An Attitude Control Method of Aircraft Based on Neural Network Batch Training Strategy*

Yitong Li
National Key Laboratory of Science and Technology on Aerospace Intelligent Control
Beijing Aerospace Automatic Control Institute
Beijing, China
aprillee105@163.com

Xiaodong Wang
China Academy of Launch Vehicle Technology
Beijing, China
e.dong@263.net

Huiping Zhang
Beijing Aerospace Automatic Control Institute
Beijing, China
yuanhaoht@126.com

Xiaodong Liu
Beijing Aerospace Automatic Control Institute
Beijing, China
k.start@163.com

2020 Chinese Automation Congress (CAC) | 978-1-7281-7687-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/CAC51589.2020.9327655

***Abstract*—In order to solve the problem of large workload of the manual design of aircraft attitude control parameters in practical engineering, a neural network PID controller based on batch input training strategy is proposed in this paper. The basic framework of model reference adaptive control is adopted, and the self-learning characteristic of neural network is used to realize the adaptive adjustment of PID control parameters. In this control system, the neural network input is changed from serial input to batch input. The simulation results show that the neural network training method based on batch input has better training results than the original serial input method.**

Keywords—neural network, PID controller, adaptive, attitude control

I. INTRODUCTION

In practical engineering applications, gain scheduled is usually adopted for the aircraft control parameter design. Firstly, the designer designs multiple sets of PID control parameters for different feature points, and then these parameters are loaded on the flight control computer, scheduling according to certain criteria to realize the change of control parameters. This method requires the designers to have high practical experience, the design process is tedious and consumes resources. In this paper, combining the neural network with PID controller can reduce human participation and workload to a certain extent.

In recent years, in various fields, the literature about the combination of neural network and PID controller emerge in endlessly. REN et al. applied this method to the temperature control [1]. ZHANG et al. applied this method to the automatic rudder control of ship, which can track the rudder command with high accuracy [2]. YUAN et al. applied it to the wind pendulum system to improve the stability and control accuracy of the control system [3]. Jun Kang et al. considered the multi-input and multi-output system, the scale of neural network expands with the increase of the input and output dimension [4]. However, in these studies, neural network training is based on a single point, that is, each input of the network is one-dimensional, and this training method does not use the information before this time point. When the characteristics of the controlled object are good, the neural network training result is effective, but when the characteristics of the controlled object are not good, the training effect are not significant. Therefore, on the basis of the original BP neural network training, this paper proposes two training methods based on the batch input neural network, and the simulation results verify that the batch input training method with fixed data size is the best.

National Natural Science Foundation of China (Grant Nos. 61803357).

II. BP NEURAL NETWORK PID CONTROL BASED ON SINGLE POINT INPUT

A. The Basic Framework of BP Neural Network

Three-layer BP neural network is adopted in this paper, which contains input layer *I*, hidden layer *H* and output layer *O*. The structure of neural network is shown in Fig.1.

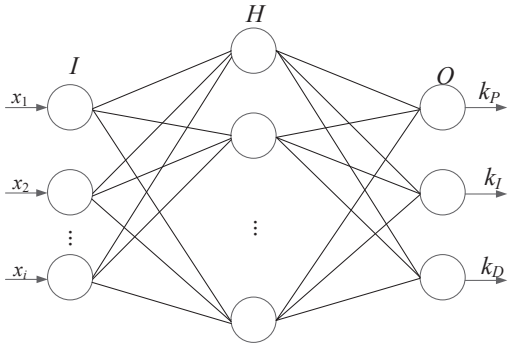

Fig. 1. Structure of BP neural network.

The neuronal nodes of input layer *I* do not use activation function, so its outputs are the same as inputs, which can be illustrated as:

$$I_i^{in}(k) = I_i^{out}(k) = x_i(k), i = 1, 2, ..., M \tag{1}$$

Where superscript *in* and *out* means the input and output respectively, k is the current time, and M is the number of input layer neural nodes. The input variables are all one-dimensional.

The input and output of hidden layer H at k are shown as:

$$H_h^{in}(k)=\sum_{i=1}^{M} w_{ih}^{IH}(k) I_i^{out}(k),\ \ h=1,2,...,N \tag{2}$$

$$H_h^{out}(k)=f\left(H_h^{in}(k)\right) \tag{3}$$

Where w_{ih}^{IH} is the network weights between input layer and hidden layer, and N is the number of hidden layer neural nodes. Tan-Sigmoid function is used as activation function of hidden layer nodes, and it is symmetric about the coordinate origin. The Tan-Sigmoid function is:

$$f(x)=\tanh(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{4}$$

The input and output of output layer O at k are shown as:

$$O_o^{in}(k)=\sum_{h=1}^{N} w_{ho}^{HO}(k) H_h^{out}(k),\ o=1,2,3 \tag{5}$$

$$O_o^{out}(k)=g\left(O_o^{in}(k)\right) \tag{6}$$

Where w_{ho}^{HO} is the network weight between input layer and hidden layer. And the output is PID parameters, which can also be illustrated by:

$$O_1^{out}(k)=k_P(k) \tag{7}$$

$$O_2^{out}(k)=k_I(k) \tag{8}$$

$$O_3^{out}(k)=k_D(k) \tag{9}$$

The output coefficients of PID parameters are 5, 0.1 and 1. The parameters cannot be negative, so the nonnegative Log-Sigmoid function is adopted as activation function of hidden layer nodes:

$$g(x)=\frac{1}{2}\left(1+\tanh(x)\right)=\frac{e^{x}}{e^{x}+e^{-x}} \tag{10}$$

B. BP Neural Network Weights Update Rule

The performance evaluation function is calculated as follows:

$$J(k)=\frac{1}{2}\left[y_d(k)-y(k)\right]^2=\frac{1}{2}e^2(k) \tag{11}$$

Where $y_d(k)$ is expected output, $y(k)$ is the feedback value from controlled object at k. Take the square of the difference between the two values and the final goal of training is to make $J(k)$ converge to a minimum. Steepest descent method is used to adjust the network weights of each layer. During the training, network weights update layer by layer from the output layer along the direction that the performance evaluation value decreases. So the gradient of $J(k)$ to w_{ho} should be calculated, and the network weights between output layer and hidden layer are updated by:

$$\Delta w_{ho}^{HO}(k)=-\eta_3\frac{\partial J(k)}{\partial w_{ho}^{HO}}+\alpha_3\Delta w_{ho}^{HO}(k-1) \tag{12}$$

Where $\eta_3>0$ is learning rate, $\alpha_3>0$ is momentum factor, and $\alpha_3\Delta w_{ho}^{HO}(k-1)$ is inertia item. The equation xx indicates that the adjustment of network weights is related two parts: one is the gradient of $J(k)$ to w_{ho}^{HO} at k; the other one is the direction and value of the weight increment at k-1. This way can increase the inertia of weight update to a certain extent, and make the network training converge smoother and faster.

According to the chain rule, the gradient $\frac{\partial J(k)}{\partial w_{ho}^{HO}}$ is calculated as follow:

$$\begin{aligned}
&\frac{\partial J(k)}{\partial w_{ho}^{HO}}\\
&=\frac{\partial J(k)}{\partial y(k)}\cdot\frac{\partial y(k)}{\partial \Delta u(k)}\cdot\frac{\partial \Delta u(k)}{\partial O_o^{out}(k)}\cdot\frac{\partial O_o^{out}(k)}{\partial O_o^{in}(k)}\cdot\frac{\partial O_o^{in}(k)}{\partial w_{ho}^{HO}(k)}\\
&=-e(k)\cdot\frac{\partial y(k)}{\partial \Delta u(k)}\cdot\frac{\partial \Delta u(k)}{\partial O_o^{out}(k)}g'\left(O_o^{in}(k)\right)\cdot\frac{\partial O_o^{in}(k)}{\partial w_{ho}^{HO}(k)}\\
&=\frac{\partial J(k)}{\partial O_o^{in}(k)}\cdot\frac{\partial O_o^{in}(k)}{\partial w_{ho}^{HO}(k)}
\end{aligned} \tag{13}$$

Due to (5):

$$\frac{\partial O_o^{in}(k)}{\partial w_{ho}^{HO}(k)}=H_h^{out}(k) \tag{14}$$

The $\frac{\partial y(k)}{\partial \Delta u(k)}$ is replaced with sign function $\operatorname{sgn}\left(\frac{\partial y(k)}{\partial \Delta u(k)}\right)$ in order to eliminate the effect of system uncertainty, and use learning rate η_3 to compensate calculation uncertainty.

The expression of incremental PID control law is:

$$\begin{aligned}\Delta u(k) &= k_P\left[e(k)-e(k-1)\right]+k_I e(k)\\ &+k_D\left[e(k)-2e(k-1)+e(k-2)\right]\end{aligned} \tag{15}$$

From (7) ~ (9) and (15) we can get:

$$\frac{\partial \Delta u(k)}{\partial O_1^{out}(k)} = e(k)-e(k-1) \tag{16}$$

$$\frac{\partial \Delta u(k)}{\partial O_2^{out}(k)} = e(k) \tag{17}$$

$$\frac{\partial \Delta u(k)}{\partial O_3^{out}(k)} = e(k)-2e(k-1)+e(k-2) \tag{18}$$

According to the above derivation, the network weights update law between the hidden layer and output layer is as follows:

$$\Delta w_{ho}^{HO}(k) = \alpha_3 \Delta w_{ho}^{HO}(k-1)+\eta_3 \delta_o^{HO}(k) H_h^{out}(k) \tag{19}$$

$$\begin{aligned}\delta_o^{HO}(k) &= \frac{\partial J(k)}{\partial O_o^{in}(k)}\\ &= e(k)\operatorname{sgn}\left(\frac{\partial y(k)}{\partial \Delta u(k)}\right)\frac{\partial \Delta u(k)}{\partial O_o^{out}(k)} g'\left(O_o^{in}(k)\right)\end{aligned} \tag{20}$$

Where $g'(x) = 2g(x)\left[1-g(x)\right]$.

Use the same method, the network weights update law between the hidden layer and input layer can be calculated by the following process.

Firstly, calculate the gradient:

$$\begin{aligned}&\frac{\partial J(k)}{\partial w_{ih}^{IH}(k)}\\ &= \sum_{o=1}^{3}\frac{\partial J(k)}{\partial O_o^{in}(k)}\cdot\frac{\partial O_o^{in}(k)}{\partial H_h^{out}(k)}\cdot\frac{\partial H_h^{out}(k)}{\partial H_h^{in}(k)}\cdot\frac{\partial H_h^{in}(k)}{\partial w_{ih}^{IH}(k)}\\ &= \sum_{o=1}^{3}\frac{\partial J(k)}{\partial H_h^{in}(k)}\cdot\frac{\partial H_h^{in}(k)}{\partial w_{ih}^{IH}(k)}\end{aligned} \tag{21}$$

where

$$\begin{aligned}\frac{\partial J(k)}{\partial H_h^{in}(k)} &= \sum_{o=1}^{3}\frac{\partial J(k)}{\partial O_o^{in}(k)}\cdot\frac{\partial O_o^{in}(k)}{\partial H_h^{out}(k)}\cdot\frac{\partial H_h^{out}(k)}{\partial H_h^{in}(k)}\\ &= \sum_{o=1}^{3}\delta_o(k) w_{ho}^{HO}(k)\cdot f'\left(H_h^{in}(k)\right)\end{aligned} \tag{22}$$

where $f'(x) = 1-f^2(x)$.

The network weights update law between the hidden layer and input layer is:

$$\Delta w_{ih}^{IH}(k) = \alpha_2 \Delta w_{ih}^{IH}(k-1)+\eta_2 \delta_h^{IH}(k) I_I^{out}(k) \tag{23}$$

Where $\eta_2 > 0$ is learning rate, $\alpha_2 > 0$ is momentum factor, and δ_h^{IH} is calculated by:

$$\delta_h^{IH}(k) = f'\left(H_h^{in}(k)\right)\sum_{o=1}^{3}\delta_o^{HO}(k) w_{ho}^{HO}(k) \tag{24}$$

III. BP NEURAL NETWORK PID CONTROL BASED ON BATCH INPUT

The most common method of neural network training is as follow: under the condition of the training sample set, the input data in the sample set is sent into the neural network. The data is propagated backwards from the input layer to output layer and get the corresponding output. Take the difference between the current neural network output and the expected output in the sample set as current error. Then the weights of the neural network are adjusted layer by layer from the output layer along the direction of error reduction, and the process is repeated until the error converges to an acceptable range.

However, in the system combining BP neural network and PID controller, there is no traditional training sample set, which can also be seen from the derivation in II. The output of BP neural network is PID parameters, and then the parameters are applied on controlled object. The output of the controlled object is sent to neural network as input. It is a real-time feedback loop process. Therefore, generally, each input of the neural network is one-dimensional, that is, the number of sample selected for each training loop is 1. For the controlled object with good characteristics, this training method can fully

meet the requirement, but when the controlled object has bad characteristics, this training method is not effective.

Based on the original neural network PID control, this paper expands the number of samples selected for each training to loop achieve batch input. This can be achieved by adding a period of time prior to the current time to the neural network input, that is, each x_i in Fig.1 is changed from an number to an array. There are two types of batch input as follow.

A. Batch Input Method of Changed Data Size

Fig.2 describes the data selection process. k is the current time, M is the maximum number of each input of the neural network. Assuming that each input dimension of the neural network at time k is 1, then it is 2 at time k+1. The number increases successively to M, then to 1, and this process proceeds back and forth. As the dimension of the network input expands, the output will also expand correspondingly. However, each loop can only output one set of control parameters, the average processing of multiple sets of PID parameters should be carried out.

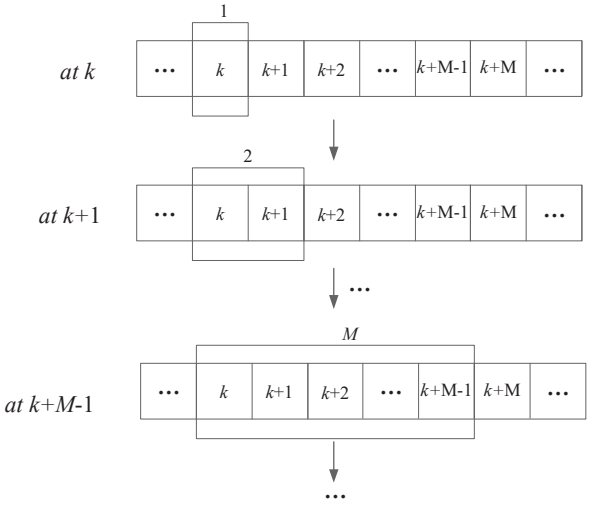

Fig. 2. Schematic diagram of batch input with changed data size.

B. Batch Input Method of Fixed Data Size

Different from the former method of data selection, each input data amount of the neural network is fixed in this way, which is M. M data before the current time is sent to neural network in each loop, and the process is shown in Fig.3.

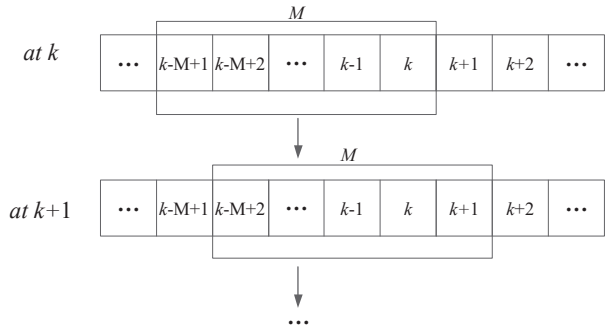

Fig. 3. Schematic diagram of batch input with fixed data size.

IV. Simulation

The pitch channel of a missile's powered phase flight segment is selected as the controlled object, and its minor deviation linearization model is as follows:

$$\begin{aligned} \Delta\dot{\theta} &= c_1\Delta\alpha + c_2\Delta\theta + c_3\Delta\delta \\ \Delta\ddot{\varphi} + b_1\Delta\dot{\varphi} + b_2\Delta\alpha + b_3\Delta\delta &= 0 \\ \Delta\varphi &= \Delta\theta + \Delta\alpha \end{aligned} \tag{25}$$

where $\Delta\theta$ is trajectory inclination angle deviation, $\Delta\alpha$ is attack angle deviation, $\Delta\delta$ is equivalent deflection angle of attitude control nozzle, $\Delta\varphi$ is pitch angle deviation, c_1 is dynamic coefficient associated with thrust and lift, c_2 is gravity coefficient, c_3 is control coefficient, b_1 is aerodynamic damping moment coefficient， b_2 is aerodynamic torque coefficient，b_3 is control torque coefficient.

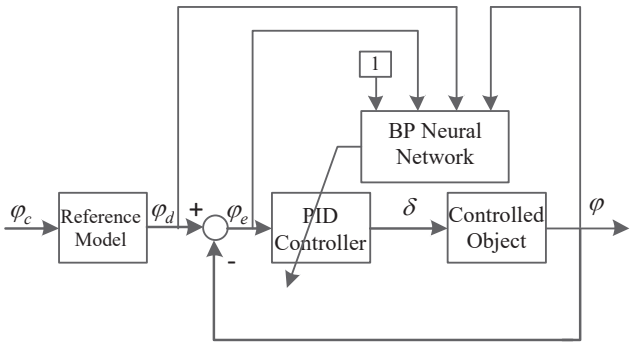

Fig. 4. Block diagram of model reference adaptive control scheme.

The model reference adaptive control scheme in [5] is adopted as the control system framework, which can smooth the system transition process [6], reduce the system response overshoot and the control variable, and this design can decrease the waste of control energy to a certain extent.

The control system framework is built in the SIMULINK toolbox for simulation, and the simulation step size is set as 0.01s. The characteristic point of missile model with the maximum control torque coefficient is selected for simulation. The reference model is selected as follows:

$$G_m(s) = \frac{25}{s^2 + 9s + 25} \tag{26}$$

Under the same setting parameters, three methods of single point input, batch input with changed data size and batch input with fixed data size are trained respectively. As shown in Fig.4, the neural network input is $[\varphi_d; \varphi_e; \varphi; 1]$, and the four inputs are arrays when it is batch input. The network structure is 4-5-3, the learning rate is set as 0.5, the momentum factor is set as 0.05, the maximum number M of each input of the neural network is set as 100, and the initial weights of the neural network is selected as a random number between [-0.5,0.5]. To simplify the simulation, step signal is

used as the pitch angle command. The initial weights of neural network are random, so this paper selects three groups of initial values for comparison, and the initial weights of the three input methods in each group are the same.

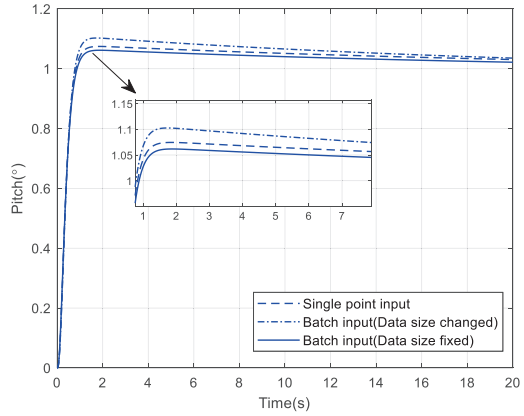

Fig. 5. Comparison graph of response for different data input methods(the first group).

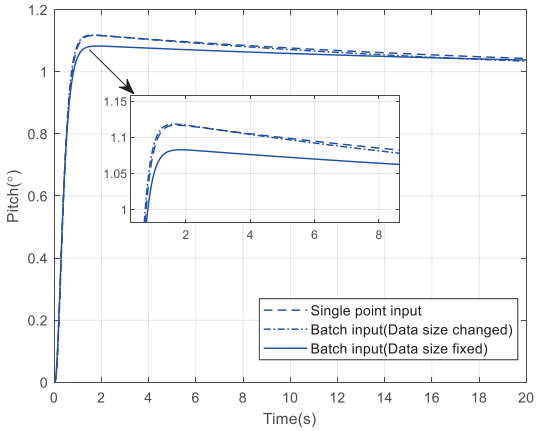

Fig. 6. Comparison graph of response for different data input methods(the second group).

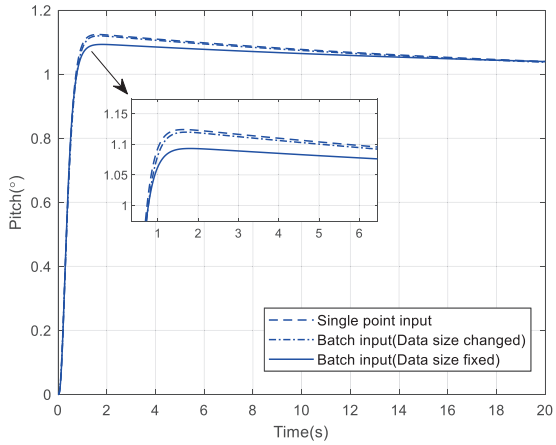

Fig. 7. Comparison graph of response for different data input methods(the thrid group).

From Fig.5 to Fig.7, it can be seen that under the same condition, the tracking response overshoot obtained by batch input training with fixed data size is the lowest, while the training effect of single point input and batch input with changed data size is bad

Take the batch input with fixed data size in the first group as an example, and its control parameters and rudder angle are shown in Fig.8 and Fig.9. From the results, it can be seen that the control parameters can converge to a certain value quickly, and the tracking effect meets the practical engineering requirements.

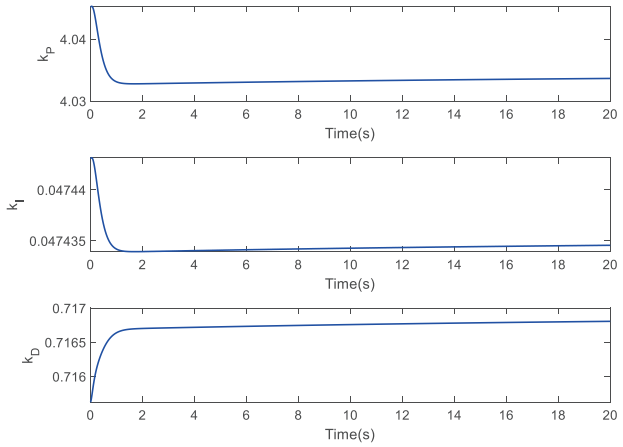

Fig. 8. Curve of control parameters.

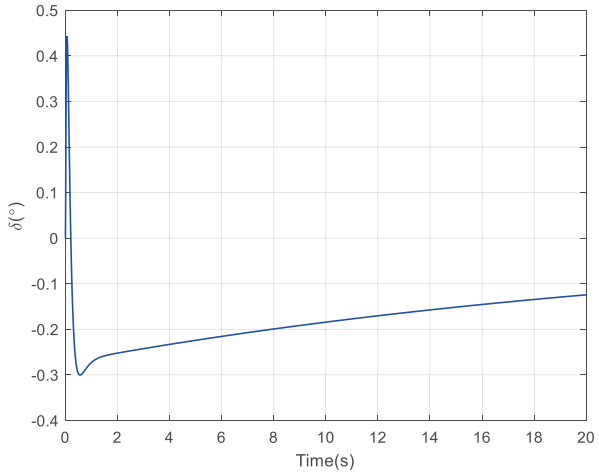

Fig. 9. Curve of the rudder angle of the attitude control nozzle.

V. CONCLUSION

Based on the training method combining neural network and PID control, a neural network training method based on batch input is proposed in this paper. The information of a period of time before the current moment is also sent to the neural network as input, and each neural network input is changed from one dimension to multi-dimension. The simulation results show that when the characteristic of controlled object is bad, the method proposed in this paper can

achieve better control and tracking effect than the single point input, and it also provides an exploratory foundation for intelligent control of aircraft attitude to a certain extent.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (Grant Nos. 61803357). The author would like to thank Dr. Liu for reviewing this paper.

REFERENCES

[1] REN Youzhi, QIAO Song, SUN Jichun, MAN Xiaofei. Application of BP neural network PID controller in temperature control of thermal oil boiler[J]. Electric Drive,2020,50(04):81-84(in Chinese).

[2] ZHANG Jingwen, ZHANG Qingsong. Control method of ship autopilot based on BP neural network and PID[J]. Ship Science and Technology, 2019,41(16):124-126(in Chinese).

[3] YUAN Jian-ping, SHI Yi-ping, JIANG Peng, JIA Rijing, YAO Deliang. Design of wind pendulum control system based on BP neural network PID algorithm[J]. Measurement & Control Technology, 2018,37(11):144-147+158(in Chinese).

[4] Jun Kang, Wenjun Meng, Ajith Abraham, Hongbo Liu. An adaptive PID neural network for complex nonlinear system control[J]. Neurocomputing,2014,135.

[5] Liu Xiaodong, Ma Fei, Zhang Yu, Du Lifu. Model reference adaptive attitude control technology based on BP neural network[J]. Aerospace Control,2019.37(06):3-7(in Chinese).

[6] Dan Zhang,Bin Wei. A review on model reference adaptive control of robotic manipulators[J]. Annual Reviews in Control,2017,43.3.