Journal of Electronics and Information Science (2023)
Clausius Scientific Press, Canada
DOI: 10.23977/jeis.2023.080109
ISSN 2371-9524 Vol. 8 Num. 1

Research on CMAC and PID Compound Control Based on GUI

Zhongqiao Zheng

School of Electrical and Information Engineering, Changzhou Institute of Technology, Changzhou, China

Keywords: CAMC, PID controller, composite control, GUI, control strategy

Abstract: The control structure and control strategy of CMAC (Cerebellar Model Articulation Controller) and PID parallel control system are designed based on the good nonlinear approximation ability and adaptation of CMAC neural network, combined with the advantages of simple structure and easy operation of common PID controller. The compound control of CMAC and PID is carried out for typical second-order control system and a good GUI (Graphical User Interface) is given. The simulation results show that, compared with the traditional PID control, the rise time of the control system by CMAC and PID composite control strategy is reduced by 0.01s, the adjustment time is reduced by 0.1s, and the overshoot is reduced by 15%. The control system has been significantly improved in the steady-state performance and dynamic performance, and has good control performance.

1. Introduction

In 1975, J.S. Albus proposed CMAC that simulates the structure and function of human cerebellum [1]. It is a learning structure of simulating human cerebellum. It is a local neural network model based on TabIe Look input and output. It has the ability of information classification storage and nonlinear approximation. It is a simple and fast neural network for local approximation. It is similar to Perception's associative memory method. It can learn multi-dimensional nonlinear mapping and has strong robustness and reliability, especially suitable for complex nonlinear and time-varying working environment. As we all know, the cerebellum of human is responsible for the coordination and control of human motion. From the perspective of system, it can be seen as a function. Its function is to store and retrieve several muscle control information that produces coordinated actions. According to the existing literature, the research on cerebellar model neural control mainly includes two parts, the first is the research on the performance of CMAC neural network, the structure, parameters and learning ability of CMAC are improved by combining with other algorithms, the second is to combine CMAC with advanced control strategies to form a hybrid control method according to the strong robustness and reliability of CMAC [2-4]. The document [5] proposes a parallel control strategy of fuzzy PID and CMAC, which learn and interpolate the fuzzy PID control process by its adaptive ability and fast learning non-linear function ability, and obtains a good control effect. The document [6] proposes a hybrid control strategy combining CMAC and BP neural network and applied to short-term prediction of photovoltaic power. In the industrial control system, because PID controller has the advantages of simple structure, easy adjustment, convenient operation, etc., PID controller has always been used in the industrial fields, and it can basically achieve good control effect on the industrial process with accurate mathematical model [7-10]. However, when the controlled object is in a relatively complex environment, the parameters will change with the time and the working environment, or the controlled process is relatively complex and has nonlinearity, pure delay and time varying, an ordinary PID controller can't guarantee robustness and anti-interference ability of control system. Therefore, a compound controller based on CMAC and PID is designed by combining advantage of CMAC and PID controller. It can quickly identify the controlled process parameters, change according to the changes of the controlled parameters, and automatically adjust the control parameters. It solves some problems of the traditional PID controller, such as the difficulty in parameter adjustment, the inability to adjust parameters in real time, and the weak robustness. It is suitable for the more complex control environment.

2. Principle of CMAC Neural Network

CMAC neural network is a neural network model, which is based on principle of cerebellum controlling limb movement. It is nonlinear and an adaptive neural network. It is a neural network model based on the principle of cerebellum controlling limb movement. CMAC is a neural network with complex nonlinear function, and also an adaptive neural network. It takes input state as a pointer and stores relevant information into the corresponding storage unit address. CMAC is a table lookup technology in principle. The detailed implementation method is to divide the large block of input space into many small blocks, and the information of each small block is stored in the location of adjacent blocks in a distributed manner. Each small block has an actual corresponding to storage location. The number of storage units is less than blocks in maximum possible input space of the problem under consideration, so the mapping of multiple block to one block can be achieved, that is, the mapping of multiple blocks to the same block, so that it can be stored in one address through the memory address. CMAC is used in nonlinear function approximation, control system and dynamic model design to effectively improve efficiency of the algorithm [11-12]. Compared with other neural networks, it's advantages are follows.

1) Continuous input and output capability.
2) It has certain generalization ability and unified planning.
3) Fast learning speed, fast response speed and high working efficiency
4) The nonlinear approximator is not very sensitive to the data order.

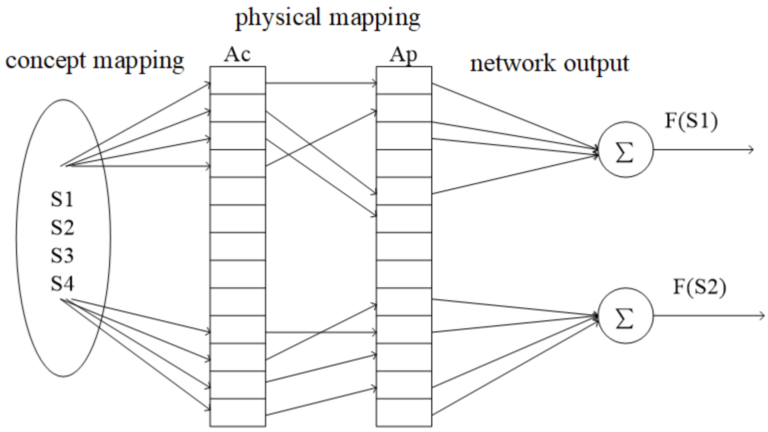

Figure 1: Topological structure of CMAC

Due to the advantages and high efficiency of CMAC, it has better nonlinear approximation ability, it is more suitable for nonlinear real-time control system [13,14]. The structure of CMAC neural network is shown in Figure 1. It includes concept mapping, physical mapping, input and output.

Each input state of the controller is taken as a state vector and an input space is formed, S is for the input space, S1, S2, S3, S4 are the state vectors. Concept mapping refers to the mapping from input space S to concept memory Ac. N-dimensional input space is divided at input layer, and each input falls into a cube cell of N-dimensional grid base. The adjacent two points in the input space motivate the partially coincident cells in the memory Ac by mapping. The closer the two points are, the more the coincident parts are. The farther away points will not overlap in the Ac, thus the local generalization of the network is realized. Physical mapping (actual mapping is for Ac→Ap) refers to that in the mapping process, the input sample is mapped to the address of the concept memory, the remainder obtained is taken as the address of the actual memory Ap, that is, c cells in concept memory are mapped to c in Ap. Output of CMAC (Ap→y) refers to input is mapped to c cells in Ap, corresponding weight value is stored in each cell. The output of CMAC is the weighted sum of c actual storage cells [15-18].

CMAC neural network consists of two input layers (nonlinear and adjustable linear), and its structure is shown in Figure 2.

Assume that the function mapping relationship to be approximated by CMAC is follows.

$$y = f(x) \tag{1}$$

where, $x = [x_1, x_2, ..., x_n]^T$, $y = [y1, y2, ..., yT]^T$ .According to Figure 2, the below mappings are obtained.

1) S:x→A, that is a=S(x)

For x, only a few elements are 1, and most elements are 0. This is the function implemented by the input layer in Figure 2. $a=[a1,a2,...am]^T$ is a N-dimensional vector in the associative space A. The value of the element of a is only 1 or 0. Several elements of a corresponds to a storage address in the input space are 1, that is, it is a local area in the associated space A.

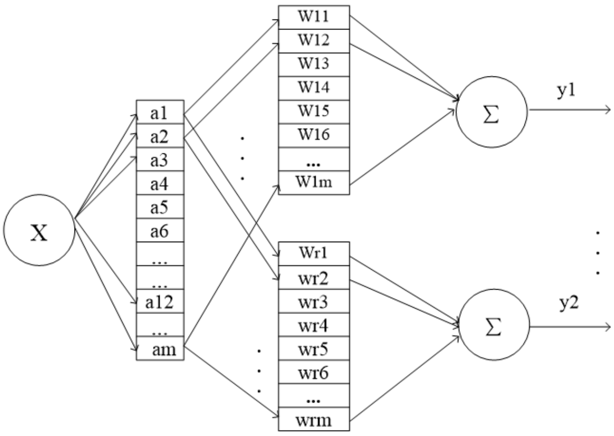

Figure 2: Structure of CMAC

2) P: A→y, that is y=p(a)=wa

The functions of the output layer can be seen from above formulas, where the connection weight is w_{ij}(i=1,2… m), it is an adjustable parameter. For the i-th output, there is the following formula.

$$y_i = P(a) = \sum_{j=1}^{m} w_{ij} a_j, \qquad i = 1, 2, ..., r \tag{2}$$

The learning algorithm of CMAC neural network is follows.

$$w_{ij}(k+1) = w_{ij}(k) + \beta(y_i^d - y_i)a_j / a^T a \tag{3}$$

Where, β is for the learning rate, y_i^d is for the expected value of the i-th output component and yi is for the actual value of the i-th output component.

3. Structure of CMAC-PID Parallel Controller

Due to the simple structure and convenient operation of PID controller, it has good control effect for linear constant control system, but the control effect is often not ideal for complex systems such as time-varying system and pure delay system, while CMAC has good robustness and adaptability, so it combines the advantages of PID controller and CMAC, the compound control of CMAC-PID is designed to achieve better control effect [9]. CMAC controller is used as feedforward controller to realize the operation of the inverse dynamic model of the controlled object, and common PID controller is used as feedback controller to ensure the stability of the system [19-20]. The structure of CMAC-PID composite control is shown in Figure 3.

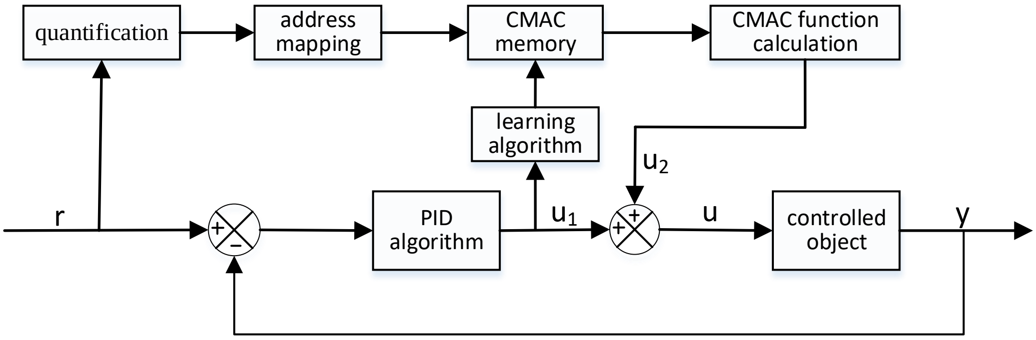

Figure 3: Structure of CMAC-PID controller

In Figure 3, r is the input signal, y is the output signal, $u_1(k)$ is the control variable of ordinary PID controller, and $u_2(k)$ is the control variable of CMAC controller, $u(k) = u_1(k) + u_2(k)$ .The supervised learning algorithm of CMAC neural network is given as follows.

$$u_2(k) = \sum_{m=1}^{c} w_m \beta_m \tag{4}$$

Where, when the input control S is within the quantization space, $\beta_m = 1$, otherwise, $\beta_m = 0$. C is the generalization parameter of CMAC neural network. At the end of a cycle, the corresponding output is calculated by the CMAC function, it is compared with the total control variable and the weight coefficient is modified, so that the objective function is the minimum. The objective function is follows.

The gradient descent method is adopted, and the weight is adjusted according to the following formula.

$$\Delta w(k) = -\eta \frac{\partial J(k)}{\partial w} = \eta \frac{u(k) - u_2(k)}{c} \beta_m = \eta \frac{u_1(k)}{c} \beta_m$$

$$w(k) = w(k-1) + \Delta w(k) + \alpha(w(k) - w(k-1)) \tag{5}$$

$$J(k) = \frac{1}{2c}\left(u_2(k) - u(k)\right)^2 \tag{6}$$

Where, η is for the network learning rate, $\eta \in (0,1)$, α is for the inertia, $\alpha \in (0,1)$

4. System Simulation of CMAC and PID Parallel Controller

The transfer function of the controlled object is follows.

$$G(s) = \frac{1200}{s^2 + 50s + 1200} \tag{7}$$

The parameters of CMAC neural network are for $N = 100$, $C = 5$, $\eta = 0.12$, $\alpha = 0.05$, PID parameters are for $k_p = 20$, $k_i = 0.01$, $k_d = 0.3$, sampling time is 1ms, and input signal is unit step signal. The system is simulated by MATLAB, and the effect comparison between CMAC-PID parallel controller and ordinary PID controller is shown in Figure 4.

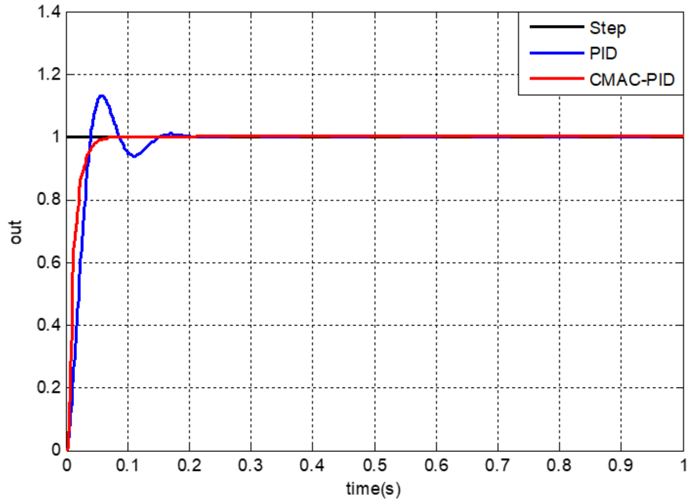

Figure 4: Comparison between CMAC-PID controller and traditional PID controller

It can be seen from Figure 4 that the CMAC-PID parallel control strategy is significantly better than the traditional PID control strategy in response speed and stability, and the control effect is significantly improved. The specific performance index comparison is shown in Table 1.

Table 1: Comparison of performance indicators between CMAC-PID controller and traditional PID controller

indicator / controller	Rise time(s)	Setting time(s)	Over shoot	Steady state error
traditional PID	0.04	0.15	15%	0
CMAC-PID	0.03	0.05	0%	0

From the comparison of performance indicators, the rise time is reduced by 0.01s if the CMAC-PID parallel control strategy is adopted, the setting time is reduced by 0.1s, and the overshoot is reduced by 15%, achieving no overshoot, and the steady-state error is 0. The CMAC-PID parallel control strategy is more ideal than the traditional PID control in both steady-state and dynamic performance.

In the CMAC-PID parallel control strategy, the control output of PID is u1, the control output of CMAC is u2, and the control output of CMAC-PID parallel controller is for u, the corresponding change curve are shown in Figure 5-7.

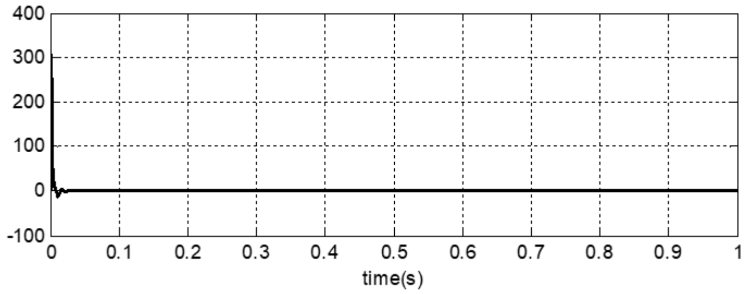

Figure 5: Control output of PID

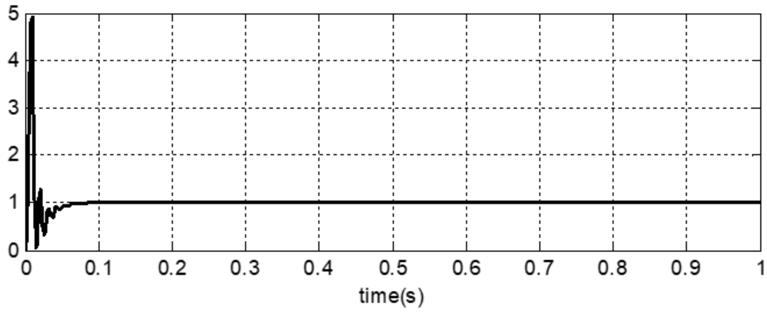

Figure 6: Control output of CMAC

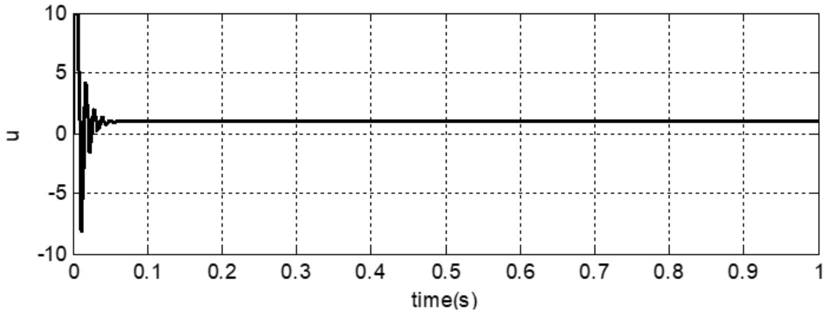

Figure 7: Control output of CMAC-PID

It can be seen from Figure 5-7 that the output of the controller tends to be stable within 0.05s, and the CMAC algorithm has relatively strong regulating effect, which can quickly adjust the parameters of the PID controller, so that the PID controller tends to be stable within 0.15s. When CMAC-PID parallel control strategy is adopted, the error curve of control system is shown in Figure 8.

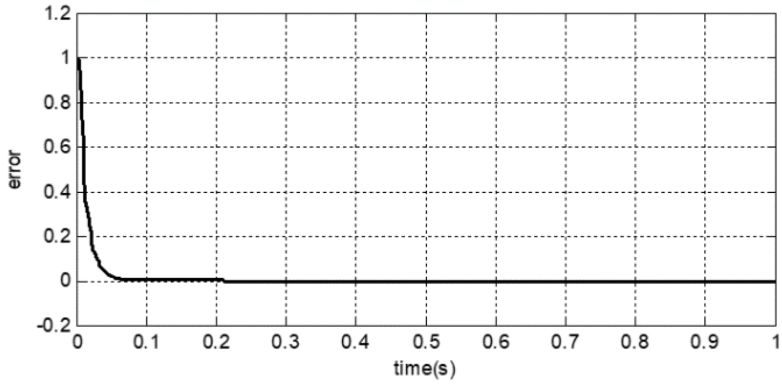

Figure 8: Error curve of control system

According to Figure 8, control system reaches a steady-state state without static error at 0.05s.In order to achieve a better human-machine interface, a GUI is written in the MATLAB environment. The GUI parameter is set of CMAC-PID parallel control strategy is shown in Figure 9.

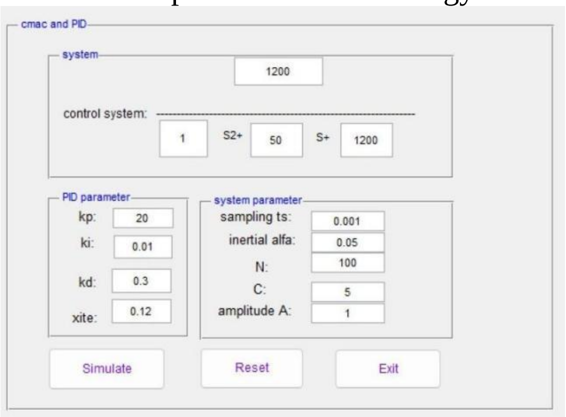

Figure 9: Interface of parameter setting

The initialization parameters of the system are set in Figure 5, click the simulate button to simulate, output of control system, output of controller and output of error can be obtained, which are as shown in Figure 10-12.

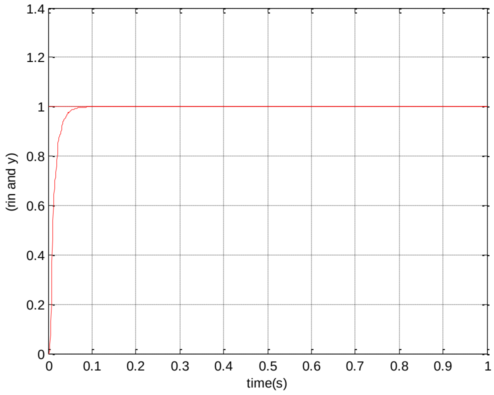

Figure 10: Output of control system

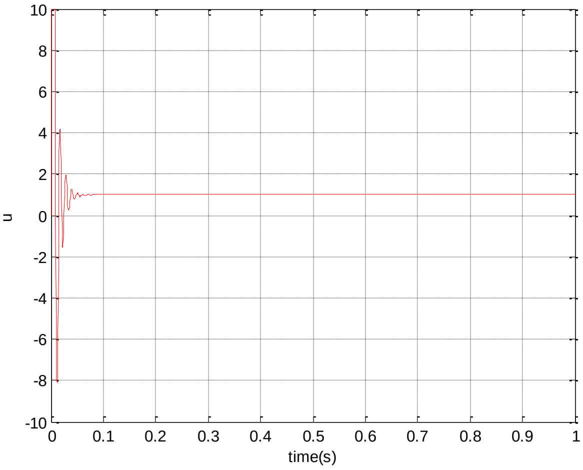

Figure 11: Output of controller

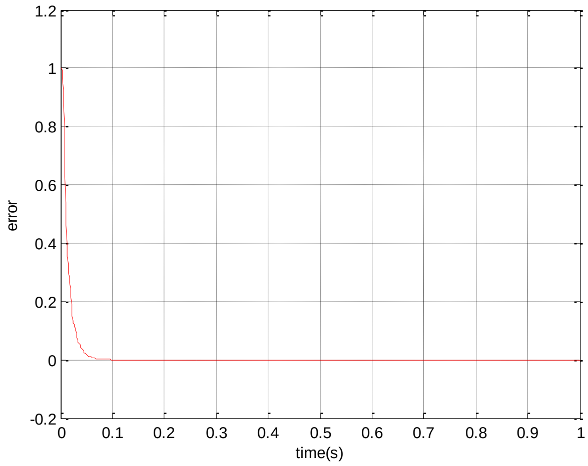

Figure 12: Output of error

According to Figure 9-12, the mathematical model parameters, PID parameters and parameters involved in the algorithm of control system are set in Figure 9, the corresponding curve of output signal, control signal and error curve of the control system are obtained, the control process of the control system can be better observed by these curves, and this system is suitable for the general second-order control system. The experimental results show that CMAC-PID parallel control strategy has better control effect in both steady-state performance index and dynamic performance index.

5. Conclusion

The principle and basic structure of CMAC neural network are firstly introduced, the advantages of CMAC neural network and common PID controller are analyzed, then CMAC-PID composite controller and its graphical user interface are designed, finally, the comparison curve between CMAC-PID composite controller and common PID controller is obtained by simulation. The simulation results show that CMAC-PID composite controller has strong robustness and adaptability.The well-designed graphical user interface also provides convenient conditions for the application of CMAC-PID composite controller.

Acknowledgements

This article was funded by the National Natural Science Foundation of China (Grant No. 62273061).

References

[1] Albus J S. Data storage in the control cerebellar model articulation control (CMAC). Trans. on Asme J. of Dynamic Systems Measurement &Control, 1975 (97): 228-233.
[2] H Ma, T Ying-Chih, LI Chen. A CMAC-based scheme for determining membership with classification of text strings. Neural Computing and Applications, 2016, 27 (7): 1-9.
[3] Rodr ģuez F O, Rubio J D J, Gaspar C R M. Hierarchical fuzzy CMAC control for nonlinear systems. Neural Computing & Applications, 2013, 23 (1): 323-331.
[4] Fu Jianjian, Yu Shixian. CMAC neural network and robust H ∞ composite control for flexible systems Mechanical Design and Manufacturing, 2020 (12): 158-166
[5] Zhao Jinsong, Wang Chunfa, Xu Jiaxiang, et al. CMAC-Fuzzy PID Control for Multi-dimensional Force Loading System of Hydraulic Drive Parallel Mechanism, Journal of Central South University (Natural Science Edition) 2020, 51 (10) :2811-2821.

[6] Liao Yuntao, Xiang Jie. Photovoltaic power short-term prediction based on CMAC-BP neural network, information technology and standardization 2020, (09): 54-58.
[7] Wang Ling. Intelligent optimization algorithm and its application Beijing: Tsinghua University Press, 2001
[8] Wang Xiaoping, Cao Liming. Genetic algorithm - theory, application and software implementation Xi'an: Xi'an Jiaotong University Press, 2002.
[9] Feng Dongqing, Xing Guangcheng, etc. Multivariable PID neural network control based on improved PSO algorithm Journal of System Simulation, 2011, 23 (2): 363-366
[10] Zhu Jiaqun, Zou Ling, Sun Yuqiang. Application research of CMAC neural network and PID compound control Microcomputer Information, 2006 (05S): 59-61
[11] Kraft L G, Campagna D P. A comparison between CMAC neural network control and two traditional adaptive control systems. Control Systems Magazine, IEEE, 1990, 10 (3): 36-43.
[12] Hu Jinzhu, Xie Shousheng, Zhai Xusheng. Engine guide vane control system based on CMAC and PID algorithm Journal of Missile and Guidance, 2009, 29 (4): 154-156
[13] Chen Hui, Zhou Ping. CMAC network structure of cerebellar model and determination of relevant parameters Computer Engineering, 2003, 29 (2): 252-254
[14] Li W, Zhang H, Xie F. CMAC and PID Concurrent Control for Aeroengine. Journal of Projectiles, Rockets, Missiles and Guidance, 2006 (3): 23-30.
[15] Yang Jie, Li Zehui, Ma Kai, Xu Chenglin. Optimal dispatching method of temperature control load based on cerebellar model neural network Power System Automation, 2022, 46 (10): 199-208
[16] Liu Dong. Research on brushless DC motor non-inductive control system based on fuzzy CMAC Master's Thesis of Harbin University of Technology, 2020
[17] Zhao Ganlin. Target trajectory prediction based on cerebellar model neural network Master's Thesis of Harbin Engineering University, 2018
[18] Fu Xingjian, Yu Shixian. CMAC neural network and robust H ∞ composite control for flexible systems Mechanical Design and Manufacturing, 2020, (12): 158-160+166
[19] T. Q. Ngo, T. V. Phuong. Robust adaptive self -organizing wavelet fuzzy CMAC tracking control for deicing robot manipulator. International Journal of Computers Communications & Control, 2015, 10 (4): 567-578.
[20] Rodr ǵuez F O, Rubio J D J, Gaspar C R M. Hierarchical fuzzy CMAC control for nonlinear systems. Neural Computing & Applications, 2013, 23 (1): 323-331.