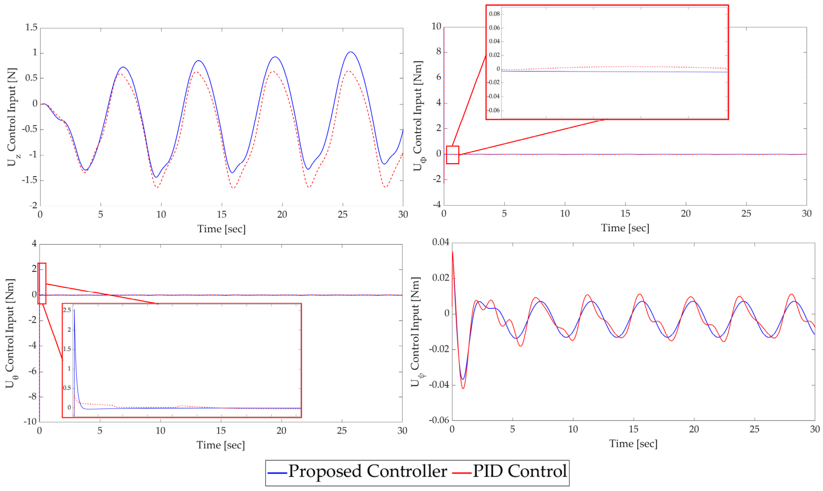

Figure 11. Comparison of the control input using the proposed controller and PID control.

5. Conclusions

In this paper, an I-PID controller was utilized that was similar to the PID controller widely used in previous quadrotor systems and showed high robustness. To enhance the robustness of the control system, the time-delay estimation error was compensated for with an RBF neural network. To address the shortcomings of neural network systems and construct a more robust controller, this paper proposed an adaptive robust term. This additional control input included an adaptative term to respond to large signal changes. It was combined with a reverse saturation filter to remove unnecessary control input in a steady state. The adaptive law of the proposed controller was designed based on Lyapunov stability to satisfy stability. The stability of the entire control system was proven by examining the boundary based on the reverse saturation filter section.

To verify the control system, simulations were performed on a quadrotor navigating a spiral path, subjected to 6-axis lumped disturbances. A conventional PID controller, commonly employed in quadrotor systems, served as the baseline for comparison. The simulation results demonstrated that the overall tracking performance of the proposed controller was superior to that of the PID control. With the proposed controller, convergence was achieved within 3 s across all six axes, with the error being reduced to as low as 0.23. Consequently, the controller introduced in this paper could be configured similarly to the conventional PID controller, yet it exhibited satisfactory performance.

In the future, we plan to make a quadrotor system to experimentally verify the controller and then conduct tracking experiments using a GPS (Global Positioning System) and hovering performance tests in situations where disturbances exist.

Author Contributions: S.-J.K. contributed to the conception of the study, designed the control system, performed the experiment, performed the data analyses, and wrote the manuscript; J.-H.S. contributed to the conception of the study, helped to write the manuscript, helped perform the analysis, and participated in constructive discussions corresponding the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under the Industrial Technology Innovation Program, grant no. 20026194, "Development of Human-Life Detection and Fire-Suppression Solutions based on Quadruped Robots for Firefighting and Demonstration of Firefighting Robots and Sensors".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Petrlík, M.; Báča, T.; Heřt, D.; Vrba, M.; Krajník, T.; Saska, M. A Robust UAV System for Operations in a Constrained Environment. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2169–2176.
2. Huang, Y.; Meng, Z. Bearing-based distributed formation control of multiple vertical take-off and landing UAVs. *IEEE Trans. Control Netw.* **2021**, *8*, 1281–1292.
3. Hu, J.; Zhang, H.; Song, L.; Schober, R.; Poor, H.V. Cooperative Internet of UAVs: Distributed trajectory design by multi-agent deep reinforcement learning. *IEEE Trans. Commun.* **2020**, *68*, 6807–6821.
4. Xu, L.; Qin, K.; Tang, F.; Shi, M.; Lin, B. UDE-Based Dynamic Surface Control for Quadrotor Drone Attitude Tracking under Non-Ideal Actuators. *Drones* **2023**, *7*, 305.
5. He, Z.; Hu, J.; Wang, Y.; Cong, J.; Bian, Y.; Han, L. Attitude-Tracking Control for Over-Actuated Tailless UAVs at Cruise Using Adaptive Dynamic Programming. *Drones* **2023**, *7*, 294.
6. Shen, Z.; Tsuchiya, T. Singular Zone in Quadrotor Yaw–Position Feedback Linearization, *Drones* **2022**, *6*, 84.
7. Yogi, S.C.; Tripathi, V.K.; Behera, L. Adaptive Integral Sliding Mode Control Using Fully Connected Recurrent Neural Network for Position and Attitude Control of Quadrotor. *IEEE Trans. Netw. Learn. Syst.* **2021**, *32*, 5595–5609.
8. Hu, M.; Ahn, H.; You, K. Finite-Time Rapid Global Sliding-Mode Control for Quadrotor Trajectory Tracking. *IEEE Access* **2023**, *11*, 22364–22375.
9. Sun, J.; Wang, Y.; Yu, Y.; Sun, C. Nonlinear Robust Compensation Method for Trajectory Tracking Control of Quadrotors. *IEEE Access* **2019**, *7*, 26766–26776.
10. Ullah, S.; Mehmood, A.; Khan, Q.; Rehman, S.; Iqbal, J. Robust Integral Sliding Mode Control Design for Stability Enhancement of Under-actuated Quadrotor. *Int. J. Control. Autom. Syst.* **2020**, *18*, 1671–1678.
11. Yu, G.; Reis, J.; Silvestre, C. Quad Neural Network Adaptive Control: Design and Experimental Validation. *IEEE Robot. Autom. Lett.* **2023**, *8*, 2574–2581.
12. Ouyang, Y.; Xue, L.; Dong, L.; Sun, C. Neural Network-Based Finite-Time Distributed Formation-Containment Control of Two Layer Quadrotor UAVs. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 4836–4848.
13. Liu, H.; Xi, J.; Zhong, Y. Robust Attitude Stabilization for Nonlinear Quadrotor Systems with Uncertainties and Delay. *IEEE Trans. Ind. Electron.* **2017**, *64*, 5585–5595.
14. Tayebi, A.; McGilvray, S. Attitude stabilization of a VTOL quadrotor aircraft. *IEEE Trans. Control Syst. Technol.* **2006**, *14*, 562–571.
15. Alexis, K.; Nikolakopoulos, G.; Tzes, A. Switching model predictive attitude control for a quadrotor helicopter subject to atmosphere disturbance. *Control Eng. Pract.* **2011**, *10*, 1195–1207.
16. Li, J.; Zhang, G.; Zhang, X.; Zhang, W. Integrating Dynamic Event-Triggered and Sensor-Tolerant Control: Application to USV-UAVs Cooperative Formation System for Maritime Parallel Search. *IEEE Trans. Intell. Transp. Syst.* **2023**, 1–13.
17. Li, J.; Zhang, G.; Zhang, W.; Shan, Q.; Zhang, W. Cooperative Path Following Control of USV-UAVs Considering Low Design Complexity and Command Transmission Requirements. *IEEE Trans. Intell. Transp. Syst.* **2023**, *9*, 715–724.
18. Fliess, M.; Join, C. Intelligent PID controller. In Proceedings of the 2008 16th Mediterranean Conference on Control and Automation, Ajaccio, France, 25–27 June 2008; pp. 326–331.
19. Wang, J.; Mounier, H.; Cela, A.; Niculescu, S. Event-driven intelligent PID controllers with applications to motion control, In Proceedings of the 18th IFAC World Congress, Milan, Italy, 28 August–2 September 2011; pp. 10080–10085.
20. Choe, Y. Optimal Tuning Strategy for 2-Degree-of-Freedom I-PID Controllers. *Trans. Korean Inst. Electr. Eng.* **2018**, *67*, 1202–1209.
21. Jung, S. A Neural Network Technique of Compensating for an Inertia Model Error in a Time-delayed Controller for Robot Manipulators, *Int. J. Control Autom. Syst.* **2020**, *18*, 1863–1871.
22. Han, S.; Tian, Y.; Christov, N.; Wang, H. Time-delay estimation based computed torque control with robust adaptive RBF neural network compensator for a rehabilitation exoskeleton. *ISA Trans.* **2020**, *97*, 171–181.
23. Slama, S.; Errachdi, A.; Benrejeb, M. Neural adaptive PID and neural indirect adaptive control switch controller for nonlinear MIMO systems. *Math. Probl. Eng.* **2019**, *2019*, 7340392.
24. Ge, S.; Wang, C. Direct adaptive NN control of a class of nonlinear systems. *IEEE Trans. Neural Netw.* **2002**, *15*, 214–221.
25. Ong, Q.; Yin, L. Robust adaptive fault accommodation for a robot system using a radial basis function neural network. *Int. J. Syst. Sci.* **2010**, *32*, 195–204.
26. Liu, J. *Intelligent Control Design and MATLAB Simulation*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 113–233.
27. Seghouane, A.; Shokouhi, N. Adaptive Learning for Robust Radial Basis Function Networks. *IEEE Trans. Cybern.* **2021**, *51*, 2847–2856.
28. Liu, J. *Radial Basis Function (RBF) Neural Network Control for Mechanical Systems: Design, Analysis and Matlab Simulation*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 1–280.
29. Kim, S.; Jin, M.; Suh, J. A Study on the Design of Error-Based Adaptive Robust RBF Neural Network Back-Stepping Controller for 2-DOF Snake Robot's Head. *IEEE Access* **2023**, *11*, 23146–23156.
30. Kim, S.; Suh, J. Adaptive Robust RBF-NN Nonsingular Terminal Sliding Mode Control Scheme for Application to Snake Robot's Head for Image Stabilization. *Appl. Sci.* **2023**, *13*, 4899.

31. Wan, M.; Chen, M.; Lungu, M. Integral Backstepping Sliding Mode Control for Unmanned Autonomous Helicopters Based on Neural Networks. *Drones* **2023**, *7*, 154.
32. Cao, X.; Shi, P.; Li, Z.; Liu, M. Neural-network-based adaptive backstepping control with application to spacecraft attitude regulation. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 4303–4313.
33. Zhang, Y.; Peng, P.; Jiang, Z. Stable neural controller design for unknown nonlinear systems using backstepping. *IEEE Trans. Neural Netw.* **2000**, *11*, 1347–1360.
34. Wang, W.; Kuo, M.; Lee, T.; Hong, C.; Leu, Y. RBF neural network adaptive backstepping controllers for MIMO nonaffine nonlinear systems. *IEEE Int. Conf. Syst. Man Cybern.* **2009**, 4946–4951.
35. Zhu, J.; Cao, Z.; Zhang, T.; Yang, Y.; Yi, Y. Sufficient Condition for the Existence of the Compact Set in the RBF Neural Network Control. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 3277–3282.
36. Bouadi, H.; Mora-Camino, F. Modeling and adaptive flight control for quadrotor trajectory tracking. *J. Aircr.* **2017**, *55*, 1–16.
37. Aboudonia, A.; El-Badawy, A.; Rashad, R. Disturbance observer-based feedback linearization control of an unmanned quadrotor helicopter. *Proc. Inst. Mech. Eng. Part J. Syst. Control. Eng.* **2016**, *230*, 877–891.
38. Bouadi, H.; Bouchoucha, M.; Tadjine, M. Sliding Mode Control based on Backstepping Approach for an UAV Type-Quadrotor. *Int. J. Mech. Mechatron. Eng.* **2007**, *1*, 39–44.
39. Jung, S. An Impedance Force Control Approach to a Quad-Rotor System Based on an Acceleration-Based Disturbance Observer. *J Intell Robot Syst.* **2014,** *73,* 175–185.
40. Fang, Y.; Fei, J.; Ma, K. Model reference adaptive sliding mode control using RBF neural network for active power filter. *Electr. Power Energy Syst*. **2015**, *73*, 249–258.
41. Liu, Q.; Li, D.; Ouyang, Z.; Tee, K.; Ge, S. Adaptive bias RBF neural network control for a robotic manipulator. *Neurocomputing* **2021**, *447*, 213–223.
42. Wang, L.; Chai, T.; Zhai, L. Neural-Network-Based Terminal Sliding-Mode Control of Robotic Manipulators Including Actuator Dynamics. *IEEE Trans. Ind. Electron*. **2009**, *56*, 3296–3304.
43. Hwang, N.; Kim, B. Active Fault Tolerant Control of Quadrotor Based on Multiple Sliding Surface Control Method. *J. KIECS* **2022**, *17*, 59–70.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.