Three-Axis Balanced Camera Stabilizer Design Based on Neural Network BP Algorithms Combine PID Control

Xiao-hui CHENG* and Yu ZHANG

Guangxi Key Laboratory of Embedded Technology and Intelligent System, Guilin University of Technology, Institute of Information Science and Engineering, Guilin, Guangxi, China

*Corresponding author

Keywords: Neural network, PID control, Triaxial balance stabilizer.

Abstract. In this paper, the self-learning ability of BP neural network is used to achieve the purpose of adaptive tuning of the PID parameters of the three-axis balanced camera stabilizer, so as to reduce the blur caused by external factors such as jitter of the camera image. Through the corresponding simulation experiments, it is verified that the PID controller can control the PID controller through the BP neural network, which can make the system have higher precision and stronger stability, can maintain the balance of the picture, and achieve precise control effect on the position of the camera.

Introduction

In the trend of the AI era, many smart devices have emerged. Artificial intelligence technology has enabled us to consistently make life easier.

The object of the adaptive control system studied in this paper is the unstable nonlinear system. The control method based on PID control theory is combined with BP neural network, and the adaptive learning of neural network is used to control and adjust the parameters of PID. The neural network PID control formed by the combination of the two not only has the advantages of traditional PID control, but also can adjust the corresponding unstable parameters through the autonomous learning and adaptive algorithm of the neural network to optimize the entire unstable nonlinear system. The simulation results also verify that the control method of the neural network BP algorithm is effective.

PID Control

PID (Proportional Integral differential) control refers to proportional, integral, and derivative control. PID control technology is divided into analog PID and digital PID. With the development of chip technology, the product will always choose a flexible digital PID control. The basic principle block diagram of PID control is shown in Figure 1.

![Figure 1. PID basic principle block diagram.](image-url)
u(t) is the control quantity output and e(t) is the given deviation value. K_p is the proportional gain, T_i is the integral time constant, and T_d is the differential time constant. The corresponding input and the corresponding PID control transfer function is:

\[ G(s) = \frac{Y(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \]

(1)

The digital PID can be obtained by discretizing the analog PID. The cumulative error of the positional PID control relative to the incremental PID is larger.

The transfer function of the incremental PID control is:

\[ \Delta y_n = y_n - y_{n-1} = K_p \left[ (e_n - e_{n-1}) + \frac{T}{T_i} e_n + \frac{T_d}{T} (e_n - 2e_{n-1} + e_{n-2}) \right] \]

(2)

According to the requirements of the camera three-axis balance stabilizer for the relevant performance, we choose the incremental PID as the Designed control algorithm.

**BP Algorithm for PID Control of Three-axis Balance Stabilizer**

**BP Neural Network Model**

Figure 2 shows the structure of a typical three-layer BP neural network. In Figure 2, there are n neurons in the input layer, p neurons in the hidden layer, and q neurons in the output layer. In the forward transfer, the input signal vector \( x_i \) is transmitted layer by layer from the input layer through the hidden layer, and finally to the output layer. If there is an error between the predicted value \( k \) and the target value \( K \), calculate the error portion and reverse the layer-by-layer transmission, and adjust the weight of each layer connection in the error reduction direction threshold. The above process is repeatedly performed until the entire learning process ends.

\[ \text{Input} = \text{net} = x_1 w_1 + x_2 w_2 + \cdots + x_n w_n \]

(3)

\[ \text{Output derivative} = f'(\text{net}) = \frac{1}{1+e^{-\text{net}}} - \frac{1}{(1+e^{-\text{net}})^2} = y(1-y) \]

(4)
Combination of BP Algorithm and PID control

The traditional PID control is to directly control the controlled object in closed loop, and calculate the corresponding $K_p$, $T_i$ and $T_d$ parameter values according to the formula, and adjust the whole system parameters. The core of a stabilizer is the algorithm, which determines the stability of the stabilizer control.

Since the transfer function of the BP neural network must be divisible, it is generally chosen to use the S-type function as its transfer function. The specific algorithm flow is as follows:

a. Setup initialization

Assigned an interval (-1,1) random number inside, The error function $\epsilon$ is set, given a calculation accuracy value $\epsilon = 0.02$~0.05 and a maximum number of learning times $N = 60$. Wherein, the input vector is: $x = (x_1, x_2, ..., x_n)$; the hidden layer input vector is: $\Phi_i = (\Phi_{i1}, \Phi_{i2}, ..., \Phi_{in})$; the hidden layer output vector is: $\Phi_o = (\Phi_{o1}, \Phi_{o2}, ..., \Phi_{on})$; the input layer input vector is: $y_i=(y_{i1}, y_{i2},...,y_{in})$; the output layer output vector is: $y_o=(y_{o1}, y_{o2},...,y_{on})$; the expected output vector is: $d_o=(d_{o1}, d_{o2}, ..., d_{on})$.

b. Select the k-th camera position input sample and corresponding desired output

$$y_{o0}(k) = f(y_{i0}(k)) \quad o = 1,2,L \quad q$$ (5)

c. Camera position test signal forward propagation process

Calculate the input and output of the camera position signal. The three nodes of the output correspond to $K_p$, $K_i$, and $K_d$.

$$\Phi_{ih}(k) = \sum_{i=1}^{n} w_{ih} x_i(k) - a_h \quad h = 1,2, ... p$$ (6)

$$\Phi_{oh}(k) = f(h_{i0}(k)) \quad h = 1,2, ... p$$ (7)

$$y_{i0}(k) = \sum_{h=1}^{p} w_{ho} h_o(k) - a_o \quad o = 1,2, ... q$$ (8)

d. Test error back propagation process

Using the desired position signal output and the actual position signal output, calculate the partial derivative of the error function for each neuron in the output layer $\delta_o(k)$.

$$\frac{\partial e}{\partial y_{i0}} = \frac{\partial \epsilon}{\partial y_{i0}} = \frac{\partial \epsilon}{\partial y_{i0}} \frac{\partial y_{i0}}{\partial y_{i0}} = -(d_o(k) - y_o(k)) f'(y_{i0}(k)) \delta_{o}(k)$$ (9)

e. Using the connection weight of the hidden layer to the output layer, the output layer $\delta_o(k)$ and the output of the hidden layer, calculate the partial derivative $\delta_h(k)$ of the error function for each neuron in the hidden layer.

$$\frac{\partial e}{\partial w_{ho}} = \frac{\partial e}{\partial y_{i0}} \frac{\partial y_{i0}}{\partial y_{i0}} = -\delta_{o}(k) \Phi_{oh}(k)$$ (10)

$$\frac{\partial e}{\partial w_{ih}} = \frac{\partial e}{\partial h_{ih}(k)} \frac{\partial h_{ih}(k)}{\partial w_{ih}}$$ (11)
\[ \frac{\partial \phi_i(k)}{\partial w_{ih}} = \frac{\partial (\sum_{i=1}^{n} w_{ih} x_i(k) - \alpha_h)}{\partial w_{ih}} = x_i(k) \]  

(12)

\[ \frac{\partial e}{\partial \phi_i(k)} = \frac{\partial \left( \frac{1}{2} \sum_{o=1}^{q} \left( w_{ho} \phi_o(k) \right)^2 \right)}{\partial \phi_i(k)} = -\left( \sum_{o=1}^{q} \delta_o(k) w_{ho} f'(\phi_i(k)) \right) @ \delta_h(k) \]  

(13)

\[ \Delta w_{ho}(k) = -\mu \frac{\partial e}{\partial w_{ho}} = \mu \delta_o(k) \phi o_h(k) \]  

(14)

\[ w_{ho}^{N+1} = w_{ho}^N + \eta \delta_o(k) \phi o_h(k) \]  

(15)

\[ \Delta w_{ih}(k) = -\mu \frac{\partial e}{\partial w_{hi}} = -\mu \frac{\partial e}{\partial \phi_i(h(k))} = \delta_h(k) x_i(k) \]  

(16)

\[ w_{ih}^{N+1} = w_{ih}^N + \eta \delta_h(k) x_i(k) \]  

(17)

\[ E = \frac{1}{2m} \sum_{k=1}^{m} \sum_{o=1}^{q} \left( d_o(k) - y_o(k) \right)^2 \]  

(18)

f. Correct the connection weight \( w_{ho}(k) \) using the output of the \( \delta_o(k) \) of the output layers \( K_p, K_i, K_d \) and the output of each neuron in the hidden layer.

g. Use the \( \delta_h(k) \) of each neuron in the hidden layer and the input of each neuron in the input layer to correct the connection weight.

h. Calculate the global error \( E \)

i. When the error reaches the preset accuracy of 0.03 to 0.05 or the number of learning times is greater than 60, end the algorithms.

Three-axis Balanced Camera Stabilizer Model

The model of the three-axis balance stabilizer is shown in Figure 4. When the camera position changes, the chip monitors the camera position change, the BP neural network calculates the difference between the feedback value and the preset value, and learns the parameter autonomously. Tuning, achieving the goal of smooth camera shooting.

![Figure 4. Three-axis camera balance stabilizer model.](image-url)
Simulation Experiment Results

This paper simulates the PID control triaxial balance camera stabilizer system based on BP neural network tuning. Through simulation experiments, when kp=8, ki=0.10, kd=10. According to the incremental PID control algorithm, the following objects are controlled:

\[
G(s) = \frac{400}{s^2+60s}
\]

Through simulation experiments, when \(k_p=8, k_i=0.10, k_d=10\).

\[
\text{(19)}
\]

It can be seen from the simulation curves in Figure 5 and Figure 6. The BP neural network PID controller has short adjustment time, good tracking characteristics, smooth transition of adjustment process, small overshoot, and can quickly meet control requirements, indicating that it is controlled by neural network. PID has good control characteristics.

Acknowledgments

As the research of the thesis is sponsored by National Natural Science Foundation of China (No: 61662017, No: 61262075), Key R & D projects of Guangxi Science and Technology Program (Gui's AB17195042), Guangxi Key Laboratory Fund of Embedded Technology and Intelligent System, we would like to extend our sincere gratitude to them.

References

[1] L Ma, Y Yao, M Wang. The Optimizing Design of Wheeled Robot Tracking System by PID Control Algorithm Based on BP Neural Network. C. IEEE International Conference on Industrial Informatics-computing Technology, (2017)34-39

[2] Fuqiagn, Gao. Ping, Li, Leimin, Zhang. PID Control and Simulation Based on BP Neural Network Tuning J. Shandong Ceramics. 40(2017)27-31.

[3] Xianhong, Li, Haibin, Yu, Peng, Zeng. Design method of saturation optimal PID controller with low order controlled process J. Information and Control. 1(2015) 43-50.

[4] Zongbo, Hou, Zhiqing, Ma, Xiangjun, Meng. MR brain tissue image segmentation method based on improved genetic neural network J. Modern Electronics Technique. 19(2015) 60-63.

[5] Ping, Guan, Zhiwei, He, Xinsheng, Ge. Neural network PID control of hypersonic vehicle J. Aerospace Control. 36(2018) 15-17.

[6] Tian Wei, Peng Yizhun, Wang Pan, et al. A application of BP neural network PID algorithm in refrigerated space temperature. J. Applied Mechanics and Materials. 599(2014) 827-830.