Neural Network Based PID Control for Quadrotor Aircraft

Dandan Zhao[1,2], Changyin Sun[1,2](✉), Qingling Wang[1,2], and Wankou Yang[1,2,3]

[1] School of Automation, Southeast University, Nanjing 210096, China
cysun@seu.edu.cn

[2] Key Lab of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing 210096, China

[3] Key Laboratory of Child Development and Learning Science of Ministry of Education, Southeast University, Nanjing 210096, China

Abstract. A back propagation neural network (BPNN) based PID control strategy for the attitude of quadrotor is proposed in this paper. Firstly, the architecture and dynamic model of quadrotor are analyzed according to the Newton-Euler Equation. Secondly, a nonlinear attitude model is established on the basis of the mathematical analysis. Thirdly, by eliminating the inverse error adaptively, a BPNN based PID controller is introduced to improve the robustness. Furthermore, PID parameters are adaptively adjusted through the training of neural network weighted coefficients. Finally, numerical examples demonstrate the performance of the designed BPNN based PID controller in terms of precision, adaptability and robustness.

Keywords: Neural network · PID controller · Attitude control · Quadrotor

1 Introduction

Quadrotor has great practical value and wide development prospects as a kind of vertical take-off and landing (VTOL) aircraft with excellent performance. Rotorcraft has interested the robotics and control community in recent years. Quadrotor, as the representative of the rotorcraft, has always been a hotspot in research of rotorcraft. Featured with multi-variable, nonlinearity, strong coupling and sensitivity of interference, quadrotor achieves six degrees of freedom (position and attitude) of aircraft movement by adjusting the four motor speeds [1].

The autonomous flight control circuit can generally be divided into three parts, namely position control, speed control and attitude control (Fig. 1). Attitude stability control (internal loop control) is the basis of micro flying platform for autonomous flight, the control result of which is of great importance to the aircraft flight characteristics. Thus the performance of attitude control is the key to the overall flight control. In these days, many researchers have being conducted on this field. The typical methods include PID control [1], adaptive control [2, 3], the sliding mode variable structure control [4, 5], back-stepping control [6, 7], robust control [8] and neural network control [9].

X. He et al. (Eds.): IScIDE 2015, Part II, LNCS 9243, pp. 287–297, 2015.
DOI: 10.1007/978-3-319-23862-3_28

The BPNN based PID control proposed in this article combines the advantages of PID control and neural network control. With the identification results of the BPNN, the selection and setting of proportion, integral and differential parameters are conducted.

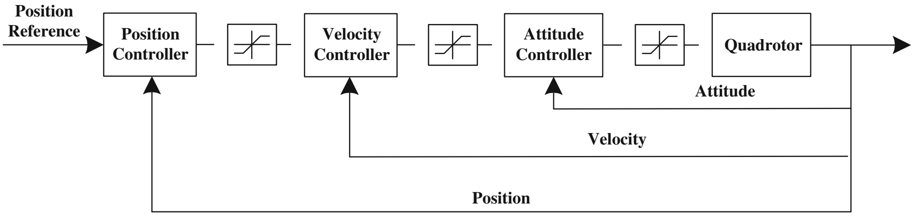

Fig. 1. The control structure of quadrotor

2 Dynamic Modeling for a Quadrotor

A quadrotor conventional structure is composed of four perpendicular arms which have motor and rotor, respectively (Fig. 2). The four rotors provide upwards propulsion as well as the direction control. In order to balance the effect of torque, the micro quadrotor system is divided into two opposite rotor pairs (rotor 1, 3 and rotor 2, 4) where one pair rotates clockwise while the other pair counters clockwise (Fig. 3) [1]. Vertical ascending (descending) flight is obtained by increasing (decreasing) thrust forces. By applying speed difference between front and rear rotors, we will have pitching motion which is defined as a rotation motion around Y axis. Using the same analogy in rolling motion, we will have a rotation motion around X axis. Yaw rotation, which is defined as a rotation motion around Z axis, is obtained through increasing (decreasing) speed of rotors (1.3) while decreasing (increasing) speed of rotors (2.4). Unlike pitch and roll motions, yaw rotation is the result of reactive torques produced by rotors rotation.

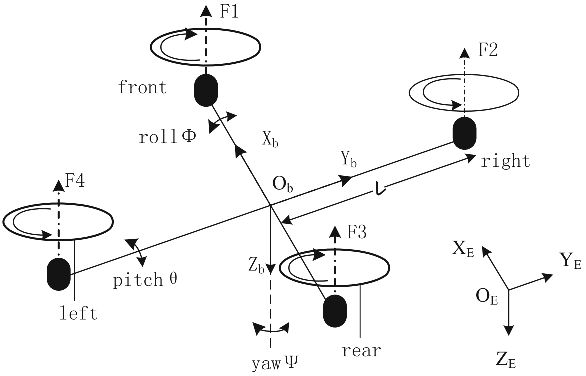

Fig. 2. Inertia and body coordinate system of the quadrotor

2.1 System Dynamics

The derivation of the dynamics is based on inertial and body frames. The quadrotor has six degrees of freedom, Cartesian (x, y, z) and angular (ϕ, θ, ψ). Using the formalism of Newton-Euler and assuming that the quadrotor has a symmetrical structure, the thrust and the drag forces are proportional to the square of the rotors speed, the dynamic model of quadrotor can be expressed as [10]:

$$\begin{cases} I_{xx}\dot{p} = qr\left(I_{yy} - I_{zz}\right) + J_r q \Omega_r + lU_1 \\ I_{yy}\dot{q} = pr(I_{zz} - I_{xx}) - J_r p \Omega_r + lU_2 \\ I_{zz}\dot{r} = pq\left(I_{xx} - I_{yy}\right) + J_r \dot{\mathit{\Omega}}_r + U_3 \\ m\ddot{x} = (\sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi)U_4 - K_1\dot{x} \\ m\ddot{y} = (-\cos\psi \sin\phi + \sin\psi \sin\theta \cos\phi)U_4 - K_1\dot{y} \\ m\ddot{z} = \cos\psi \cos\phi U_4 + K_2\dot{z} + mg \end{cases} \tag{1}$$

Where:

- p, q, r is the angular velocity around the x, y, z axis in body coordinate respectively
- $I_{xx,yy,zz}$ is the inertia of the x, y, z axis respectively
- J_r is the inertia of the quadrotor
- l is the lever length
- $\Omega_r = \Omega_2 + \Omega_4 - \Omega_1 - \Omega_3$, $\Omega_{1,2,3,4}$ is angular velocity of rotor 1, 2, 3, 4, respectively
- U_1, U_2, U_3, U_4 are the defined control signals instead of use rotor speeds as control inputs, b is the thrust coefficient, d is the drag coefficient and $F_{1,2,3,4}$ is the rotor lift, $F_i = b\Omega_i^2$

$$\begin{cases} U_1 = b(\Omega_4^2 - \Omega_2^2) \\ U_2 = b(\Omega_1^2 - \Omega_3^2) \\ U_3 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ U_4 = \sum_{i=1}^{4} F_i = -b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{cases} \tag{2}$$

From the above analysis, we can safely draw the conclusion that the attitude control of quadrotor is independent and the displacement control is affected by the attitude control.

2.2 Attitude Modeling

Due to the symmetrical structure of quadrotor, the design of the roll controller and the pitch controller can be exactly the same, so this paper just considers the specific design method of roll control.

If quadrotor is close to hover state, then the dynamic can be linearized as the gyroscopic effect and air resistance can be ignored [11]. Meanwhile, the attitude angular is very small in which case $(p, q, r) \approx (\dot{\phi}, \dot{\theta}, \dot{\psi})$. The linearized attitude model is:

$$\begin{cases} I_{xx}\ddot{\phi} = lU_1 \\ I_{yy}\ddot{\theta} = lU_2 \\ I_{zz}\ddot{\psi} = U_3 \end{cases} \tag{3}$$

It can be seen from the linear attitude model that there is no coupling between the attitude channels, the controllers can be designed separately.

Extract the attitude equation from Eq. (3):

$$I\ddot{\varphi} = \tau \tag{4}$$

τ is proportional to the difference of the square of the rotor angular velocity, I is the inertia of the quadrotor and φ is the attitude angular. Assuming the transform function of command signal to the motor and the actual speed difference output is first-order inertia link [12], the transfer function of motor instruction signal and the attitude angular can be obtained as

$$G_{\varphi u} = \frac{\varphi(s)}{u(s)} = \frac{k_{\varphi}}{s^2(\tau s + 1)} \tag{5}$$

The unknown parameters can be identified through manual flight inputs and outputs data fitting. The roll channel transfer function is identified as:

$$G_{\phi u} = \frac{53.46}{s^3 + 12.05s^2 + 101.8s} \tag{6}$$

And the yaw channel transfer function is identified as:

$$G_{\psi u} = \frac{1.6762}{0.1256s^3 + 0.7087s^2 + s} \tag{7}$$

3 BPNN Based PID Control Strategy

Attitude control is the core of the quadrotor control and its control performance will greatly affect the flight stability. BPNN based PID is introduced as attitude control method due to its effective transient and steady performance. The BPNN based PID controller can reduce the computation load and improve the system response speed [13]. According to the decoupled linear model, the roll channel, pitch channel and yaw channel BPNN based PID controllers can be designed independently.

The BPNN based PID controller (Fig. 3) is introduced to eliminate the inverse error adaptively and improve the robustness of the controller.

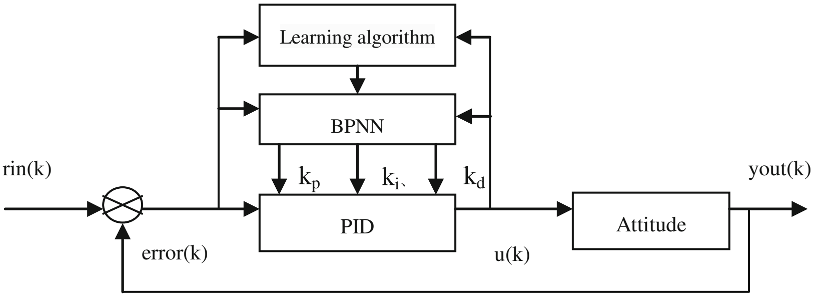

Fig. 3. BPNN based PID control structure

The typical PID control algorithm is as follow:

$$u(t) = k_p\left[e(t) + \frac{1}{T_I}\int_0^t e(t)dt + T_D\frac{de(t)}{dt}\right] \tag{8}$$

For the convenience of simulation, using the typical incremental digital PID control equation:

$$u(k) = u(k-1) + kp(e(k) - e(k-1)) + kie(k) + kd(e(k) - 2e(k-1) + e(k-2)) \tag{9}$$

kp, *ki*, *kd* stands for proportional, integral, differential coefficient respectively. Regard *kp*, *ki*, *kd* as dependent coefficients based on the adjustments of system operating status. Therefore Eq. (9) can be described as:

$$u(k) = f[u(k-1), kp, ki, kd, e(k), e(k-1), e(k-2)] \tag{10}$$

$f(.)$ is a nonlinear function of *kp*, *ki*, *kd*, u(k-1), e(k). Such a control law $u(k)$ can be obtained by BPNN training and learning as it has the ability to approximate any nonlinear function.

BPNN is one of the most widely used neural network model currently. BPNN is the multilayer feedforward network based on the training of error back propagation algorithm. BPNN can learn and store a lot of input - output model mapping without the knowledge of mathematical equations of the mapping relationship. The learning rule is that using the steepest descent method to minimize the error sum of squares of the network by back propagation, which can constantly adjust the network weights and threshold [14]. BPNN topological structures include input layer, hidden layer and output layer (Fig. 4).

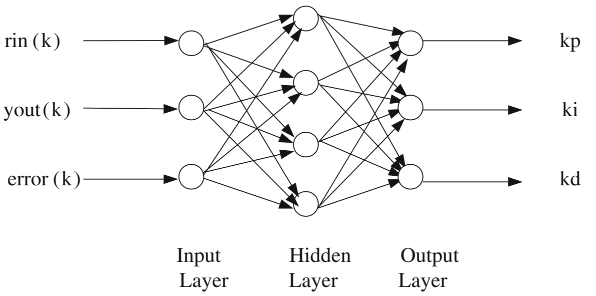

Fig. 4. BPNN topological structure

The inputs of the network input layer:

$$o_j^{(1)} = x(j), j = 1, 2\ldots m \tag{11}$$

The inputs and outputs of the network hidden layer:

$$net_i^{(2)}(k) = \sum_{j=1}^{m} w_{ij}^{(2)} o_j^{(1)}, j = 1, 2\ldots m, i = 1, 2\ldots q \tag{12}$$

$$o_i^{(2)}(k) = f(net_i^{(2)}(k)), i = 1, 2\ldots q \tag{13}$$

$\left\{ w_{ij}^{(2)} \right\}$ stand for the hidden layer weighted coefficient, superscript (1), (2), (3) represent the input layer, hidden layer and output layer, $f(x) = (e^x - e^{-x})/((e^x + e^{-x})$.

The inputs and outputs of the network output layer:

$$net_l^{(3)}(k) = \sum_{i=1}^{q} w_{li}^{(3)} o_i^{(2)}(k) \tag{14}$$

$$o_l^{(3)}(k) = g(net_l^{(3)}(k)), l = 1, 2, 3 \tag{15}$$

$$o_1^{(3)}(k) = kp, o_2^{(3)}(k) = ki, o_3^{(3)}(k) = kd \tag{16}$$

$\{w_{li}^{(3)}\}$ stand for the output layer weighted coefficient, the output layer neuron activation function is the nonnegative function Sigmoid $g(x) = e^x/(e^x + e^{-x})$.

Take a performance index function:

$$E(k) = \frac{1}{2}(rin(k) - yout(k))^2 \tag{17}$$

Modify the weights of the network according to the gradient descent method with an inertia item that speeds up the convergence search:

$$\Delta w_{li}^{(3)}(k) = -\eta \frac{\partial E(k)}{\partial w_{li}^{(3)}} + \gamma \Delta w_{li}^{(3)}(k-1) \tag{18}$$

Where η is the learning coefficient and γ is the Inertia coefficient.

$$\frac{\partial E(k)}{\partial w_{li}^{(3)}} = \frac{\partial E(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial u(k)} \cdot \frac{\partial u(k)}{\partial o_l^{(3)}(k)} \cdot \frac{\partial o_l^{(3)}(k)}{\partial net_l^{(3)}(k)} \cdot \frac{\partial net_l^{(3)}(k)}{\partial w_{li}^{(3)}} \tag{19}$$

Using $\operatorname{sgn}\left(\frac{y(k)-y(k-1)}{u(k)-u(k-1)}\right)$ to approximately representative $\frac{\partial y(k)}{\partial u(k)}$.
According to Eq. (9):

$$\begin{aligned} \frac{\partial u(k)}{\partial o_1^{(3)}(k)} &= e(k) - e(k-1) \\ \frac{\partial u(k)}{\partial o_2^{(3)}(k)} &= e(k) \\ \frac{\partial u(k)}{\partial o_3^{(3)}(k)} &= e(k) - 2e(k-1) + e(k-2) \end{aligned} \tag{20}$$

The output layer weight can be obtained:

$$\Delta w_{li}^{(3)}(k) = \eta \delta_l^{(3)} o_i^{(2)}(k) + \gamma \Delta w_{li}^{(3)}(k-1) \tag{21}$$

$$\delta_l^{(3)} = e(k) \frac{\partial \widehat{y}(k)}{\partial u(k)} \frac{\partial u(k)}{\partial o_l^{(3)}(k)} g'(net_l^{(3)}(k)), l = 1,2,3 \tag{22}$$

Similarly, the hidden layer weight can be obtained:

$$\Delta w_{ij}^{(2)}(k) = \eta \delta_i^{(2)} o_j^{(1)}(k) + \gamma \Delta w_{li}^{(2)}(k-1) \tag{23}$$

$$\delta_i^{(2)} = f'(net_i^{(2)}(k)) \sum_{l=1}^{3} \delta_l^{(3)} w_{li}^{(3)}(k), i = 1,2,\ldots q \tag{24}$$

The steps of BPNN based PID algorithm:
Step 1: Determine the structure of BP network, the input layer node number M, the number of hidden layer nodes Q; the weighted coefficient of each layer and the selected learning rate and inertia coefficient are initialized; k = 1 for this step
Step 2: Sampling to get $rin(k)$, $yout(k)$; Calculate $error(k) = rin(k) - yout(k)$
Step 3: Calculate the inputs and outputs of each layer, the outputs of output layer are the parameters *kp*, *ki*, *kd*

Step 4: Calculate u(k) of the PID controller
Step 5: Do neural network learning to adjust the weighted coefficient to realize the adaptive adjustment of PID control parameters; Back to Step 2.

4 Simulations

4.1 Attitude Model Validations

The compared results of the model outputs and the real flight outputs are presented in Fig. 5. The figures demonstrate the validation of the attitude model.

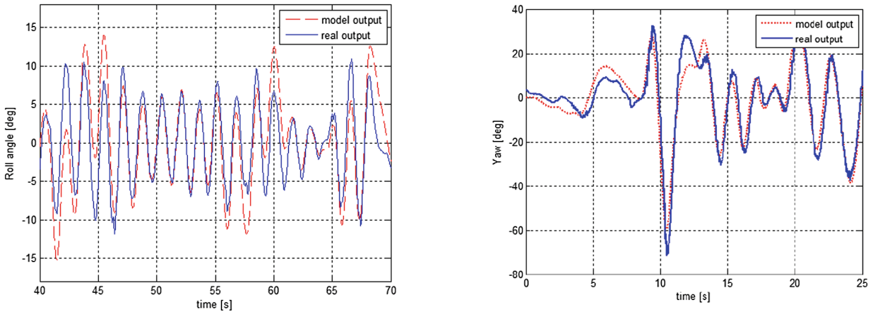

Fig. 5. Attitude model validations

4.2 Attitude Control Simulations

The neural network structure of the roll channel simulations consists of 3 input layer nodes, 4 hidden layer nodes and 3 output layer nodes. The initial values of the rate and inertia coefficient are 0.3 and 0.05 respectively.

The roll channel simulation results are in the following Figs. 6 and 7. Figure 6 shows that roll channel can follow the step signal with quite low overshot of 5 % and shorter rise time of 0.4 s in comparison of typical PID control of 1 s.

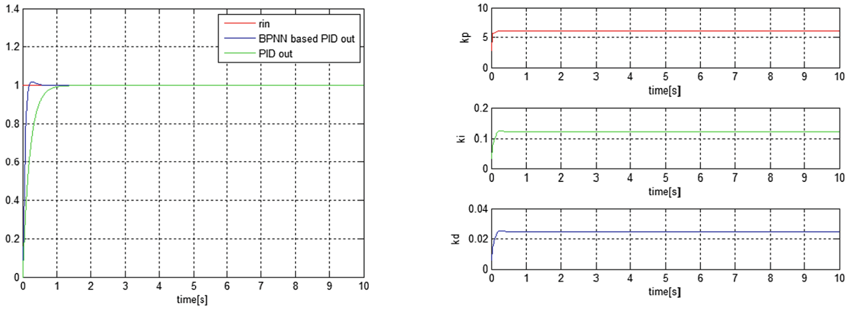

Fig. 6. The step response of roll channel

Furthermore, in order to test the capacity of resisting model uncertainty, the transfer function is slightly changed in the simulation of Fig. 7. The typical PID control can not follow the step signal with expected performance. Nevertheless, the BPNN based PID method can still follow the step signal with desired overshot and rising time due to the adaptability of the PID coefficient values. The theory presented in the paper is tested as having good robustness.

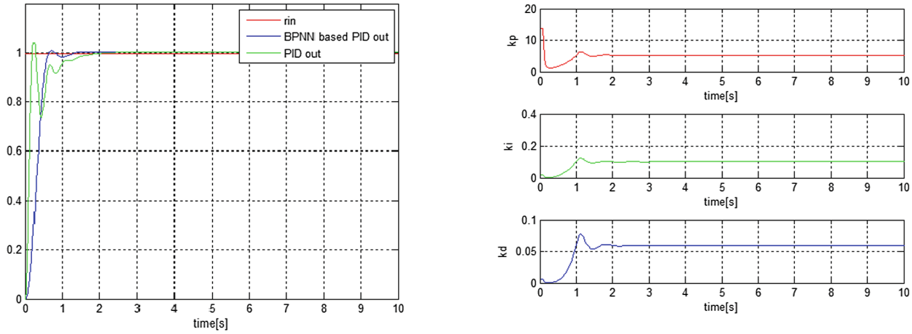

Fig. 7. The step response of roll channel with model uncertainty

The neural network structure of the yaw channel simulations consists of 3 input layer nodes, 4 hidden layer nodes and 3 output layer nodes. The initial values of the rate and inertia coefficient are 0.25 and 0.05 respectively.

Similarly, the yaw channel simulation results are in the following Figs. 8 and 9. Figure 8 shows that roll channel can follow the step signal with quite low overshot of 5 % and shorter rise time of 0.3 s in comparison of typical PID control of 0.9 s.

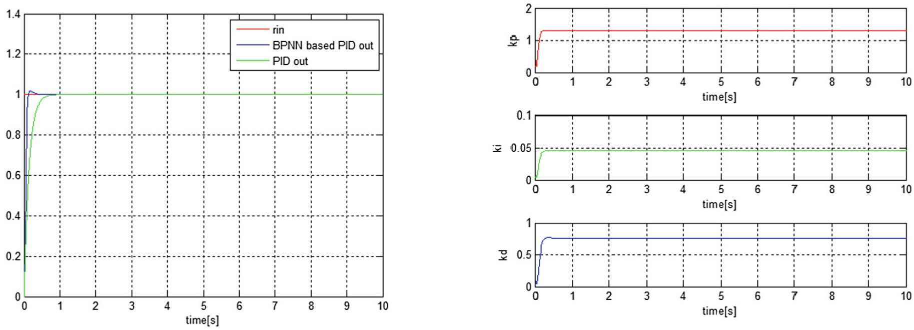

Fig. 8. The step response of yaw channel

The BPNN based PID method can still follow the step signal with desired overshot and rising time due to the adaptability of the PID coefficient values while the typical PID control performs not so well (Fig. 9).

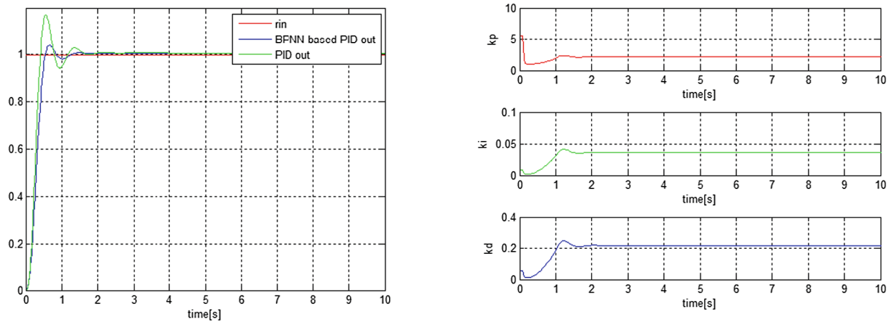

Fig. 9. The step response of yaw channel with model uncertainty

The simulation results demonstrate that attitude angular can follow the step signal with quite low overshot and shorter rise time in comparison of typical PID control. Besides, taking the modeling uncertainty into consideration, the transfer functions indentified before may have a certain error. The capacity of resisting model uncertainty through the BPNN based PID method is tested in the above simulations.

5 Conclusions

This paper proposed a BPNN based PID control of quadrotor attitude. Quadrotor dynamic model and attitude model were established. Furthermore, BPNN based PID controller was designed toachieve the adaptive adjustment of PID parameters by neural network back propagation. Considering the precision, adaptability and robustness of the control law, Matlab/Simulink based simulations were conducted. The performances of the control law were validated by simulation results.

Acknowledgments. The work was supported in part by Natural Science Foundation of Jiangsu Province of China under Grant No. BK20130471 and No. BK20140638, China Postdoctoral Science Foundation under grant No. 2013M540404, Jiangsu Planned Projects for Postdoctoral Research Funds under grant No. 1401037B, open fund of Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education under Grant No. MCCSE2013B01, the Open Project Program of Key Laboratory of Child Development and Learning Science of Ministry of Education, Southeast University (No. CDLS-2014-04), and A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), and the Fundamental Research Funds for the Central Universities.

References

1. Cavalcante Sa, R., De Araujo, A.L.C, Varela, A.T, et al.: Construction and PID control for stability of an unmanned aerial vehicle of the type quadrotor. In: 2013 Latin American, Robotics Symposium and Competition (LARS/LARC), pp. 95–99. IEEE (2013)

2. Lee, T.: Robust adaptive attitude tracking on with an application to a quadrotor UAV. IEEE Trans. Control Syst. Technol. **21**(5), 1924–1930 (2013)
3. Mohammadi, M., Shahri, A.M.: Adaptive nonlinear stabilization control for a quadrotor UAV: theory, simulation and experimentation. J. Intell. Robot. Syst. **72**, 105–122 (2013). doi:10.1007/s10846-013-9813-y
4. Besnard, L., Shtessel, Y.B., Landrum, B.: Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer. J. Franklin Inst. **349**, 658–684 (2012)
5. Efe, M.: Battery power loss compensated fractional order sliding mode control of a quadrotor UAV. Asian J. Control **14**(2), 413–425 (2012)
6. Huang, B.B., Chun-Juan, L.I., Yong, H.E., et al. Back-stepping control for an under-actuated micro-quadrotor. J. Luoyang Inst. Sci. Technol. **23**(2), 56–61 (2013)
7. Honglei, A., Jie, L., Jian, W., et al.: Backstepping-based inverse optimal attitude control of quadrotor. Int. J. Adv. Robot. Syst. **10** (2013)
8. Raffo, G.V., Ortega, M.G., Rubio, F.R.: An integral predictive/nonlinear H ∞ control structure for a quadrotor helicopter. Automatica **46**, 29–39 (2010)
9. Wang, D.L, Qiang, L., Liu, F.: Quadrotor attitude control based on L_1 neural network adaptive control method. Comput. Eng. Des. **33**(12), 4758–4761 (2012)
10. Li, J., Li, Y.: Dynamic analysis and PID control for a quadrotor. In: 2011 International Conference on Mechatronics and Automation (ICMA), pp. 573–578. IEEE (2011)
11. Kendoul, F., Lara, D., Fantoni-Coichot, I., et al.: Real-time nonlinear embedded control for an autonomous quadrotorn helicopter. J. Guid. Control Dyn. **4**, 1049–1061 (2007)
12. Wang, W., Ma, H., Xia, M., et al.: Attitude and altitude controller design for quad-rotor type MAVs. Math. Prob. Eng. **58**(7), 98–107 (2013)
13. Dierks, T., Jagannathan, S.: Neural network output feedback control of a quadrotor UAV[C]. In: 47th IEEE Conference on Decision and Control, pp. 3633–3639. CDC 2008, IEEE (2008). doi:10.1109/CDC.2008.4738814
14. Chang, X.U., Jian-Hong, L., Cheng, M., et al. Neural network PID adaptive control and its application. Control Eng. China **14**(3), 284–286 (2007)