Mathematical Biosciences and Engineering

MBE, 18(2): 1774–1793.
DOI: 10.3934/mbe.2021092
Received: 10 November 2020
Accepted: 28 January 2021
Published: 19 February 2021

http://www.aimspress.com/journal/MBE

Research article

Optimized neural network based sliding mode control for quadrotors with disturbances

Ping Li *, Zhe Lin, Hong Shen, Zhaoqi Zhang and Xiaohua Mei

College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China

*** Correspondence:** Email: pli@hqu.edu.cn.

Abstract: In this paper, optimized radial basis function neural networks (RBFNNs) are employed to construct a sliding mode control (SMC) strategy for quadrotors with unknown disturbances. At first, the dynamics model of the controlled quadrotor is built, where some unknown external disturbances are considered explicitly. Then SMC is carried out for the position and the attitude control of the quadrotor. However, there are unknown disturbances in the obtained controllers, so RBFNNs are employed to approximate the unknown parts of the controllers. Furtherly, Particle Swarm optimization algorithm (PSO) based on minimizing the absolute approximation errors is used to improve the performance of the controllers. Besides, the convergence of the state tracking errors of the quadrotor is proved. In order to exposit the superiority of the proposed control strategy, some comparisons are made between the RBFNN based SMC with and without PSO. The results show that the strategy with PSO achieves quicker and smoother trajectory tracking, which verifies the effectiveness of the proposed control strategy.

Keywords: quadrotor; sliding mode control (SMC); radial basis function neural network (RBFNN); particle swarm optimization (PSO); disturbance

1. Introduction

Quadrotors have been widely used in military, photography, navigation and other application fields due to their properties of economy, portability and flexibility. Comparing with helicopter or airplane, quadrotors are with simpler structure, so they are more available for engineering [1].

However, quadrotors are under-actuated systems, and how to design appropriate controllers for them have been attracting substantial attention.

Sliding mode control (SMC) is a nonlinear robust control method, which is considered to be one of the effective methods against uncertainties and disturbances [2]. In [3], the principle of sliding mode control was introduced and specific control strategies were designed for a class of underactuated systems. In [4], comparisons were made between SMC and feedback linearization method, the results indicated SMC was more robust for noises and disturbances. In [5], the authors divided a quadrotor into fully actuated and underactuated subsystems, and second order SMC was employed for the design of the underactuated subsystems, whose coefficients were obtained by Hurwitz stability. In [6,7], terminal sliding mode control (TSMC) was designed for quadrotors, with which faster convergence of trajectory tracking was achieved. In [8], sliding mode observers were used to estimate all states of the quadrotor through the measurable attitude and position of the quadrotor, and PID SMC was employed to guarantee the convergence of the trajectory tracking. An integrated sliding mode control (ISMC) was proposed in [9], which divided the quadrotor model into inner and outer loops. The outer loop mainly generated reference signals for the roll and pitch angles, and the inner loop achieved the position and attitude tracking with ISMC. In [10], model-free TSMC was designed, where the model-free method guaranteed that the tracking error of the quadrotor was bounded, while the terminal sliding mode guaranteed the convergence of the error. In [11], adaptive SMC was proposed to realize finite-time stability of a quadrotor through self-tuning. In [12], a super twisting control algorithm was proposed to enforce the sliding mode of the attitude of a quadrotor on the desired manifold, therefore, the robustness of the attitude tracking was guaranteed. In order to realize the tracking of uncertain dynamical systems, an adaptive nonsingular fast terminal sliding-mode control (ANFTSMC) algorithm was proposed in [13]. In [14], the ANFTSMC was applied to a quadrotor which was subject to modeling uncertainties and unknown external disturbances. In [15], disturbance observer based adaptive SMC was applied to a quadrotor subjected to parametric uncertainties and external disturbances, the effectiveness of the control was tested by numerical simulations and experiments. In [16], the authors presented an adaptive dynamic surface SMC for a quadrotor, they used nonlinear observer to estimate the states of the quadrotor, and used minimum learning technology to reduce computational burden**,** the controller was designed to ensure all signals of the system were uniformly ultimately bounded. In [17], the position loop was controlled by backstepping method, and the attitude loop was controlled by the fast TSMC, both of these methods were combined with adaptive technology to estimate the controller parameters, and upper bounds of the uncertainties and disturbances.

Neural network has been developed rapidly in recent years, and it has been introduced into the control of quadrotors to deal with nonlinearities and uncertainties. The authors of [18] combined neural networks with SMC, by adjusting the parameters of the sliding surface with back propagation algorithm, the overshoot of the system response and the steady state error were effectively reduced. A PID neural network control strategy was proposed in [19], where proportional, integral, and differential neurons were defined to construct a network, by which control inputs were designed to achieve quicker response of a quadrotor. Adaptive radial basis function neural networks (RBFNNs) were employed in [20] for the control of a quadrotor. In [21], RBFNN was used to generate the control signals of a quadrotor, by which the characteristic roots of the quadrotor system located in the

left half plane, so the stability of the system was guaranteed. Backstepping design was combined with neural network in [22] to realize the control of a quadrotor with unknown input saturation. In [23] a learning-based scheme with neural networks was used to obtain the optimal control law for a quadrotor with time-varying and coupling uncertainties, the stability of the closed-loop system was proved by theory. The authors of [24] proposed RBFNNs based proportional derivative-sliding mode control (RPD-SMC) for the outer loop for position tracking, and employed the robust integral of the signum of error (RISE) to guarantee attitude convergence.

Optimization algorithms have been gradually employed to improve the performance of neural networks. Particle swarm optimization (PSO) algorithm base RBF network was designed in [25] to get better PID parameters. The authors of [26] used genetic algorithms to optimize RBFNNs for more accuracy modeling. In [27], genetic algorithm was also used for optimizing the hyperparameters of a two hidden layer neural network, by which the training and running time of the neural network was reduced. However, using the optimized neural networks in SMC to realize robust control of quadrotors has been rarely considered so far.

This paper aims to improve robust tracking performance of a quadrotor with unknown disturbances. SMC is employed to design the position and attitude control laws in the inner and outer loops, RBFNNs are introduced to approximate the unknown parts of the control laws. The weights of the RBFNNs are updated by designed adaptive laws. Besides, the center and the width values of the RBFNNs are optimized by PSO. The proposed strategy can guarantee the stability and robustness of the controlled quadrotor.

The rest of this paper is organized as follows: the dynamics model of a quadrotor is built in section 2. In Section 3, the position and attitude controllers are designed for the quadrotor based on RBFNN-SMC. PSO is conducted in section 4 to optimize the center and the width values of the RBFNNs. Simulation results are introduced in Section 5. Finally, Section 6 concludes the paper.

2. Dynamic modeling of a quadrotor

The physical structure of a quadrotor is presented in Figure 1. As the figure shows, there are two coordinate systems to describe the motion of the quadrotor [28], one is the body coordinate system, and the other is the inertial coordinate system. The transformation of the two coordinate systems can be realized by a rotation matrix. It depends on the three attitude angles of the quadrotor, that are the roll angle ϕ, the pitch angle θ, and the yaw angle ψ. All the attitude angles are bounded, the roll angle and the pitch angle belong to $(-\pi/2, \pi/2)$, and the yaw angle is constrained by $(-\pi, \pi)$.

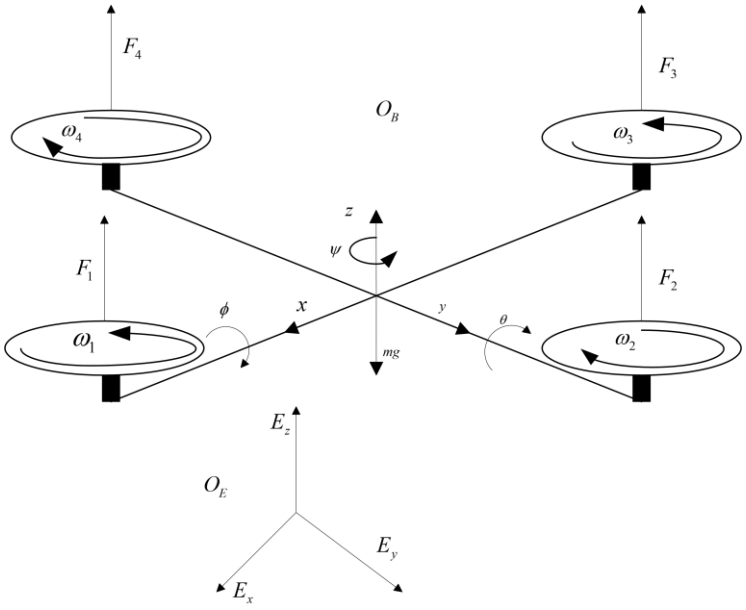

Figure 1. Structure Diagram of a Quadrotor.

Based on the attitude angles, the rotation matrix R_B^E which transforms the position of the quadrotor in body coordinate system to inertial coordinate system can be expressed as (2.1).

$$R_B^E = \begin{bmatrix} C\psi C\theta & -C\phi S\psi + S\phi S\theta C\psi & S\phi S\psi + C\phi S\theta C\psi \\ C\theta S\psi & C\phi C\psi + S\theta S\phi S\psi & -S\phi C\psi + C\phi S\theta S\psi \\ -S\theta & S\phi C\theta & C\phi C\theta \end{bmatrix} \tag{2.1}$$

where $C = cos \quad ,S = sin \quad .$

Let *x, y, z* be the position variables of the center of gravity of the quadrotor along the three axes of the inertial coordinate system. Define $w = [x, y, z]^T$, then from Newton's Second Law of Motion, one can gets that

$$m\ddot{w} = R_B^E \begin{bmatrix} 0 & 0 & u_1 \end{bmatrix}^T - [0 \quad 0 \quad mg]^T - f \tag{2.2}$$

where *m* is the mass of the quadrotor, *g* is the gravitational acceleration, u_1 is the lifting force generated by the rotation of the propellers, and *f* is the air friction during the flying of the quadrotor.

Owing to external disturbances existed, (2.2) should be rewritten as

$$m\ddot{w} = R_B^E \begin{bmatrix} 0 & 0 & u_1 \end{bmatrix}^T - [0 \quad 0 \quad mg]^T - f + d_p \tag{2.3}$$

where d_p is the unknown external disturbance.

Suppose that the quadrotor is a symmetric rigid body, and let *p, q, r* be the angular velocity of the quadrotor's rotation about the three axes of the body coordinate system. Define $\omega = [p, q, r]^T$, then based on the Newton–Euler equation, one can obtain that

$$J\frac{d}{dt}(\omega)+\omega\times J\omega=M \tag{2.4}$$

where $J=diag(I_x,I_y,I_z)$, is the diagonal matrix of moment of inertia, I_x,I_y,I_z denote the moment of inertia of the three axes, M is the total rotation torque. By considering the components of M, (2.4) can be rewritten as

$$J\frac{d}{dt}(\omega)=-\omega\times J\omega+M_i+M_f+M_p+d_r \tag{2.5}$$

where M_i represents the input torque, which is supplied by the rotation of the four propellers. M_f represents the air friction torque, M_p is gyroscopic effect on the quadrotor, and d_r denotes the unknown external disturbance.

Suppose the quadrotor is flying in small attitude angles, one has that $\dot{\phi}=p$, $\dot{\theta}=q$, $\dot{\psi}=r$ and $\ddot{\phi}=\dot{p}$, $\ddot{\theta}=\dot{q}$, $\ddot{\psi}=\dot{r}$, the state equations of the quadrotor dynamics can be derived from (2.3) and (2.5) as

$$\begin{cases}
\ddot{x}=\frac{u_1}{m_s}(\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi)-\frac{k_1}{m_s}\dot{x}+d_x\\
\ddot{y}=\frac{u_1}{m_s}(\cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi)-\frac{k_2}{m_s}\dot{y}+d_y\\
\ddot{z}=\frac{u_1}{m_s}\cos\phi\cos\theta-\frac{k_3}{m_s}\dot{z}-g+d_z\\
\ddot{\phi}=\mathrm{qr}\frac{I_y-I_z}{I_x}+\frac{J_r}{I_x}q\omega_r+\frac{l}{I_x}u_2-\frac{k_4l}{I_x}p+d_\phi\\
\ddot{\theta}=pr\frac{I_z-I_x}{I_y}-\frac{J_r}{I_y}p\omega_r+\frac{l}{I_y}u_3-\frac{k_5l}{I_y}q+d_\theta\\
\ddot{\psi}=pq\frac{I_x-I_y}{I_z}+\frac{C}{I_z}u_4-\frac{k_6l}{I_z}r+d_\psi
\end{cases} \tag{2.6}$$

where k_i (i = 1, ⋯, 6) are the air resistance coefficients, l represents the distance of each propeller to the center of the quadrotor, u_2,u_3,u_4 represent the input torques of roll, pitch and yaw respectively, C is the moment-force proportional coefficient, d_x, d_y, d_z, d_ϕ, d_θ, d_ψ denote the unknown disturbances to the corresponding states, J_r denotes the inertia of the propeller, $\omega_r=\omega_2+\omega_4-\omega_1-\omega_3$ is the total residual speed of the propellers, with ω_i being the speed of the i^{th} propeller, i = 1, ⋯, 4. The relationships among ω_i and u_i, i = 1, ⋯, 4, are formulated as (2.7).

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ b & 0 & -b & 0 \\ 0 & -b & 0 & b \\ -k & k & -k & k \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{2.7}$$

where b and k are thrust and drag coefficients, respectively.

3. Flight control design

This section introduces the design of sliding mode control with radial basis function neural networks. The control structure is shown in Figure 2.

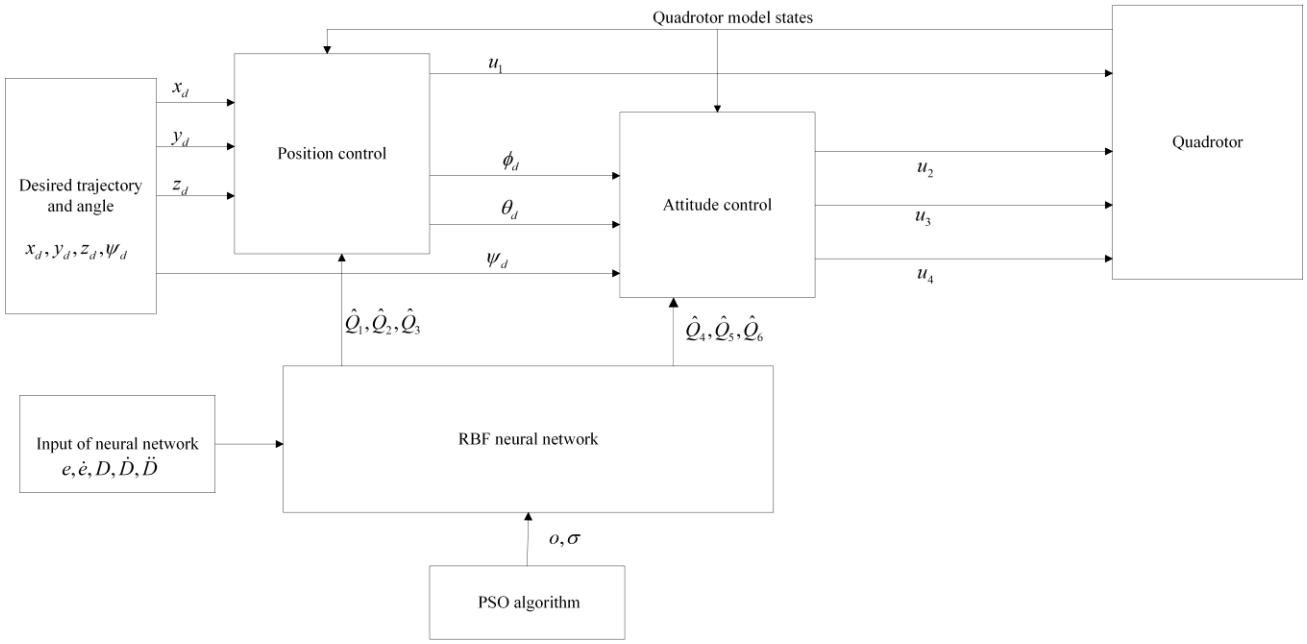

Figure 2. Control Structure of the Quadrotor.

3.1. Position control

At first, rewrite the state equation of the quadrotor's position as follows.

$$\begin{cases} \ddot{x} = \dfrac{u_x}{m} - \dfrac{k_1}{m}\dot{x} + d_x \\ \ddot{y} = \dfrac{u_y}{m} - \dfrac{k_2}{m}\dot{y} + d_y \\ \ddot{z} = \dfrac{u_z}{m} - \dfrac{k_3}{m}\dot{z} - g + d_z \end{cases} \tag{3.1}$$

with

$$\begin{cases} u_x = u_1(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \\ u_y = u_1(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \\ u_z = u_1\cos\phi\cos\theta \end{cases}$$

From the above definition, one can see that all the control inputs u_x, u_y, u_z are corresponding to the lifting force u_1 because the quadrotor is an underactuated system. In the design showing below, the control input u_z will be employed to derive the needed force u_1, and u_x, u_y are auxiliary inputs to obtain the desired roll and pitch angles.

As SMC is employed for the control design, three sliding surfaces are defined here as (3.2) - (3.4).

$$s_1 = c_1(x - x_d) + (\dot{x} - \dot{x}_d) = ce_x + \dot{e}_x \tag{3.2}$$

$$s_2 = c_2(y - y_d) + (\dot{y} - \dot{y}_d) = c_2e_y + \dot{e}_y \tag{3.3}$$

$$s_3 = c_3(z - z_d) + (\dot{z} - \dot{z}_d) = c_3e_z + \dot{e}_z \tag{3.4}$$

where $c_i > 0$ i = 1, 2, 3, are design coefficients, x_d, y_d, z_d are the desired position coordinates of the quadrotor, $e_x = x - x_d$, $e_y = y - y_d$, $e_z = z - z_d$ are the tracking errors between the actual and the desired trajectories.

According to the principle of SMC, if the control inputs are designed as follows,

$$u_x = m(-c_1\dot{x} + c_1\dot{x}_d + \ddot{x}_d + k_1\frac{\dot{x}}{m} - \tau_1 s_1 - \xi_1 sign(s_1) - d_x) \tag{3.5}$$

$$u_y = m(-c_2\dot{y} + c_2\dot{y}_d + \ddot{y}_d + k_2\frac{\dot{y}}{m} - \tau_2 s_2 - \xi_2 sign(s_2) - d_y) \tag{3.6}$$

$$u_z = m(-c_3\dot{z} + c_3\dot{z}_d + \ddot{z}_d + g + k_3\frac{\dot{z}}{m} - \tau_3 s_3 - \xi_3 sign(s_3) - d_z) \tag{3.7}$$

with $\tau_i > 0$ and $\xi_i > 0$ being the design parameters. By the above control inputs, the derivative of s_i will satisfy that $\dot{s}_i = -\tau_i s_i - \xi_i sign(s_i)$, i = 1,2,3, which implies the convergence of the tracking errors.

Since there are unknown disturbances in the designed control inputs, we introduce RBFNN to approximate the disturbances. Rewrite the expressions (3.5) - (3.7) as

$$u_x = m(Q_1 + \varepsilon_1 - \tau_1 s_1 - \xi_1 sign(s_1)) \tag{3.8}$$

$$u_y = m(Q_2 + \varepsilon_2 - \tau_2 s_2 - \xi_2 sign(s_2)) \tag{3.9}$$

$$u_z = m(Q_3 + \varepsilon_3 - \tau_3 s_3 - \xi_3 sign(s_3)) \tag{3.10}$$

where Q_i is obtained from a RBFNN, and ε_i represents the approximation error which is bounded, i = 1,2,3. The structure of a 3 layers RBFNN is shown in Figure 3.

The three layers of the RBFNN are input layer, hidden layer and output layer [29]. The mappings from the input layer to the hidden layer are nonlinear, while the ones from the hidden layer to the output layer are linear. The inputs of the RBFNN are the desired variable D and its derivatives $\dot{D}$ and $\ddot{D}$, and the error between actual variable and the desired one e, as well as its derivative $\dot{e}$.

Generally, Gaussian functions are selected to form the nonlinear mappings of a RBFNN, then the output of the i^{th} node of the hidden layer can be expressed as

$$\Phi_i = exp(-\frac{\|\eta - o_i\|^2}{\sigma_i^2}) \tag{3.11}$$

where Φ_i is the output of the hidden layer, $\eta = \left[e, \dot{e}, D, \dot{D}, \ddot{D}\right]$ is the input vector, o_i is the center of the i^{th} Gaussian function, and σ_i is the corresponding width, i = 1, ⋯, 5. Since the output Q is a liner function of Φ_i, it can be written as

$$Q = \sum_{i=1}^{5} W_i \Phi_i \tag{3.12}$$

where W_i is the weight of the RBFNN, i = 1, ⋯, 5.

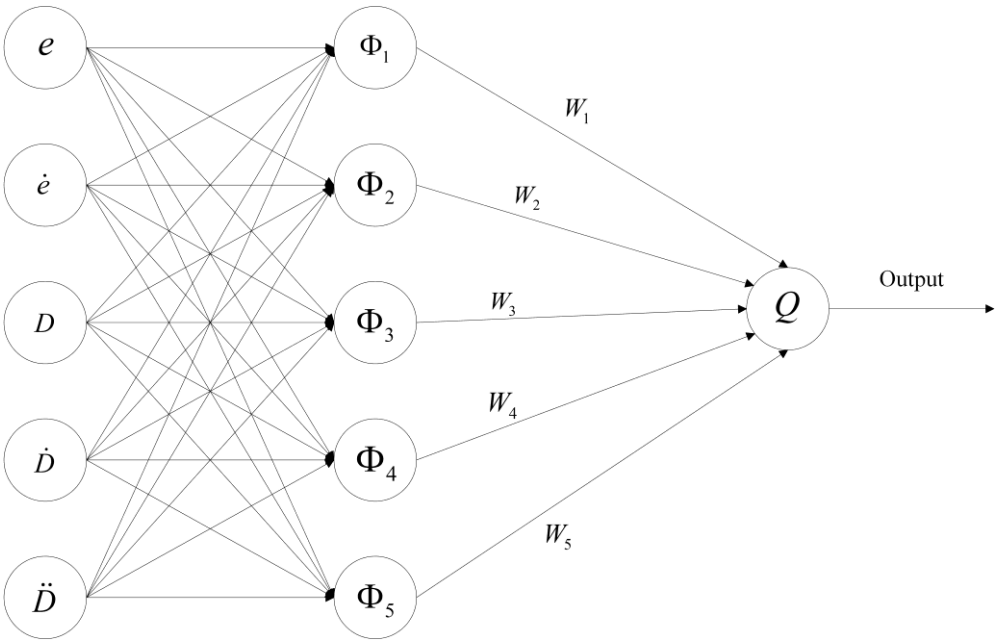

Figure 3. The Structure of a RBFNN.

Because the weights of the RBFNNs are unknown for the approximation task, we should estimate their values online. By denoting $\hat{W}_i$ as the estimation of W_i, i = 1, ⋯, 5, the estimated output of the RBFNN $\hat{Q}$ can be obtained as

$$\hat{Q} = \sum_{i=1}^{5} \hat{W}_i \Phi_i \tag{3.13}$$

Employing $\hat{Q}_i$, and considering the robustness of SMC, the position controllers are designed as

$$u_x = m(\hat{Q}_1 - \tau_1 s_1 - \xi_1 sign(s_1)) \tag{3.14}$$

$$u_y = m(\hat{Q}_2 - \tau_2 s_2 - \xi_2 sign(s_2)) \tag{3.15}$$

$$u_z = m(\hat{Q}_3 - \tau_3 s_3 - \xi_3 sign(s_3)) \tag{3.16}$$

The proof of the controllers' stability, and the adaptive laws of the weights $\hat{W}_i$, i = 1, ⋯, 5, will be presented later. Based on the above controllers, the lifting force u_1 and the desired roll and pitch angles are obtained as

$$u_1 = \frac{u_z}{\cos\phi \cos\theta} \tag{3.17}$$

$$\theta_d = \arctan\left(\frac{u_x \cos\psi_d + u_y \sin\psi_d}{u_z}\right) \tag{3.18}$$

$$\phi_d = \arctan\left(\frac{\cos\theta_d (u_x \sin\psi_d - u_y \cos\psi_d)}{u_z}\right) \tag{3.19}$$

where ϕ_d, θ_d, ψ_d are the desired roll, pitch and yaw angles, with ψ_d being given by the external order.

3.2. Attitude control

The purpose of attitude control is to guarantee the attitude angles converge to the desired ones. The convergence of ϕ and θ guarantees the tracking errors e_x, e_y move to zero quickly.

Similarly, the sliding surfaces are defined as (3.20) – (3.22)

$$s_4 = c_4(\phi - \phi_d) + (\dot{\phi} - \dot{\phi}_d) = c_4 e_\phi + \dot{e}_\phi \tag{3.20}$$

$$s_5 = c_5(\theta - \theta_d) + (\dot{\theta} - \dot{\theta}_d) = c_5 e_\theta + \dot{e}_\theta \tag{3.21}$$

$$s_6 = c_6(\psi - \psi_d) + (\dot{\psi} - \dot{\psi}_d) = c_6 e_\psi + \dot{e}_\psi \tag{3.22}$$

where $c_i > 0$, $i = 4,5,6$, are design coefficients, $e_\phi = \phi - \phi_d$, $e_\theta = \theta - \theta_d$, $e_\psi = \psi - \psi_d$ are the tracking errors between the actual and the desired angles.

The attitude control laws of the quadrotor are designed as

$$u_2 = \frac{I_x}{l}(-c_4 e_\phi + \ddot{\phi}_d - \frac{I_y - I_z}{I_x} qr + k_4 \frac{pl}{I_x} - \tau_4 s_4 - \xi_4 sign(s_4) - \frac{J_r}{I_x} q\omega_r - d_\phi) \tag{3.23}$$

$$u_3 = \frac{I_y}{l}(-c_5 e_\theta + \ddot{\theta}_d - \frac{I_z - I_x}{I_y} pr + k_5 \frac{ql}{I_y} - \tau_5 s_5 - \xi_5 sign(s_5) + \frac{J_r}{I_y} p\omega_r - d_\theta) \tag{3.24}$$

$$u_4 = \frac{I_z}{C}(-c_6 e_\psi + \ddot{\psi}_d - \frac{I_x - I_y}{I_z} pq + k_6 \frac{\psi}{I_z} - \tau_6 s_6 - \xi_6 sign(s_6) - d_\psi) \tag{3.25}$$

with $\tau_i > 0$, $\xi_i > 0$ being the design parameters, $i = 4, 5, 6$. Based on the above control laws, the derivative of s_i satisfies that $\dot{s}_i = -\tau_i s_i - \xi_i sign(s_i)$, $i = 4, 5, 6$, which implies the convergence of the tracking errors.

By employing RBFNNs to approximate the unknown disturbances in (3.23) - (3.25), and using their estimated output to construct the control laws, we can obtain the attitude control laws as

$$u_j = m(\hat{Q}_i - \tau_i s_i - \xi_i sign(s_i)) \tag{3.26}$$

with $j = 2, 3, 4$, $i = 4, 5, 6$.

In order to realize the control design, the weights of each RBFNN need to be updated. Therefore, the following adaptive law is designed.

$$\dot{\hat{W}} = -s_i \Phi \tag{3.27}$$

with $W = [W_1 \quad W_2 \quad W_3 \quad W_4 \quad W_5]^T$ and $\Phi = [\Phi_1 \quad \Phi_2 \quad \Phi_3 \quad \Phi_4 \quad \Phi_5]^T$ being the weight and the Gaussian function vectors of the used RBFNN.

Assumption 3.1. *The approximation error* ε_i *is bounded and satisfies that* $|\varepsilon_i| \le \xi_i$, $i = 1, \cdots, 6$.

Theorem 3.1. *With the auxiliary inputs* (3.14) – (3.16)*, and the control laws* (3.17), (3.26)*, and the adaptive law* (3.27) *for updating the weight vector of the used RBFNN, all states of the quadrotor system in* (2.6) *are guaranteed to converge to their desired signals, and the trajectory of the*

quadrotor can track the given reference.

Proof. Let $\tilde{Q}_i = \hat{Q}_i - Q_i$ (i =1, ⋯ ,6) be the estimate error of the RBFNN's output, and $\tilde{W} = \hat{W} - W$ be the corresponding estimate error of the weight vector. Choose a Lyapunov candidate function [30] as

$$V_i = \frac{1}{2} s_i^2 + \tilde{W}^T \tilde{W}, (i = 1, \cdots, 6) \tag{3.28}$$

Take the time derivative of V_i, one can get

$$\begin{aligned} \dot{V}_i &= s_i \dot{s}_i + \tilde{W}^T \dot{\hat{W}} \\ &= s_i(-Q_i + \hat{Q}_i - \varepsilon_i - \tau_i s - \xi_i sign(s_i)) + \tilde{W}^T \dot{\hat{W}} \\ &= \tilde{W}^T(\Phi s_i + \dot{\hat{W}}) - \varepsilon_i s_i - \tau_i s_i^2 - \xi_i | s_i | \end{aligned} \tag{3.29}$$

Taking the adaptive law (3.27) into (3.29), one can get that

$$\dot{V} = -\varepsilon_i s_i - \tau_i s_i^2 - \xi_i | s_i | \tag{3.30}$$

According to Assumption 1, $|\varepsilon_i| \le \xi_i$, so $\dot{V} \le 0$ can be guaranteed, which implies that $s_i \to 0$, i = 1, ⋯ ,6. By the definition of the sliding surfaces, and the characteristics of SMC, the convergence of e_x, e_y, e_z, e_ϕ, e_θ and e_ψ can be guaranteed, which means that all states of the quadrotor system converge to their desired signals, and the trajectory of the quadrotor tracks the given reference.

4. Optimization of the sliding mode controller

Based on the above control design, the stability of the quadrotor system can be guaranteed. However, the influence of the center o_i and the width σ_i of the Gaussian function Φ_i, i =1, ⋯ ,5, cannot be ignored. In this section, particle swarm optimization algorithm (PSO) will be used to adjust these parameters offline for better control performance.

PSO originates from the imitating of bird predation and has the advantages of simple implementation and fast convergence. Each bird or particle will carry the training parameters, and move according to the situations of itself and the whole group.

An evaluation function is needed for PSO, which is used to evaluate the position of the current particle. Here we choose the sum of absolute errors as the evaluation function, which is shown in (4.1).

$$f_e = \sum_{i=1}^{n} |\mu(i)| \tag{4.1}$$

where f_e denotes the evaluation function, and $\mu(i)$ stands for the error between a sample value and the output of the corresponding RBFNN, $i = 1, \cdots, n$, n is the number of training samples for PSO. The training samples are the parts of control inputs which are substituted by RBFNNs in (3.5) - (3.7) and (3.23) – (3.25) without disturbances.

Then, define the number of iterations G and the size N of the particle swarm. Each particle in the PSO contains the following information: the current position X, which represents the parameters of RBFNN that are needed to be optimized; the moving velocity V, which represents the updating rate of a particle position; as well as the particle dimension H, which is determined by the total number of the optimized parameters.

We also define the optimal value of the group as ***gb***, and the optimal value of an individual particle as ***pb***. At first, we initialize each ***pb*** by the value of the evaluation function at the initial position of the corresponding particle, and assign the smallest ***pb*** to ***gb***, then update ***pb*** and ***gb*** by the following rules: for each particle, if the value of the evaluation function (4.1) is smaller than the value in last iteration, we will assign ***pb*** with the value at the current position, otherwise ***pb*** will be unchanged. And ***gb*** selects the smallest ***pb*** among all the particles.

During each iteration, the position value of each particle is brought into the RBFNN to approximate the training samples, then the evaluation function can be calculated, by which ***pb*** and ***gb*** can be updated according to the rules mentioned above.

The updating values of the velocity and the position of a particle are calculated as

$$v_i(k+1) = v_i(k) + C_1 \cdot R_1 \cdot (pb_i(k) - X_i(k)) + C_2 \cdot R_2 \cdot (gb_i(k) - X_i(k)) \tag{4.2}$$

$$X_i(k+1) = X_i(k) + v_i(k+1) \tag{4.3}$$

where i=1, $\cdots$, N, C_1 , C_2 are the learning factors which are generally set to be 2, and R_1 , R_2 are random numbers belong to [0,1].

In order to improve the global optimization ability of the PSO, an inertia factor ρ and a scale factor λ are employed for the velocity and position updating of a particle.

$$v_i(k+1) = \rho \cdot v_i(k) + C_1 \cdot R_1 \cdot (pb_i(k) - X_i(k)) + C_2 \cdot R_2 \cdot (gb_i(k) - X_i(k)) \tag{4.4}$$

$$X_i(k+1) = X_i(k) + \lambda \cdot v_i(k+1) \tag{4.5}$$

Remark: In order to maintain the integrity of the employed RBFNN, the weights of each RBFNN are also placed in X for optimizing, but only the center value o and the width value σ are taken as the optimization result, since the weight W is estimated by the adaptive law (3.27).

The parameters of our PSO are listed in Table 1. And the process of PSO is shown in Figure 4.

Table 1. Parameters of the PSO.

Variable	Quantity	Value/Range
G	The number of iterations	250
N	Population size	50
H	Particle dimension	15
o	Center value of RBF neural network	[-3,3]
σ	Width value of RBF neural network	0.3
ρ	Inertial factor	0.3
C_1,C_2	learning factors	2
λ	Scale factor	0.7

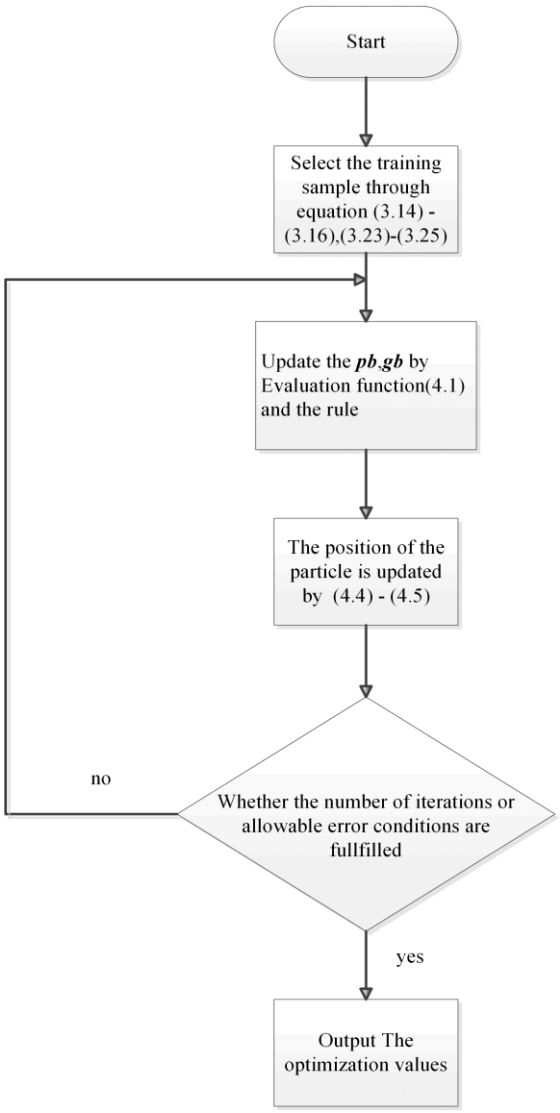

Figure 4. The process of PSO algorithm.

5. Simulation result

Some simulation experiments are conducted to verify the effectiveness of the proposed control strategy, and the results of the experiments are demonstrated in this section. The parameters of the quadrotor model are given in Table 2. And the parameters of the designed sliding mode controllers are listed in Table 3. The disturbances for the simulation experiments are considered as follows, whose curves are plotted in Figures 5–6.

$$d_x = d_y = d_z = 0.5 \cdot \sin(0.5t) \tag{5.1}$$

$$d_\phi = d_\theta = d_\psi = 0.25 \cdot \cos(\pi t) \tag{5.2}$$

In order to exposit the superiority of the designed sliding mode controllers which are based on RBFNNs with PSO(SMC-RBFNN-PSO), some comparisons are made between SMC-RBFNN-PSO and sliding mode controllers based on RBFNNs without optimization (SMC-RBFNN). The center and the width values of SMC-RBFNN are determined by the experience of the designer. The comparison results are shown in Figures 7–10.

The trajectories of the controlled quadrotor with the two methods, together with the desired one are shown in Figure 7. The initial values of the position and attitude of the quadrotor are given in Table 4.

Table 2. Parameters of the quadrotor model.

Variables	Value	Unit
m	2	kg
l	0.21	m
I_x	1.25	Ns^2/rad
I_y	1.25	Ns^2/rad
I_z	2.5	Ns^2/rad
k_1,k_2,k_3	0.1	Ns/m
k_4,k_5,k_6	0.12	Ns/m
b	0.5	Ns^2
k	2	N/ms^2
C	1	
J_r	0.2	Ns^2/rad

Table 3. Control Parameters.

Variables	Value
c_1,c_2,c_3,c_4,c_5,c_6	10
τ_1,τ_2	15
τ_3	55
ξ_1,ξ_2,ξ_3	0.1
τ_4,τ_5,τ_6	10
ξ_4,ξ_5,ξ_6	0.1

Table 4. The reference position and attitude.

Variable	Quantity (Unit)	Value
X_d	The desire trajectory of the X axis(m)	2*sin (0.1*t)
Y_d	The desire trajectory of the Y axis(m)	2*cos (0.1*t)
Z_d	The desire trajectory of the Z axis(m)	0.1*t
[X_0, Y_0, Z_0]	Initial positions(m)	[0,2,0]
ψ_d	The desire value of the yaw Angle(rad)	0
[ϕ_0, θ_0, Ψ_0]	Initial angles(rad)	[0,0,0]

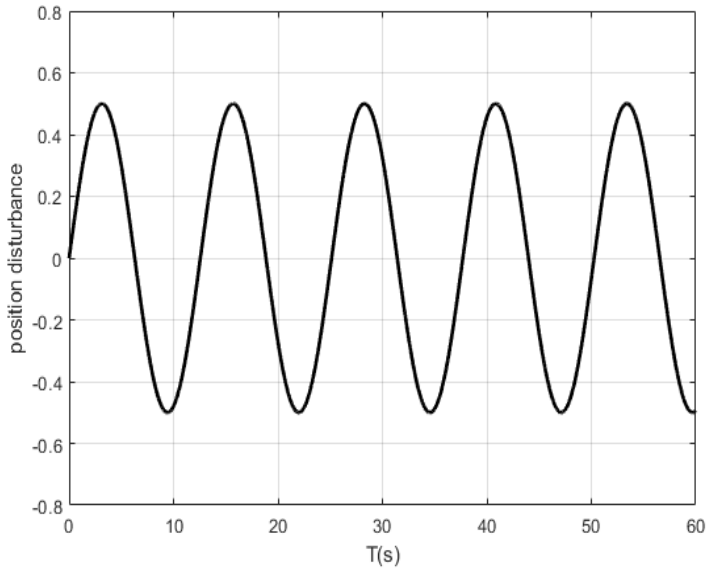

Figure 5. The curve of disturbances d_x, d_y, d_z.

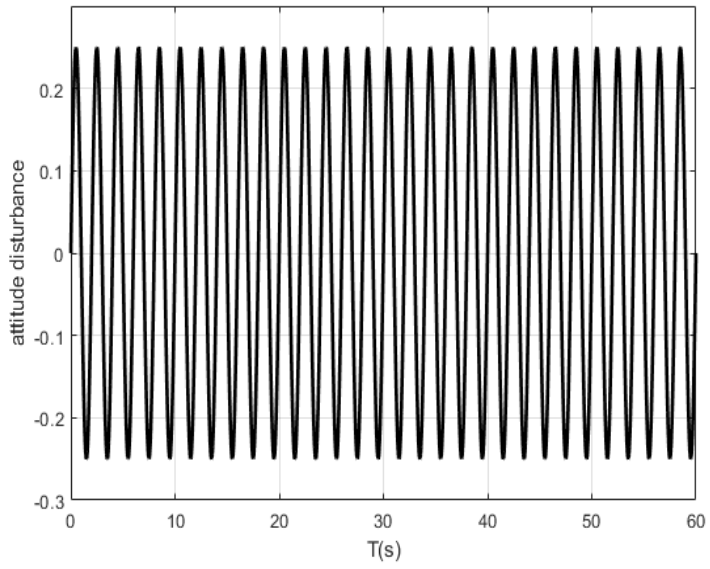

Figure 6. The curve of disturbances d_θ, d_ϕ, d_ψ.

In order to show more details of the flight trajectories, the position and angle variables of the quadrotor are shown in Figures 8–9. It can be seen that both of the control methods can drive the quadrotor to track the desired trajectory, but SMC-RBFNN-PSO can achieve quicker convergence of all the states. This fact indicates that the center and the width values of the neural networks are effectively optimized by the PSO. The desired roll angle and pitch angle are calculated by (3.18)–(3.19), which are related to the auxiliary inputs u_x, u_y, so the angles ϕ and θ with the two methods are different, which can be seen in Figure 9.

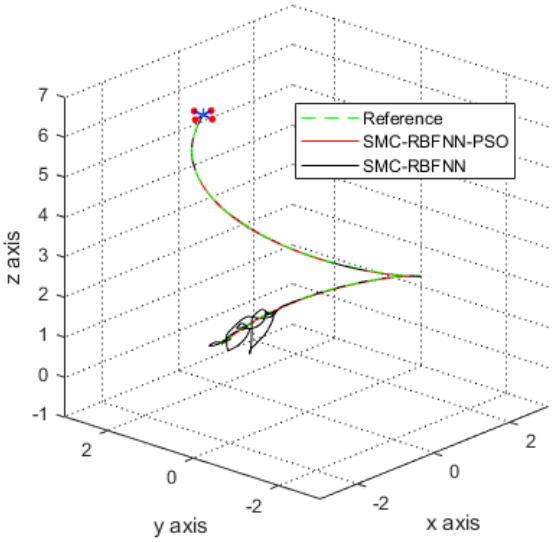

Figure 7. The flight trajectories of the quadrotor.

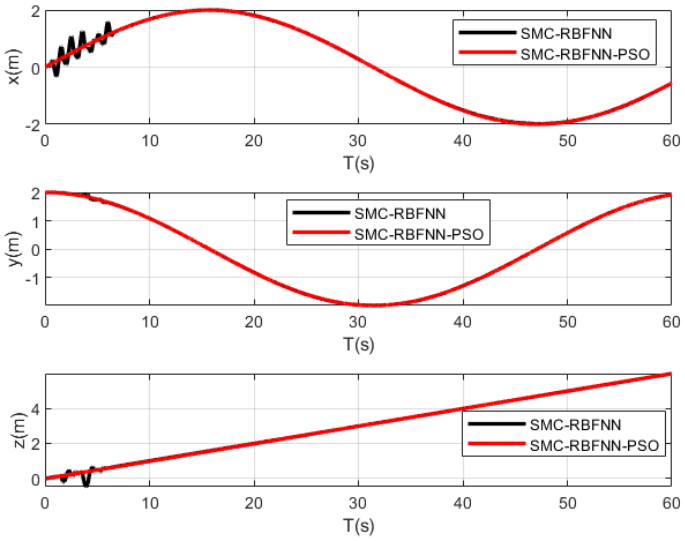

Figure 8. The position curves of the quadrotor.

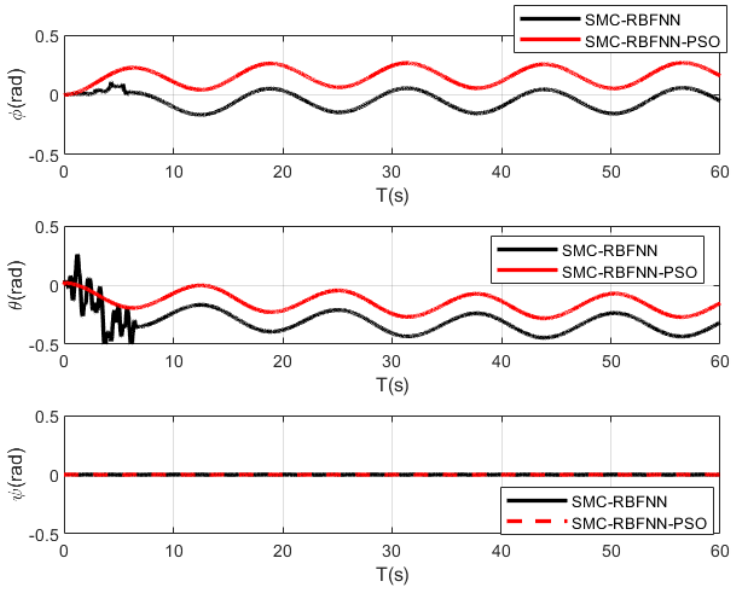

Figure 9. The attitude curves of the quadrotor.

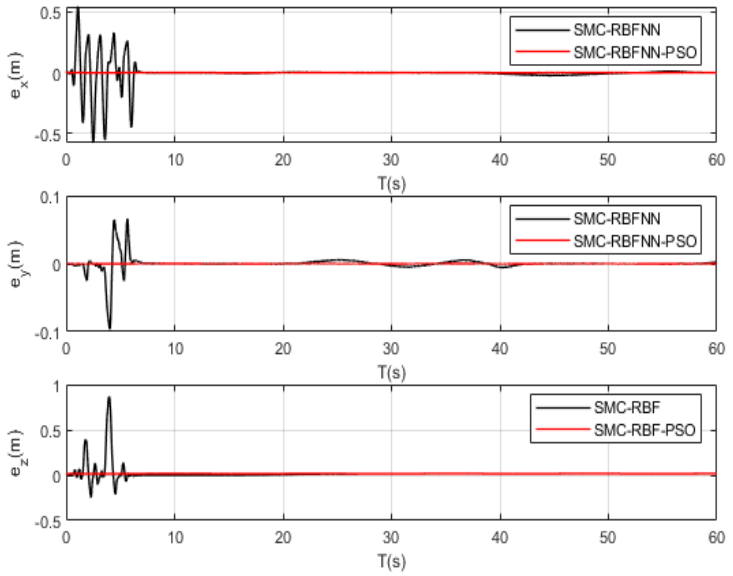

Figure 10. The errors of the position variables.

Figure 10 shows the tracking errors of the three position variables with the two methods. By SMC-RBFNN, the maximum absolute errors of x, y and z are 0.5775m, 0.09m and 0.8676m, respectively. And the errors converge to zero at 6.88s, 7.24s and 6.85s, respectively. It can be seen from Figure 10 that there are unexpected fluctuations in the position tracking of the quadrotor with SMC-RBFNN. However, by SMC-RBFNN-PSO, the quadrotor can track the desired trajectory more quickly and smoothly without fluctuation.

6. Conclusions

A RBFNN based SMC strategy with PSO is proposed in this paper for the tracking control of quadrotors. Based on the SMC, the position and the attitude controllers can be designed at first. However, these controllers cannot be exerted directly because of the unknown disturbances. Therefore, the RBFNNs are employed to approximate some items of the control laws. The weights of the RBFNNs are updated by adaptive mechanism. Besides, the center and the width values of the RBFNNs are optimized by PSO, with which better trajectory tracking performance can be achieved for the quadrotor. Simulation results demonstrate the stability of the system and the convergence of the states to their desired signals. Also, the comparison results show obvious superiority of the proposed SMC-RBFNN-PSO to SMC-RBFNN.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61603144, in part by the Natural Science Foundation of Fujian Province under Grant 2018J01095, in part by the Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University under Grant ZQN-PY509, in part by the Science and Technology Major Project Oriented to Colleges in Fujian Province for Industry Study Cooperation (2013H6016).

Conflict of interest

The authors declare there is no conflict of interest

References

1. N. Fethalla, M. Saad, H. Michalska, J. Ghommam, Robust observer-based dynamic sliding mode controller for a quadrotor UAV, *IEEE Access*, **6** (2018), 45846–45859.
2. A. Levant, Principles of 2-sliding mode design, *Automatica*, **43** (2007), 576–586.
3. R. Xu, Ü. Özgüner, Sliding mode control of a class of underactuated systems, *Automatica*, **44** (2008), 233–241.
4. D. Lee, H. J. Kim and S. Sastry, Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter, *Int. J. Control Autom. Syst*., **7**(2009), 419–428.
5. E. H. Zheng, J. J. Xiong, J. L. Luo, Second order sliding mode control for a quadrotor UAV, *ISA Trans.*, **53** (2014), 1350–1356.
6. J. J. Xiong, E. H. Zheng, Position and attitude tracking control for a quadrotor UAV, *ISA Trans.*, **53** (2014), 725–731.
7. J. J. Xiong, G. B. Zhang, Global fast dynamic terminal sliding mode control for a quadrotor UAV, *ISA Trans.*, **66** (2017), 233–240.
8. H. Ríos, R. Falcón, O. A. González, A. Dzul, Continuous sliding-mode control strategies for quadrotor robust tracking: real-time application, *IEEE Trans. Ind. Electron.*, **66** (2019), 1264–1272.

9. B. X. Mu, K. W. Zhang and Y. Shi, Integral sliding mode flight controller design for a quadrotor and the application in a heterogeneous multi agent system, *IEEE Trans. Ind. Electron.*, **64** (2017), 9389–9398.
10. H. P. Wang, X. F. Ye, Y. Tian, G. Zheng, N. Christov, Model-free based terminal SMC of quadrotor attitude and position, *IEEE Trans. Aerosp. Electron. Syst.*, **52** (2016), 2519–2528.
11. O. Mofid, S. Mobayen, Adaptive sliding mode control for finite time stability of quadrotor UAVs with parametric uncertainties, *ISA Trans.*, **72** (2018), 1–14.
12. L. Derafa, A. Benallegue, L. Fridman, Super twisting control algorithm for the attitude tracking of a four rotors UAV, *J. Franklin Inst.*, **349** (2012), 685–699.
13. M. Boukattaya, N. Mezghani, T. Damak, Adaptive nonsingular fast terminal sliding-mode control for the tracking problem of uncertain dynamical systems, *ISA Trans.*, **77** (2018), 1–19.
14. M. Labbadi, M. Cherkaoui, Robust adaptive nonsingular fast terminal sliding-mode tracking control for an uncertain quadrotor UAV subjected to disturbances, *ISA Trans.*, **99** (2020), 290–304.
15. B. Wang, X. Yu, L. X. Mu, Y. M. Zhang, Disturbance observer-based adaptive fault-tolerant control for a quadrotor helicopter subject to parametric uncertainties and external disturbances, *Mech. Syst. Signal Proc.*, **120** (2019), 727–743.
16. G. Q. Zhu, S. Wang, L. F. Sun, W. C. Ge, X. Y. Zhang, Output feedback adaptive dynamic surface sliding-mode control for quadrotor UAVs with tracking error constraints, *Complexity*, **2020** (2020), 1–23.
17. M. Labbadi, M. Cherkaoui, Robust adaptive backstepping fast terminal sliding mode controller for uncertain quadrotor UAV, *Aerosp. Sci. Technol.*, **93** (2019), 105306.
18. H. Razmi, S. Afshinfar, Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV, *Aerosp. Sci. Technol.*, **91** (2019), 12–27.
19. Y. M. Chen, Y. L. He, M. F. Zhou, Decentralized PID neural network control for a quadrotor helicopter subjected to wind disturbance, *J. Cent. South Univ.*, **22** (2015), 168–179.
20. S. S. Li, Y. N. Wang, J. H. Tan, Y. Zheng, Adaptive RBFNNs/integral sliding mode control for a quadrotor aircraft, *Neurocomputing.*, **216** (2016), 126–134.
21. O. Doukhi, D. J. Lee, Neural network-based robust adaptive certainty equivalent controller for quadrotor UAV with unknown disturbances, *Int. J. Control Autom. Syst.*, **17** (2019), 2365–2374.
22. Q. Z. Xu, Z. S. Wang, Z. Y. Zhen, Adaptive neural network finite time control for quadrotor UAV with unknown input saturation, *Nonlinear Dyn.*, **98** (2019), 1973–1998.
23. C. X. Mu, Y. Zhang, Learning-based robust tracking control of quadrotor with time-varying and coupling uncertainties, *IEEE Trans. Neural Netw. Learn. Syst.*, **31** (2020), 259–273.
24. Z. Li, X. Ma, Y. B. Li, Robust tracking control strategy for a quadrotor using RPD-SMC and RISE, *Neurocomputing.*, **331** (2019), 312–322.
25. Y. Wang, Y. Chenxie, J. Tan, C. Wang, Y. Wang and Y. Zhang, Fuzzy radial basis function neural network PID control system for a quadrotor UAV based on particle swarm optimization, *IEEE Int. Conf. Inf. Autom.*, (2015), 2580–2585.
26. R. Zhang, J. Tao, R. Lu, Q. Jin, Decoupled ARX and RBF neural network modeling using PCA and GA optimization for nonlinear distributed parameter systems, *IEEE Trans. Neural Netw. Learn. Syst.*, **29** (2018), 457–469.

27. S. Q. Wang, M. Roger, J. Sarrazin, C. Lelandais-Perrault, Hyperparameter optimization of two-hidden-layer neural networks for power amplifiers behavioral modeling using genetic algorithms, *IEEE Microw. Wirel. Compon. Lett.*, **29** (2019), 802–805.
28. W. J. Cai, J. H. She, M. Wu, Y. Ohyama, Disturbance suppression for quadrotor UAV using sliding-mode-observer-based equivalent-input-disturbance approach, *ISA Trans.*, **92** (2019), 286–297.
29. W. K. Alqaisi, B. Brahmi, J. Ghommam, M. Saad, V. Nerguizian, Adaptive sliding mode control based on RBF neural network approximation for quadrotor, *IEEE Int. Symp. Robot. Sensors Environ.*, (2019), 1–7.
30. J. K. Liu, X. H. Wang, *Advanced sliding mode control for mechanical systems: Design, analysis and MATLAB simulation*, Tsinghua University Press, Beijing, 2011.

AIMS Press