Hindawi
International Journal of Aerospace Engineering
Volume 2019, Article ID 1510341, 9 pages
https://doi.org/10.1155/2019/1510341

Research Article

Research on Adaptive Trajectory Tracking Algorithm for a Quadrotor Based on Backstepping and the Sigma-Pi Neural Network

Zhiming Chen,[1] **Kang Niu**,[2] **and Lei Li**[1]

[1]*Micro Satellite Research Center, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*
[2]*Shanghai Institute of Space Technology Research Center, Shanghai 201109, China*

Correspondence should be addressed to Kang Niu; kangniu@nuaa.edu.cn

Received 7 August 2018; Revised 18 December 2018; Accepted 26 January 2019; Published 27 March 2019

Academic Editor: Paolo Castaldi

In this paper, adaptive tracking control is applied to improve performances of an underactuated quadrotor helicopter with respect to attitude and position control. Firstly, the dynamic model is presented. Then a new trajectory tracking algorithm is designed by using the sigma-pi neural network and backstepping. The paper designs the sigma-pi neural network compensation control law and gives the Lyapunov-type stability analysis. Then the corresponding numerical simulations are performed by using MATLAB. Simulation results are shown to demonstrate the effectiveness of the proposed control strategy, which could reduce tracking error, decrease tracking time, and improve the anti-interference ability of the system.

1. Introduction

Presently, the quadrotor is very popular since it can not only vertically take-off and land but also swiftly manoeuvre in any direction [1]. However, the quadrotor is well known to be a nonlinear, underactuated, and highly coupled system. Therefore, how to design a flying control system has become a challenging research.

In recent years, researchers have proposed many control methods for the quadrotor system, such as PID method and LQR [2–5]. Although those methods can control UAVs well in some respect, the tracking accuracy performance is terrible when the speed of the UAV increases or the UAV performs large maneuvering. In addition to the above methods, the signal compensation method is used to improve attitude control performances of roll and pitch channels in the condition of a large maneuvering flight [6]. Reference [7] introduces the LQT to improve tracking performance, but this method mainly focuses on decreasing energy consumption. The feedback linearization is used to divide the quadrotor system into two fully actuated subsystems [8], but this method focuses on stabilizing the quadrotor instead of tracking a desired trajectory. The gain scheduling control method is used to get acceptable performance, but this method is the severe trade-off between control performance and the number of the required trim points [9]. Backstepping technique with command filtering is used to finish the control of trajectory tracking, and it uses second-order filter quaternion to get the desired angular velocity vector [10]. The adaptive backstepping sliding mode control technique is used to finish the attitude control, which can reduce the tremor phenomenon of the first-order fixed gain sliding mode [11]. Reference [12] uses the integrator backstepping approach in position and yaw trajectory tracking of the quadrotor. Reference [13] derives the BIOAC (backstepping-based inverse optimal attitude controller) to solve the attitude rapidly located problem, when the input limit is in a large manoeuvre condition.

In recent years, many researchers apply the neural network to UAV control. Reference [14] discusses the BP neural network to train the data and construct the dynamics of the UAV. Reference [15] uses the only INS data and neural network to generate hover command and applies this method on JR Ergo 60. The multilayer feedforward network (MFN) is designed to detect obstacles in highways and uses

backpropagation algorithm to supervise the training of the proposed neural network by minimizing the square error function via descending gradient criteria [16]. Reference [17] designs an adaptive inverse controller to realize UAV formation flight, combining with the BP neural network. Reference [18] designs an adaptive sliding mode control based on two neural networks for quadrotor stabilization, which presents solutions to conventional control drawbacks as chattering phenomenon and dynamical model imprecision.

Combining the sigma-pi neural network with backstepping, this paper presents an algorithm of trajectory tracking control for quadrotor UAVs. The sigma-pi neural network control law is established to compensate for the trajectory tracking error. At the same time, the paper gives the proof of stability by using Lyapunov function. The comparative simulation shows that the tracking error will be reduced considerably, the tracking time will be shortened, and the jitter of the attitude angle will be decreased.

The paper is organized as follows. The nonlinear quadrotor UAV dynamic model and the sigma-pi neural network are given in Section 2. In Section 3, the sigma-pi neural network control law is designed and the stability analysis of the control law will also be given. Some comparative simulations are given in Section 4. At last, Section 5 will conclude this paper.

2. Quadrotor Dynamic Modelling

2.1. Coordinate Definition. To describe the UAV motion, the Earth coordinate frame S_E and the body coordinate frame S_B are used in this paper. At the same time, in the paper, S_E is assumed as the inertial coordinate and the Earth's rotation is negligible. S_B is fixed to the quadrotor body. The detailed information of those two reference coordinate frames are shown in Figure 1.

As shown in Figure 1, $(F_1\ F_2\ F_3\ F_4)$ represent the rotor thrust. $(T_1\ T_2\ T_3\ T_4)$ represent the moment generated by each rotor. $(\varphi\ \theta\ \psi)$ are Euler angles. m_g represents the gravity.

2.2. Quadrotor Dynamic Modelling and Parameters. When modelling the quadrotor UAV, the paper considers that the UAV is just affected by gravity, rotor thrust, and aerodynamic drag during the flight. Using Newton's kinematics law, the dynamic equations can be written in the following form:

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} \dfrac{u_x U_1}{m} - \dfrac{f_x \dot{x}}{m} \\ \dfrac{u_y U_1}{m} - \dfrac{f_y \dot{y}}{m} \\ \dfrac{-g + u_z U_1}{m} - \dfrac{f_z \dot{z}}{m} \\ \dfrac{(I_y - I_z)\dot{\theta}\dot{\psi}}{I_x} + \dfrac{U_2 d}{I_x} \\ \dfrac{(I_z - I_x)\dot{\varphi}\dot{\psi}}{I_y} + \dfrac{U_3 d}{I_y} \\ \dfrac{(I_x - I_y)\dot{\theta}\dot{\varphi}}{I_z} + \dfrac{U_4}{I_z} \end{pmatrix}, \tag{1}$$

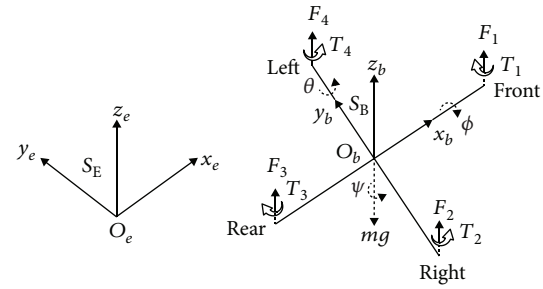

Figure 1: Reference coordinate frames S_E and S_B.

where $(f_x f_y f_z)$ denotes the aerodynamic drag coefficient. U_i $(i = 2, 3, 4)$ represents the resultant moment. $U_1 = (\sum_{i=1}^{4} F_i)$ represents the resultant force. $(I_x\ I_y\ I_z)$ represents the moment of inertia about the x-axis, y-axis, and z-axis. u_x, u_y, and u_z represent U_1 in the x, y, z direction, which is equal to the following formulas, respectively.

$$\begin{aligned} u_x &= (\cos(\varphi)\sin(\theta)\cos(\psi) + \sin(\varphi)\sin(\psi)), \\ u_y &= (\cos(\varphi)\sin(\theta)\sin(\psi) - \sin(\varphi)\sin(\psi)), \\ u_z &= (\cos(\varphi)\cos(\theta)). \end{aligned} \tag{2}$$

3. Sigma-Pi Neural Network and Control Law Design

3.1. Sigma-Pi Neural Network. Compared with other neural networks, the sigma-pi neural network uses summation neurons and quadrature neurons to construct the hidden layers. It not only preserves the highly nonlinear mapping ability but also increases the flexibility of the network. So, according to the specific problems, which can construct appropriate neural networks to improve the learning efficiency of the neural network.

Figure 2 takes a three-layer *sigma-pi neural network* as an example, which includes the input layer, product layer, and output layer. The input x is an N-dimensional vector and x_k is the k_{th} component of x. The inputs are weighted and fed to a layer of K linear product units, where K is the desired order of the network. Let h_{ji} be the output of the j_{th} product units for the i_{th} output, y_i. Then

$$\begin{cases} h_{ji} = \prod_k w_{kji} x_k \beta + \theta_{ji}, \\ y_i = \sigma\left(\sum_j h_{ji}\right), \end{cases} \tag{3}$$

where w_{kji} is an adjustable weight from input x_k to the j_{th} product units of the i_{th} output and θ_{ji} is an adjustable threshold of the j_{th} product units of the i_{th} output. β represents the basis function vector. $\sigma(x)$ denotes the nonlinear activation function and is selected as the logistic function, $\sigma(x) = 1/(1 + e^{-x})$ for all the results reported in this paper. In this paper, it should be noted that connections from product units to an output have fixed weights. Thus, there is no notion of hidden units in the network and fast-learning rules can be used.

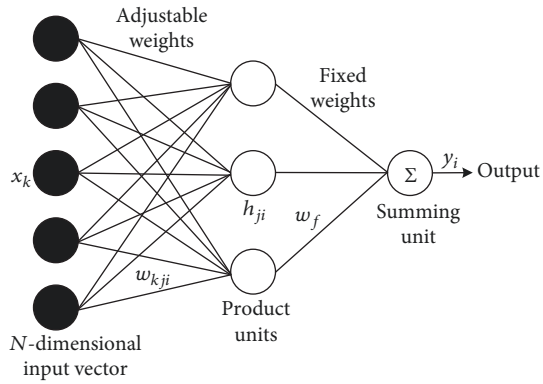

Figure 2: A sigma-pi neural network.

In this paper, the signal layer sigma-pi neural network which just have the input layer and product units to compensate for the error of attitude and position will be used. Then the attitude compensation control law will be designed first, and in this section, the detailed derivation process and the Lyapunov stability analysis will be provided. Followed by using the same method, this paper will give the position compensation control law.

3.2. Adaptive Control Law Design. Combing with the backstepping control method, this paper introduces the sigma-pi neural network into the position and attitude control system to improve trajectory tracking performances. The control loop with sigma-pi neural network compensation control is shown in Figure 3, which consists of the command filter module, position control module, attitude control module, dynamic model of the UAV, and sigma-pi neural network.

As can be seen in Figure 3, the "state" outputted by the dynamic model consists of $(x, y, z, \dot{x}, \dot{y}, \dot{z}, \varphi, \theta, \psi, \dot{\varphi}, \dot{\theta}, \dot{\psi})$. Before designing the sigma-pi neural network compensation attitude and position control law, it is more convenient to establish a state-space model to design a control system for the quadrotor UAV. Then the state-space system for the UAV dynamic model described in (1) can be expressed as follows.

3.3. Attitude Control Law Design. In this section, the attitude control law will be designed. As shown in Figure 4, which is part of Figure 3, the quadrotor attitude control loop is composed of the command filter, attitude control module, and sigma-pi neural network module.

As can be seen in Figure 4, the attitude angle control instruction $(\varphi_d, \theta_d, \psi_d)$ will be filtered by the command filter module. Then it combines the states of the quadrotor UAV to get the attitude errors. In addition, the sigma-pi neural network module will combine the states of the quadrotor UAV with U_{ang} which is the current attitude control input $U_{\text{ang}} = [U_2, U_3, U_4]^{\mathrm{T}}$ to get the compensation control input $U_{\text{S-P}}$. Then combining $U_{\text{S-P}}$ with attitude errors, the attitude control module will calculate the control input for the roll, pitch, and yaw of the quadrotor. Using these attitude control inputs (U_1, U_2, U_3, U_4), the dynamic model calculates the parameter in each channel to control the attitude and output states of the quadrotor UAV. And the "state" outputted by the dynamic model contains $(x, y, z, \dot{x}, \dot{y}, \dot{z}, \varphi, \theta, \psi, \dot{\varphi}, \dot{\theta}, \dot{\psi})$.

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{pmatrix} = \begin{pmatrix} x_2 \\ \dfrac{(I_y - I_z)\dot{\theta}\dot{\psi}}{I_x} + \dfrac{U_2 d}{I_x} \\ x_4 \\ \dfrac{(I_z - I_x)\dot{\varphi}\dot{\psi}}{I_y} + \dfrac{U_3 d}{I_y} \\ x_6 \\ \dfrac{(I_x - I_y)\dot{\theta}\dot{\varphi}}{I_z} + \dfrac{U_4}{I_z} \\ x_8 \\ \dfrac{u_x U_1}{m} - \dfrac{f_x \dot{x}_8}{m} \\ x_{10} \\ \dfrac{u_y U_1}{m} - \dfrac{f_y \dot{y}_{10}}{m} \\ x_{12} \\ \dfrac{-g + u_z U_1}{m} - \dfrac{f_z \dot{z}_{12}}{m} \end{pmatrix}. \tag{4}$$

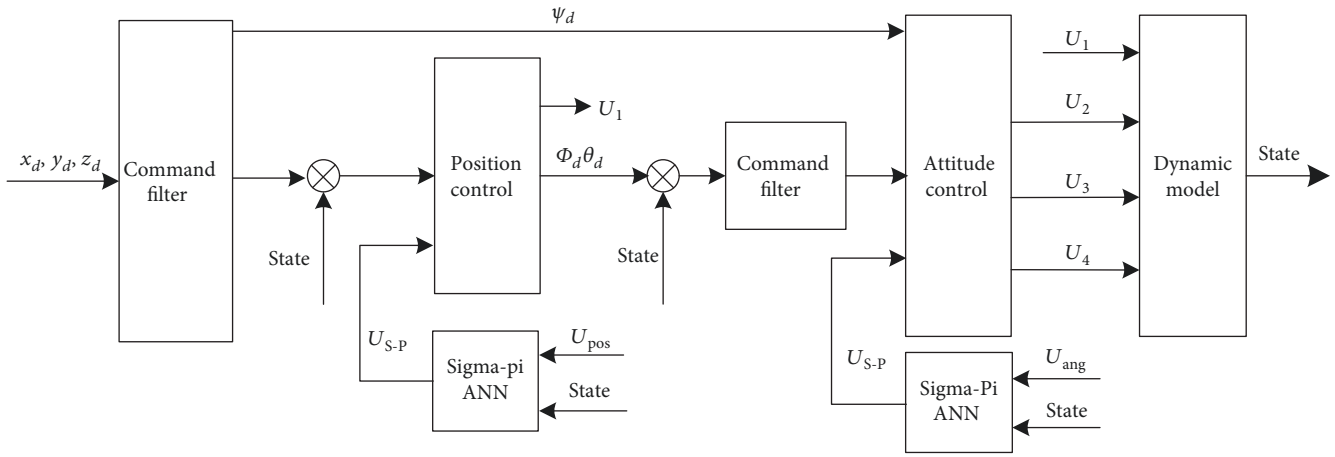

Figure 3: The control loop with sigma-pi neural network compensation.

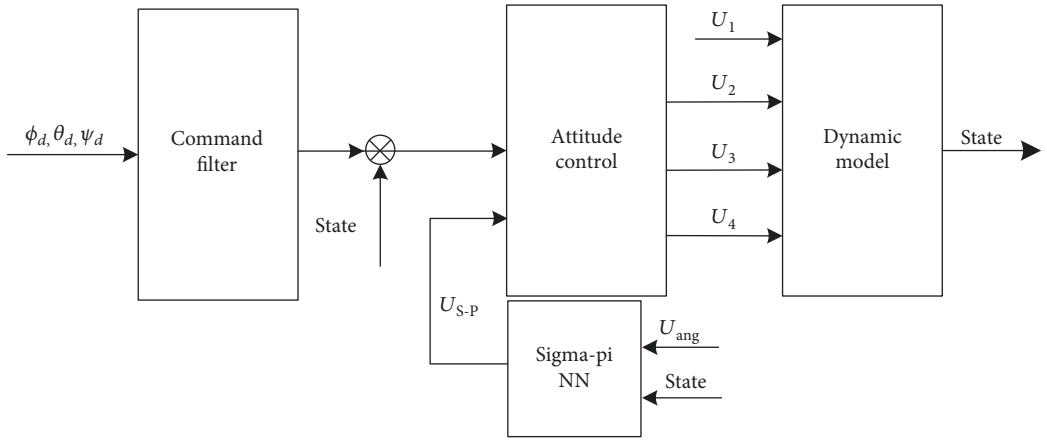

Figure 4: The quadrotor attitude control loop.

Now, this paper begins designing the attitude control law. Considering the modelling error and external disturbances, equation (4) can be shown in following equation.

$$\begin{cases} \dot{X}_1 = X_2, \\ \dot{X}_2 = f(X_1, X_2, U) + U_{S\text{-}P} - \tilde{e^{\sim}}, \end{cases} \tag{5}$$

where $X_1 = [\varphi\, \theta\, \psi\, x\, y\, z]^T$; $e^{\sim} = [e_\varphi\, e_\theta\, \tilde{e}_\psi\, e_x\, e_y\, e_z]^T$ denotes the modelling error and external disturbances; $U_{S\text{-}P}$ represents the sigma-pi control input.

In this paper, the single-layer sigma-pi neural network is used to compensate for the error. So, $U_{S\text{-}P}$ can be expressed as follows.

$$U_{S\text{-}P} = W^T\beta, \tag{6}$$

where W^T denotes the weight coefficient matrix; β represents the basis function vector which is defined as follows:

$$\beta = \text{kron}[\text{kron}[C_1, C_2], C_3], \tag{7}$$

where "kron" represents the Kronecker product and in the attitude control system, C_1, C_2, C_3 are defined as follows:

$$\begin{cases} C_1 = \left[0.01\, \dot{w}\, \dot{w}^2\right]^T, \\ C_2 = \left[0.01\, \varphi_d\theta_d\, \psi_d\, U_{S\text{-}P}^{\varphi}\, U_{S\text{-}P}^{\theta}\, U_{S\text{-}P}^{\psi}\right]^T, \\ C_3 = [\varphi\, \theta\, \psi]^T. \end{cases} \tag{8}$$

To design the adaptive control law, the paper defines that U^* is the optimal sigma-pi neural network control input. Then we will get the following equation.

$$\begin{aligned} U^* &= W^{*T}\beta, \\ U_{S\text{-}P} - U^* &= W^T\beta - W^{*T}\beta = \tilde{W}^{\sim *T}\beta, \end{aligned} \tag{9}$$

where W^{*T} is the optimal weight coefficient matrix; then the error between the optimal control input and the modelling error can be expressed as follows:

$$U_{\text{S-P}} - e^{\sim} = \tilde{W}^{\sim *T}\tilde{\beta} + U^{*} - e^{\tilde{\sim}}. \tag{10}$$

Now, this paper takes the roll channel as an example to design the appropriate adaptive control law and give the Lyapunov-type stability analysis. Firstly, we define the roll angle $x_1(\varphi)$ tracking error as follows:

$$z_1 = x_{1d} - x_1. \tag{11}$$

Then the Lyapunov-type equation $V(z_1)$ and the appropriate first-order differential equation $\dot{V}(z_1)$ can be defined as the following equations:

$$V(z_1) = \frac{1}{2}z_1^2, \tag{12}$$

$$\dot{V}(z_1) = z_1(\dot{x}_{1d} - x_2). \tag{13}$$

According to equation (13), x_2 can be defined as $x_2 = \dot{x}_{1d} + \alpha_1 z_1 (\alpha_1 > 0)$ to satisfy the stability of the Lyapunov function. Then $\dot{V}(z_1) = -\alpha_1 z_1^2 < 0$.

The roll angle rate $\dot{x}_1(\varphi)$ error can be defined using the same method $z_2 = x_{2d} - x_2$. Then we can define the Lyapunov function which is rated to z_1 and z_2.

$$V(z_1, z_2) = \frac{1}{2}z_1^2 + \frac{1}{2}z_2^2 + \frac{1}{2\gamma}\tilde{W}_{\varphi}^{\sim T}\tilde{W}^{\sim}_{\varphi}. \tag{14}$$

Then the first order differential equation $\dot{V}(z_1, z_2)$ can be defined as the following equation:

$$\begin{cases} \dot{V}(z_1, z_2) = z_1\dot{z}_1 + z_2\dot{z}_2 + \dfrac{1}{\gamma_{\varphi}}\tilde{W}_{\varphi}^{\sim T}\dot{\tilde{W}}^{\sim}_{\varphi}, \\ z_2 = x_{2d} - x_2. \end{cases} \tag{15}$$

Combining with equations (4), (5), and (11) and defining $x_{2d} = \dot{x}_{1d} + \alpha_1 z_1$, $\dot{x}_2$ in equation (4) can be expressed as follows:

$$\dot{x}_2 = x_4 x_6 a_1 + b_1 U_2 + \tilde{W}_{\varphi}^{\sim T}\beta_{\varphi} + U_{\text{S-P}}^{*\varphi} - e_{\varphi}, \tag{16}$$

where $a_1 = (I_y - I_z)/I_x$ and $b_1 = d/I_x$.

Then substituting equation (16) into (15) can get the following equation:

$$\begin{cases} \dot{V}(z_1, z_2) = z_1\dot{z}_1 + z_2\dot{z}_2 + \dfrac{1}{\gamma_{\varphi}}\tilde{W}_{\varphi}^{\sim T}\dot{\tilde{W}}^{\sim}_{\varphi}, \\ \dot{z}_2 = \ddot{x}_{1d} + \alpha_1\dot{z}_1 - \dot{x}_2, \\ \dot{x}_2 = x_4 x_6 a_1 + b_1 U_2 + \tilde{W}_{\varphi}^{\sim T}\beta_{\varphi} + U_{\text{S-P}}^{*\varphi} - e_{\varphi}. \end{cases} \tag{17}$$

According to (17), we can select the control input U_2 and the sigma-pi neural network compensation control law as follows:

$$\begin{cases} U_2 = \dfrac{1}{b_1}\left(z_1 + \alpha_1 z_2 - \alpha_1^2 z_1 - x_4 x_6 a_1 + \alpha_2 z_2\right), \\ \alpha_1 > 0, \\ \alpha_2 > 0, \\ \dot{\tilde{W}}^{\sim}_{\varphi} = -\gamma_{\varphi} z_2 \beta_{\varphi}. \end{cases} \tag{18}$$

Substituting equation (18) into (17), the first-order differential Lyapunov function $\dot{V}(z_1, z_2)$ is as follows:

$$\begin{aligned} \left\{\dot{V}(z_1, z_2) = -\alpha_1 z_1^2 - \alpha_2 z_2^2 + z_2\left(U_{S\_P}^{*\varphi} - e_{\varphi}\right) \le -\alpha_1 z_1^2 - \alpha_2 z_2^2 + \varepsilon|z_2|\right. \\ = -\alpha_1 z_1^2 - |z_2|(\alpha_2|z_2| - \varepsilon). \end{aligned} \tag{19}$$

According to (18), if $|z_2|$ satisfies $|z_2| > \varepsilon/\alpha_2$, the Lyapunov function is $\dot{V}(z_1, z_2) < 0$. The system is stable. Otherwise, the system is not. If $|z_2|$ does not satisfy $|z_2| > \varepsilon/\alpha_2$, then the paper defines the sigma-pi adaptive control law $\dot{\tilde{W}}^{\sim}_{\varphi} = 0$.

Using the same method, we can get the adaptive sigma-pi neural network control law of the pitch and the yaw channel.

$$\begin{cases} U_3 = \dfrac{1}{b_2}\left(z_3 + \alpha_3 z_4 - \alpha_3^2 z_3 - x_2 x_6 a_3 + \alpha_4 z_4\right), \\ \dot{\tilde{W}}^{\sim}_{\theta} = -\gamma_{\theta} z_4 \beta_{\theta}, \\ U_4 = \dfrac{1}{b_3}\left(z_5 + \alpha_5 z_6 - \alpha_5^2 z_5 - x_4 x_2 a_5 + \alpha_6 z_6\right), \\ \dot{\tilde{W}}^{\sim}_{\psi} = -\gamma_{\psi} z_6 \beta_{\psi}. \end{cases} \tag{20}$$

Using the same approaches of the roll channel, then z_3, z_4, z_5, z_6 are defined as follows:

$$\begin{cases} z_3 = x_{3d} - x_3, \\ z_4 = x_{4d} - x_4 (x_{4d} = \dot{x}_{3d} + \alpha_3 z_3 (\alpha_3 > 0)), \\ z_5 = x_{5d} - x_5, \\ z_6 = x_{6d} - x_6 (x_{6d} = \dot{x}_{6d} + \alpha_6 z_6 (\alpha_6 > 0)). \end{cases} \tag{21}$$

3.4. Position Control Law Design. To reduce the trajectory tracking error, this paper also adds the sigma-pi neural network to quadrotor position control loop which is shown in Figure 5, which is part of Figure 3. As can be seen in Figure 5, the position control loop is also composed of the command filter module, position control module, and sigma-pi neural network module.

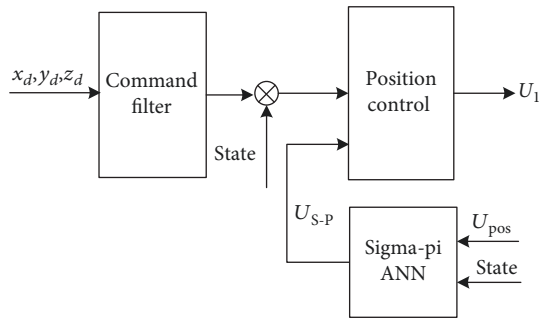

Figure 5: The quadrotor position control loop.

As can be seen in Figure 5, (x_d, y_d, z_d) represent the desired position instruction, which will be filtered by the command filter module. The "state" outputted by the dynamic model contains $(x, y, z, \dot{x}, \dot{y}, \dot{z}, \varphi, \theta, \psi, \dot{\varphi}, \dot{\theta}, \dot{\psi})$. It is different from the attitude control loop and the position control loop which defines the basis function vector β as follows:

$$\beta = \text{kron}\left[\text{kron}\left[C_{p1}, C_{p2}\right], C_{p3}\right], \tag{22}$$

where "kron" represents Kronecker product and in the position control loop, C_{p1}, C_{p2}, C_{p3} are defined as follows:

$$\begin{cases} C_{p1} = \left[0.01\ d\ d^2\right]^T, \\ C_{p2} = \left[0.01 x_d y_d z_d xyz U_{S\text{-}P}^x U_{S\text{-}P}^y U_{S\text{-}P}^z\right]^T, \\ C_{p3} = \left[xyz\right]^T, \end{cases} \tag{23}$$

where d in (23) denotes the distance error between the desired position and the current states.

Using the same method, we can get the sigma-pi neural network compensation control law in the x, y, z direction.

$$\left\{ \dot{\tilde{W}}_x = -\gamma_x z_x \beta_x, \dot{\tilde{W}}_y = -\gamma_y z_y \beta_y, \dot{\tilde{W}}_z = -\gamma_z z_z \beta_z, \right. \tag{24}$$

where z_x, z_y, z_z represent the position error.

After deriving the formula, integrating equations (18), (20), and (24) can get the sigma-pi neural network compensation control law of attitude and position.

$$\begin{cases} W = \left[W_\varphi, W_\theta, W_\psi\right]^T, \\ W = \left[W_x, W_y, W_z\right]^T. \end{cases} \tag{25}$$

Then substituting equations (7), (22), and (25) into equation (6) can get the sigma-pi neural network compensation of the system.

Table 1: Parameters using in simulation.

Parameter	Definition	Value
x_d (m)	Desired x	t
y_d (m)	Desired y	$3\cos\frac{t}{2}$
z_d (m)	Desired z	$3\sin\frac{t}{2}+3$
(x_0, y_0, z_0) (m)	Initial position	(0,1,3)

4. Simulation and Discussion

Some comparative simulations are carried out to illustrate the trajectory tracking algorithms proposed in this paper, which can decrease the tracking error, improve the tracking precision, reduce the tracking time, and improve the antijamming capability. The first case wants the UAV to track the 3D spiral trajectory by using the backstepping method as a contrast. The second case will add the disturbing items into the dynamic model to illustrate the ability of anti-interference.

5. 3D Spiral Trajectory Tracking

In this case, we will use the sigma-pi neural network compensation control method proposed in this paper and backstepping method in [19] to track the 3D spiral trajectory. The desired trajectory and initial position in simulation are shown in Table 1, and the simulation results are shown in Figures 6–9.

The trajectory tracking result is shown in Figures 6 and 7. Figure 6 uses the backstepping method only, and Figure 7 uses the sigma-pi neural network to compensate for the control error. By comparison, it can be seen clearly that the tracking performance of using the adaptive sigma-pi neural network is better than using backstepping only.

The curve of the attitude angle and tracking error by using these two methods can be seen in Figure 8. By contrast, it is clearly seen that the tracking error curve which has the sigma-pi neural network compensation control approach is obviously small especially in x and z directions. And to be more specific, the tracking error in the x, y, z direction will all converge to 0 at around 10 s.

Figure 9 gives the attitude angle change of the UAV by using these two methods. To be more specific, it is clear that the change range of the attitude by using the sigma-pi neural network compensation control method is smaller than backstepping. Apart from this, the jitter of the quadrotor UAV is significantly reduced before 5 s, when using the sigma-pi neural network compensation control method. Moreover, the curve of the attitude angle is smoother without mutation by using the sigma-pi neural network compensation control approach.

5.1. Disturbance Rejection Simulation. In this section, disturbance rejection properties of the using sigma-pi neural network compensation control method will be analyzed. This

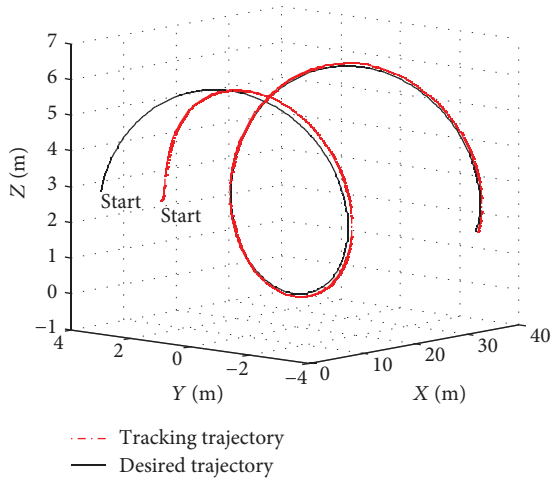

Figure 6: Simulation result using backstepping only.

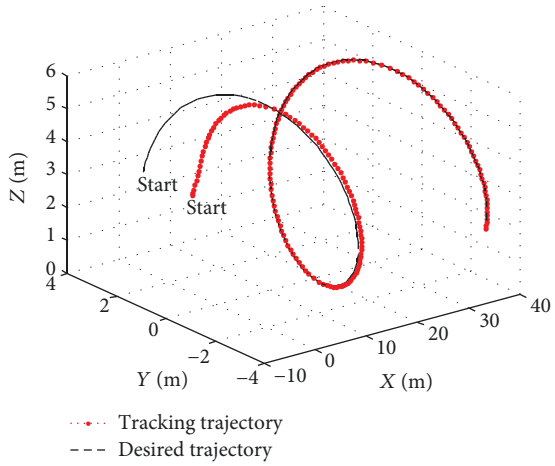

Figure 7: Simulation result using adaptive sigma-pi neural network control.

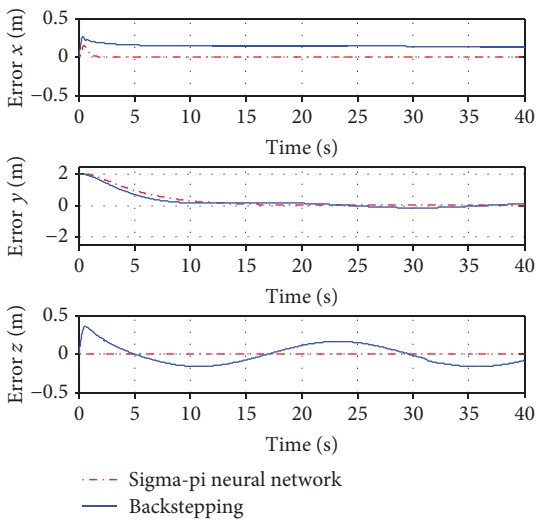

Figure 8: Curves of trajectory tracking error.

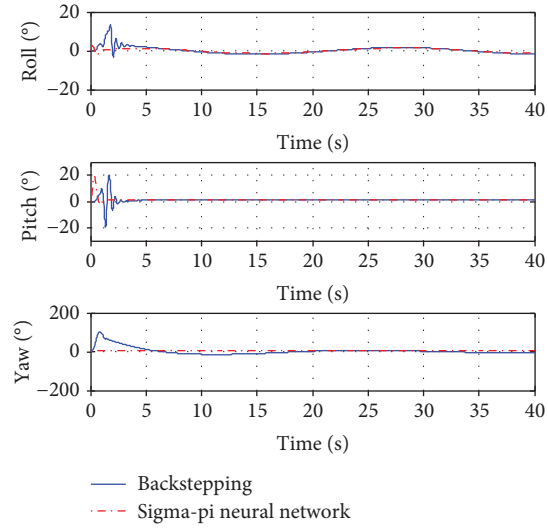

Figure 9: Curves of the trajectory attitude angle.

paper will regard gust force as one of the disturbance terms added to the quadrotor UAV. Then the equation of translation motion can be expressed as follows:

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} \dfrac{u_x U_1}{m} - \dfrac{f_x \dot{x}}{m} + d_x + U^x_{\text{S-P}} - \tilde{e}_x \\ \dfrac{u_y U_1}{m} - \dfrac{f_y \dot{y}}{m} + d_y + U^y_{\text{S-P}} - \tilde{e}_y \\ \dfrac{-g + u_z U_1}{m} - \dfrac{f_z \dot{z}}{m} + d_z + U^z_{\text{S-P}} - \tilde{e}_z \end{pmatrix}, \tag{26}$$

where d_x, d_y, and d_z represent additional disturbance terms in the x, y, z direction; as can be seen in equation (20), disturbances are added to the dynamics as additional external forces; therefore, they are in the units of (N). The disturbance term equation is shown as follows, and the corresponding disturbance force curve is shown in Figure 10.

$$V_w = \begin{cases} 0, & t < T, \\ \left(\dfrac{V_{g\max}}{2}\right)\left\{1 - \cos\left[2\pi\left(\dfrac{t - T_1}{T_g}\right)\right]\right\}, & T_1 \le t \le T_1 + T_g, \\ 0, & t > T_1 + T_g. \end{cases} \tag{27}$$

As can be seen in Figure 10, it is clear that for the motion in the x, y, and z direction, a strong constant disturbance is added to the system at 10, 20, and 30 seconds, respectively. Since constant disturbances are very strong, they last for 1 second; thus, they can be considered as impulse disturbances. All of these different sizes of disturbing external forces are added simultaneously in the whole simulation process.

In this simulation, the desired trajectory and initial position are shown in Table 2. And the trajectory tracking result, attitude angle, and the tracking error in disturbance circumstance are shown in Figures 11–13.

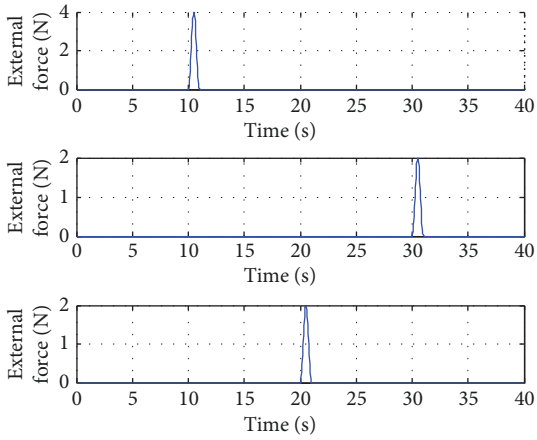

Figure 10: Disturbance term curves.

Table 2: Parameters used in this simulation.

Parameter	Definition	Value
x_d (m)	Desired x	t
y_d (m)	Desired y	$3\cos\frac{t}{2}$
z_d (m)	Desired z	$3\sin\frac{t}{2}+3$
(x_0, y_0, z_0) (m)	Initial position	(0,1,3)

As can be seen in Figure 11, disturbances are successfully rejected. At 10, 20, and 30 seconds, the quadrotor is exposed to strong impulse disturbances and deviates from the desired path. However, as can be seen in Figure 12, the system tries to return to the desired path by changing Euler angles significantly. And the change of Euler angles is very small in each direction.

Figure 12 gives the curve of the attitude angle. It is clear that when the additional disturbance terms occur in each direction at 10, 20, and 30 seconds, the quadrotor UAV briefly fluctuates and then restores stability soon. This illustrates that the sigma-pi neural network compensation control method has the ability of anti-inference.

Figure 13 shows the change of tracking error curves. It is clear that at 10, 20, and 30 seconds, despite the appearing error fluctuations, the tracking error is still very small in the whole process and soon converges to 0. Therefore, it also proves that the quadrotor UAV can track the desired trajectory well in disturbance circumstances.

6. Conclusions

This paper proposes a new trajectory tracking algorithm of a quadrotor UAV by introducing the sigma-pi neural network to backstepping. Firstly, the paper gives the quadrotor UAV dynamics. Then the paper establishes the sigma-pi neural network compensation control law and gives the Lyapunov stability analysis. At last, some contrastive simulations show that the method proposed in this paper is valid. By comparison, using the proposed method in this paper, we can improve the tracking performance, reduce the tracking time, improve the tracking accuracy, and decrease the jitter of the system.

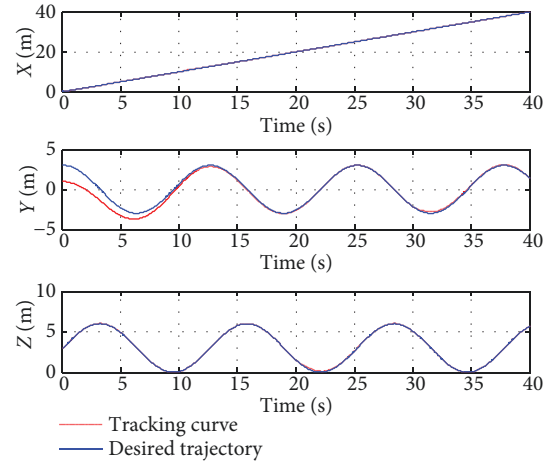

Figure 11: The curve of trajectory tracking.

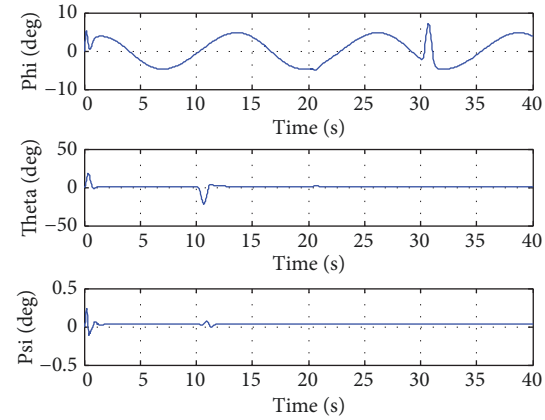

Figure 12: The curve of the attitude angle.

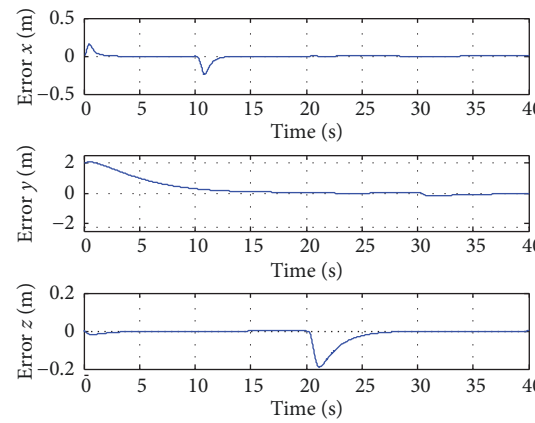

Figure 13: The curve of tracking error.

Data Availability

All data used in this paper are obtained by using MATLAB. Researchers can replicate the analysis and conduct secondary analyses. All data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no competing interests.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (no. 61673212) and the Natural Science Foundation of Jiangsu Province (no. BK20161490).

References

[1] P.-i. Pipatpaibul and P. R. Ouyang, "Application of online iterative learning tracking control for quadrotor UAVs," *ISRN Robotics*, vol. 2013, Article ID 476153, 20 pages, 2013.

[2] G. Szafranski and R. Czyba, "Different approaches of PID control UAV type quadrotor," in *Proceedings of the International Micro Air Vehicles conference*, pp. 70–75, 't Harde, Netherlands, 2011, summer edition.

[3] D. C. Tosun, Y. Işık, and H. Korul, "LQR control of a quadrotor helicopter," in *Proceedings of the International Conference on Pure Mathematics*, pp. 247–252, Wienna, Austria, 2014.

[4] K. Djamel, M. Abdellah, and A. Benallegue, "Attitude optimal backstepping controller based quaternion for a UAV," *Mathematical Problems in Engineering*, vol. 2016, Article ID 8573235, 11 pages, 2016.

[5] M. Kerma, A. Mokhtari, B. Abdelaziz, and Y. Orlov, "Nonlinear H_∞ control of a quadrotor (UAV), using high order sliding mode disturbance estimator," *International Journal of Control*, vol. 85, no. 12, pp. 1876–1885, 2012.

[6] X. Wang, Y. Chen, G. Lu, and Y. Zhong, "Robust attitude tracking control of small-scale unmanned helicopter," *International Journal of Systems Science*, vol. 46, no. 8, pp. 1–14, 2015.

[7] E. C. Suicmez and A. T. Kutay, "Optimal path tracking control of a quadrotor UAV," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 115–125, Orlando, FL, USA, May 2014.

[8] R. Bonna and J. F. Camino, "Trajectory tracking control of a quadrotor using feedback linearization," in *Proceedings of the XVII International Symposium on Dynamic Problems of Mechanics*, pp. 22–27, Natal, RN, Brazil, February 2015.

[9] Y. Choi, H. Jimenez, and D. N. Mavris, "Statistical gain-scheduling method for aircraft flight simulation," *Aerospace Science and Technology*, vol. 46, pp. 493–505, 2015.

[10] S. Zhao, W. Dong, and J. A. Farrell, "Quaternion-based trajectory tracking control of VTOL-UAVs using command fltered backstepping," in *2013 American Control Conference*, pp. 1018–1023, Washington, DC, USA, June 2013.

[11] G. Regula and B. Lantos, "Backstepping based control design with state estimation and path tracking to an indoor quadrotor helicopter," *Periodica Polytechnica Electrical Engineering*, vol. 53, no. 3-4, pp. 151–161, 2009.

[12] D. B. Lee, C. Nataraj, T. C. Burg, and D. M. Dawson, "Adaptive tracking control of an underactuated aerial vehicle," in *2011 American Control Conference*, pp. 2326–2331, San Francisco, CA, USA, June-July 2011.

[13] A. Honglei, L. Jie, W. Jian, W. Jianwen, and M. Hongxu, "Backstepping-based inverse optimal attitude control of quadrotor," *International Journal of Advanced Robotic Systems*, vol. 10, no. 5, p. 223, 2013.

[14] A. Sarotama and B. Kusumoputro, "Position difference for system identification and control of UAV alap-alap using back propagation algorithm neural network with Kalman filter," *American Journal of Intelligent Systems*, vol. 5, pp. 18–26, 2015.

[15] G. Buskey, G. Wyeth, and J. Roberts, "Autonomous helicopter hover using an artificial neural network," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 2, pp. 1635–1640, Seoul, Korea, May 2001.

[16] C. A. DAI Guzmán, A. A. Torres, and M. A. M. Bárcenas, "Design of an Artificial Neural Network to detect obstacles on highways through the flight of an UAV," *Research in Computing Science*, vol. 105, pp. 31–40, 2009.

[17] X. Y. Wang, X. M. Wang, and C. C. Yao, "Design of UAVs formation flight controller based on neural network adaptive inversion," *Control and Decision*, vol. 28, pp. 837–843, 2013.

[18] H. Boudjedir, F. Yacef1, O. Bouhali, and N. Rizoug, "Dual neural network for adaptive sliding mode control of quadrotor helicopter stabilization," *International Journal of Information Sciences and Techniques*, vol. 2, no. 4, pp. 1–14, 2012.

[19] E. C. Suiçmez, *Trajectory Tracking of a Quadrotor Unmanned Aerial Vehicle (UAV) Via Attitude and Position Control*, Educational & Industrial Television, 2014.