A Fuzzy CMAC Neural Network Model Based on Credit Assignment

Daqi Zhu and Min Kong

Research Centre of Control Science and Engineering,
Southern Yangtze University,Wuxi, China, 214122

zdq367@yahoo.com.cn

Abstract

In order to improve online learning speed and accuracy of CMAC, a fuzzy CMAC neural network model based on credit assignment concept is designed. In the conventional CMAC and fuzzy CMAC learning scheme, the corrected amounts of errors are equally distributed into all addressed hypercubes, regardless of the credibility of those hypercubes values. The proposed improved learning approach is to use the learned times of the addressed hypercubes as the credibility (confidence) values of hypercubes learned, the corrected amounts of errors are proportion to the inversion of the learned times of the addressed hypercubes, With this idea, the learning speed demonstrated from examples can indeed become very fast.

Keyword: FCMAC, FCACMAC, Credit assignment, Nonlinear identification.

I. Introduction

During online identification of the nonlinear dynamic systems, the conventional reasoning method and Kalman algorithm can not meet the expected effect for their real time problem; the BP neural network[1-2],etc, for its low speed converge and lowest points existing in particular areas, can not only meet the real time requirement, but also has an inaccuracy in identification, and is not often used in online learning. Since the time when Albus proposed a cerebellar modal articulation controller CMAC[3-4], the CMAC was paid extensive attention for its advantages of fast converge speed and partial generalization ability[5-6]. Document [7] proved that the CMAC is able to approximate any nonlinear function in unlimited accuracy if the input space was quantized perfectly and the resolution is high enough, but the higher the quantized class, and the higher the storage expense and more complex calculating, which lead to lower converge speed; at the same time, because of the part receptive field, the storage is sparsely distributed, and the usual solution is to use HASHING technique which produces data collision and may also lower the approximate ability of CMAC. In view of this, documents [8-9] introduced fuzzy set to CMAC and proposed a fuzzy CMAC algorithm, document [10] produced a self-organizing hierarchical CMAC on the high dimension classification problems, which omits discreting, quantizing, coding, HASHING mapping and etc., improves the learning accuracy and speed, but the converge speed can not also meet the requirement of real time online application. Carefully analyzed, we can see that all the former models of improved CMAC is on the storage hypercube being addressed and output calculating stage; on the network weight adjusting stage, the corrected amounts of errors are equally distributed into all addressed hypercube, regardless of the credibility of those hypercube values. Such weight-refreshing algorithm violates the credit assignment concept, and the refreshed effect should be associated with the credibility of the addressed hypercube[11]. This paper introduced the credit assignment concept in document[12] into fuzzy CMAC (FCMAC) weight adjusting, proposed a credit assignment-based fuzzy CMAC (FCACMAC) learning algorithm, and the learning of the network is more rational and effective. The simulation result shows that FCACMAC has a relative better learning speed and accuracy.

II. Conventional CMAC and fuzzy CMAC model

A. *Conventional CMAC*

The basic concept of CMAC is to store the learning data (knowledge) into overlapped storage hypercube(remembering space), and its output is the addition of data in addressed hypercube. Two kinds of operating are included in the conventional CMAC, one is calculating the output result and the other is the adjustment of weight. The CMAC network. can be applied to approximate function $y = f(x)$, in which $x \in X \subset R^n, y \in Y \subset R^m$, and can be realized by mapping $X \to A \to Y$, A stands for N dimensioned storage space, $a \in A \subset R^N$ is the binary associate vector, and let the input x addressed $N_L(N_L < N)$ storage hypercube; mapping $A \to Y$ realized the weight sum of the storage hypercube:

$$y_i = \sum_{j=1}^{N_L} w_j a_j(x) \qquad i = 1,...,m \tag{1}$$

In (1), w_j is the weight of the jth storage hypercube, if $a_j(x)$ is addressed, then its values is 1, else is 0, and there are only N_L storage hypercube has an affection to the output.

The similar input addressed storage hypercube are overlapped, and will produce similar output. The mapping $X \to A$ will usually undergo discreting, quantizing, coding, HASHING mapping and etc., as the increasing of the input dimension and approximation accuracy, the quantizing grade will definitely increase, the complexity of the CMAC operation will jump sharply, and the real time learning characteristic will slow down. On the stage of weight adjusted of the network, the conventional CMAC equally distributed the corrected amounts of errors into all addressed hypercube, and to storage hypercube j which is addressed by a certain input x, its weight adjusting rule is:

$$w_j^k = w_j^{k-1} + \eta(\overline{y_s} - \sum_{j=1}^{N_L} w_j^{k-1} a_j(x)) / N_L \qquad j = 1,2,..., N_L \tag{2}$$

Where $\overline{y_s}$ is the desired value, η is the learning step, in the conventional CMAC, η is a constant, then the errors were equally distributed into N_L addressed hypercubes. We know that the weights of CMAC has included the former learned knowledge after $k-1$ times of iteration, however not each addressed hypercube has the same learning experience, which leads to the differences in the reliability of each addressed hypercube’ s weight and each addressed hypercube has different contribution to errors, that’s to say, no same credit assignment should be existed in N_L addressed hypercube. If these differences were ignored and all addressed hypercube gained the same amount of errors, then the errors produced by the state not been learned would cause "corruption" to the former learned information, and so in the network learning process, the desired data could only be gained after many times of learning.

For dynamic systems online identification, the real time characteristic is highly demanded, and under certain circumstance, the work should be finished in one circle or two, so the conventional CMAC can not meet the demand.

B. *Fuzzy CMAC neural network*

In order to improve the accuracy and real time character of CMAC learning algorithm, on the network output calculating stage, document [6] and [7] absorbed fuzzy self-organization competing algorithm to restructure the conventional CMAC neural network. The following definition are made here:

Definition 1: assume that the N_L hypercube addressed by a certain input x in CMAC can be recognized as a subspace Ψ_j which is centered at z_j and has a width of 2δ, we call Ψ_j as association field. For the conventional CMAC, if $a_j \in \Psi_j$, then $a_j = 1$, or else $a_j = 0$. The association fields are overlapped, which enables a certain generality ability of the network.

Definition 2: assume that input $x \in R^n$, association field Ψ_j $(j = 1,2,..., N_L)$ and is centered at z_j, the radium is δ, if each hypercube is manifested by a vector a_j which has the same dimension with the input, then the association index will be:

$$a_{fj} = \begin{cases} \dfrac{\delta - \|a_j - x\|}{\delta}, & \|a_j - x\| \le \delta \\ 0 & others \end{cases} \tag{3}$$

Base on the definition of association field, a fuzzed association vector $a_{fj}(x) = (a_{f1},..., a_{fN_L})^T$ will be gained, and the output of FCMAC will be:

$$y_i = \sum_{j=1}^{N_L} w_j a_{fj}(x) \qquad i = 1,2,..., m, \qquad j = 1,2,..., N_L \tag{4}$$

If $a_{fj} = 1, (j = 1,..., N_L)$, and $a_i = 0$ when other situation, then $a_{fj}(x)$ becomes binary vector $a_j(x)$, and we can see that the conventional CMAC is a particular situation of FCMAC. At the same time, as the absorption of association field, the discreting, quantizing, coding, HASHING mapping are unnecessary. The self-organizing competing algorithm is adopted to decide the area of association field, and the self-organizing partition can be realized.

Document [8] adopted the following algorithm to realize the weights adjusted of the network:

$$w_j^k = w_j^{k-1} + \eta(k)(\overline{y_s} - \sum_{j=1}^{N_L} a_{fj} w_j^{k-1}) a_{fj} / \sum_{j=1}^{N_L} a_{fj} \qquad j = 1,2,..., N_L \tag{5}$$

Here the updating of weights is in proportion to the hypercube' s association index a_{fj}, and the learning step η is still a constant, meaning that during the weight updating of fuzzy CMAC, errors are equally distributed into each addressed hypercube regardless of each addressed hypercube' s contribution ratio to the errors; that's also to say, for addressed hypercube of different updating times, after k times of learning, the reliability of their weights will be seen as the same. Such weight updating algorithm is the same with conventional CMAC, completely violates the definition of credit assignment, and leads to the bad results that those hypercube who should not be updated or update slightly needs constant learning and updating, while those who made great contribution to the errors and should undergo more times of updating, conversely decrease their updating times. In order to reach the predetermined approximation accuracy, the network has to learn many times, therefore the learning efficiency is decreased and the learning time is prolonged.

C. Credit assignment-based fuzzy CMAC(FCA-CMAC)

In order to improve the learning efficiency of CMAC and avoid the " correction" effect, the correcting errors should be distributed in accordance to the hypercube' s reliability. However, no effective methods have been developed to decide which hypercube should pay much more responsibilities to the current errors. In other words, no good methods have be produced to decide the reliability of the hypercube' s weight. The only available information is the current weight updating times of the hypercube. Document [12] assume that the more the hypercube updates, the more reliable the stored data are. So the learning times of the hypercube is seen as credit index. The more times the addressed hypercube have learned, the higher the credit index is, and the less the weight will be adjusted. The adjustment of the addressed hypercube is proportional to the inverse of their former learning times. Here we combine the concept of credit assignment with self-organizing competing algorithm of fuzzy CMAC, and proposed a credit assignment-based fuzzy CMAC learning algorithm, then (5) is changed into:

$$w_j^k = w_j^{k-1} + \frac{a_{fj}}{\sum_{j=1}^{N_L} a_{fj}} \left\{ \frac{(f(j)+1)^{-1}}{\sum_{l=1}^{N_L}(f(l)+1)^{-1}} \right\} (\overline{y_s} - \sum_{j=1}^{N_L} a_{fj} w_j^{k-1}) \tag{6}$$

In the equitation, $f(j)$ represents the former learning times of the j th hypercube, considered $f(j)=0$ at the initial learning stage and may cause the probability that denominator becomes zero, we use $(f(j)+1)$ to replace $f(j)$. N_L is the numbers of hypercube addressed by a certain state. The weight updating concept is that the correcting errors should be in converse proportion to the learning times of addressed hypercube. Here $(f(j)+1)^{-1} / \sum_{l=1}^{N_L}(f(l)+1)^{-1}$ is used to replace learning step constant η in (5), which effectively improves the learning property. Such weight updating concept is completely different from the conventional CMAC and fuzzy CMAC, and adjusts the weight according to the credit assignment. To be more specific, for hypercube which has less former learning times and greater contribution to the errors, their weight reliability is low, and current updating of weight is more; conversely, for those who has more former learning times and less contribution to the errors, their weight reliability is high, and the current updating of weight learning is less, thus the updating of weight is much more rational and effective, and under the condition of same approximation accuracy, the learning time decreases greatly and the real time characteristic of online learning is improved.

III. Simulation examples and results analysis

In order to illustrate the online learning effect of FCACMAC in further, take the following nonlinear functions as examples to study the learning effectiveness of conventional CMAC and fuzzy CMAC(FCMAC) as well as our proposed the FCACMAC.

A. One-dimensional nonlinear learning example

assume the following nonlinear function:

$$y(x) = \sin x + \cos x \qquad -\pi \le x \le \pi \tag{7}$$

in the learning process, the TAE(the total absolute error) and the RMSE(root mean square error) are adopt to show the modal's learning speed and accuracy.

$$TAE = \sum_{s=1}^{n} \left| (\overline{y_s} - y_s) \right| \tag{8}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{s=1}^{n} (\overline{y_s} - y_s)^2} \tag{9}$$

in the formula, n is the total state numbers, $\overline{y_s}$ is the desired output value of state s, y_s is the real output data of state s. For CMAC, FCMAC, FCACMAC, their learning results from the first circle to the twentieth circle are shown in figure 1 and figure 2, the error descent can be seen.

B. *Two-dimensional nonlinear learning example*

Assume the following nonlinear function:

$$y(x_1, x_2) = \sin(x_1 x_2) \qquad -1 \le x_1, x_2 \le 1 \tag{10}$$

in the network's learning process, TAE (The total absolute error) and RMSE(root mean square error) is the same with (8) and (9). for CMAC,FCMAC,FCACMAC, their learning results from the first circle to the twentieth circle are shown in figure 3 and figure 4, the error descent can be seen.

C. Discussion of the results

From figure 1 to 4, we can see that no matter one dimensional nonlinear target function or two, in the network's learning process, different CMAC models has great differences in the error descent speed, in which conventional CMAC has the lowest converge speed, and credit assignment-based fuzzy CMAC converge most quickly, while fuzzy CMAC stands between them.

At the same time we can see that the learning accuracy of FCACMAC is higher than conventional CMAC and FCMAC. As the dimension of the target function increases, the gaps between their learning effects are even more wide, that's because in low dimension condition, the calculation is smaller, and the error descent is quick for all kind of models, in twenty cycles a considerable approximation accuracy can be all reached; as the increase of the dimension, the calculation mounts increase sharply, and the rational characteristic of FCMAC and FCACMAC will embody more evident effect, which makes their gaps in learning speed and accuracy even more obvious, and adequately show the advantage of FCACMAC in online learning.

Obviously, the credit assignment-based fuzzy CMAC neural network learning algorithm proposed in this paper, for its combination of the FCMAC self-organizing algorithm and the credit assignment-based weight updating algorithm, not only in addresses the CMAC hypercube, omits the tricky operation such as discretizing, quantizing, coding, HASHING mapping and etc. on the results output stage, but also distributes the error according to the weight credit of each addressed hypercube on the updating of weight learning stage, which makes the updating of weight more rational and effective, greatly improves the learning speed and accuracy of CMAC neural network.

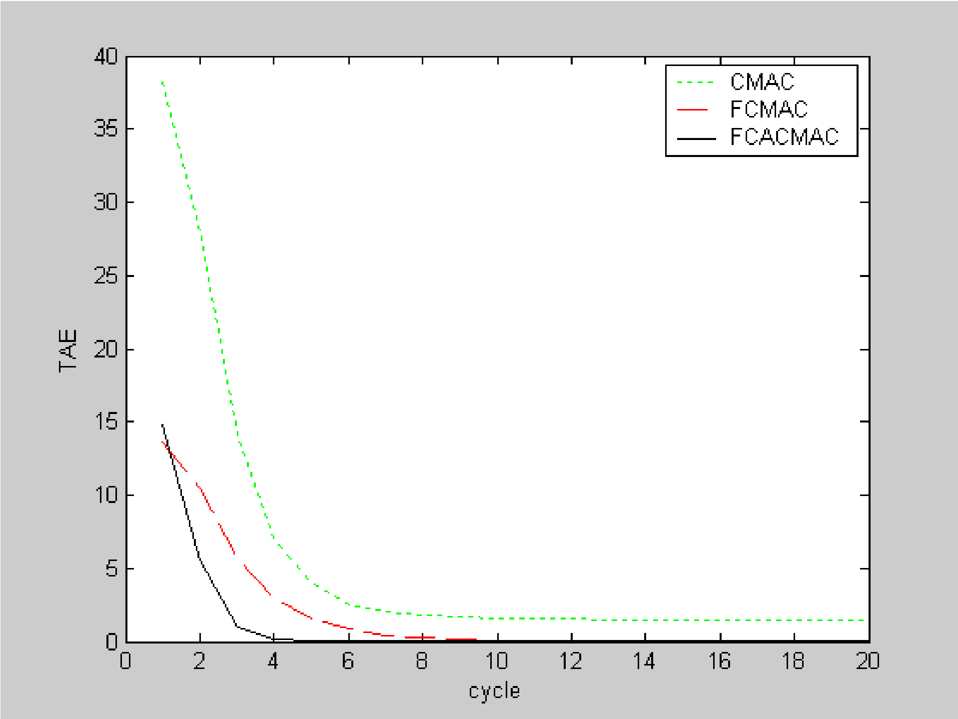

Figure 1 one dimensional TAE in CMAC, FCMAC and FCACMAC

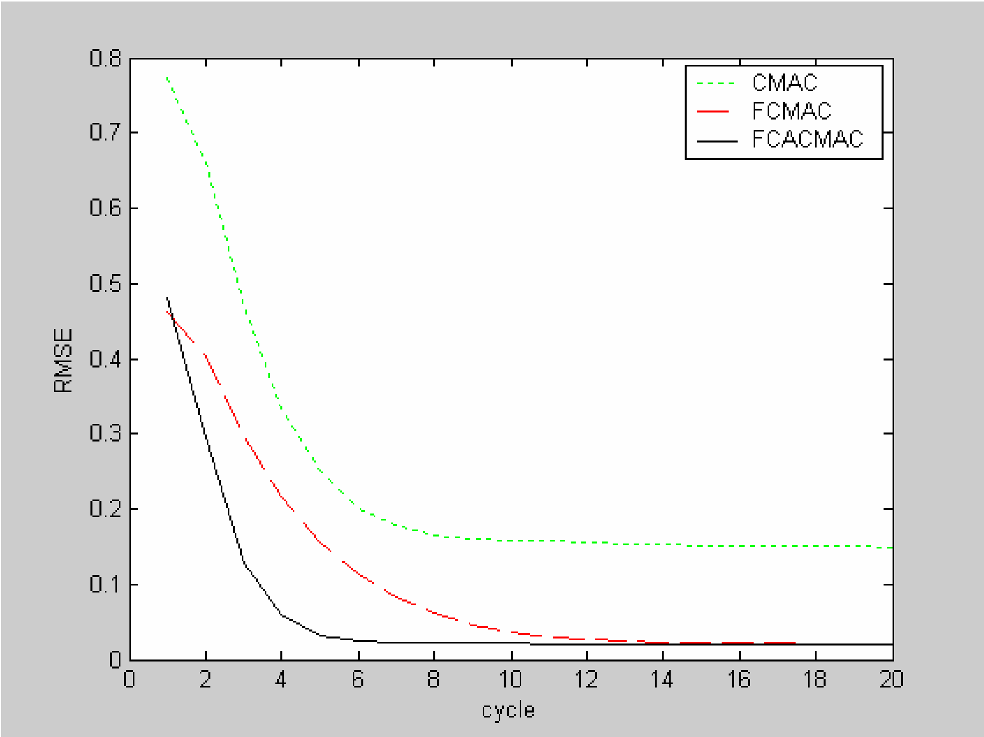

Figure 2 one dimensional $RMSE$ in CMAC, FCMAC and FCACMAC

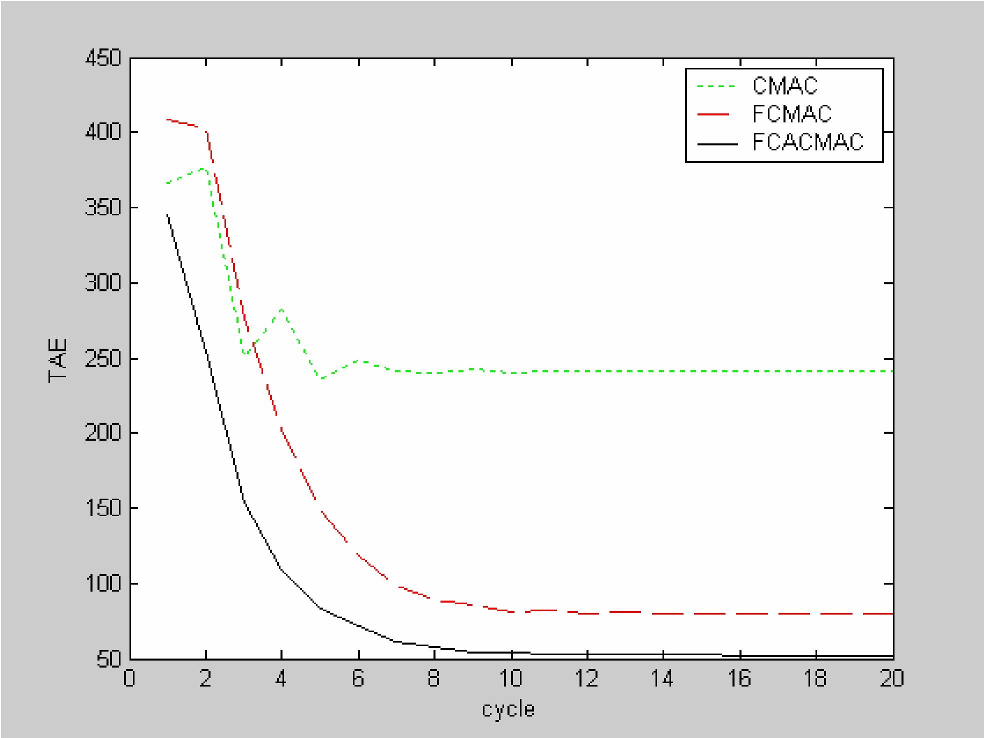

Figure 3 two-dimensional TAE in CMAC, FCMAC and FCACMAC

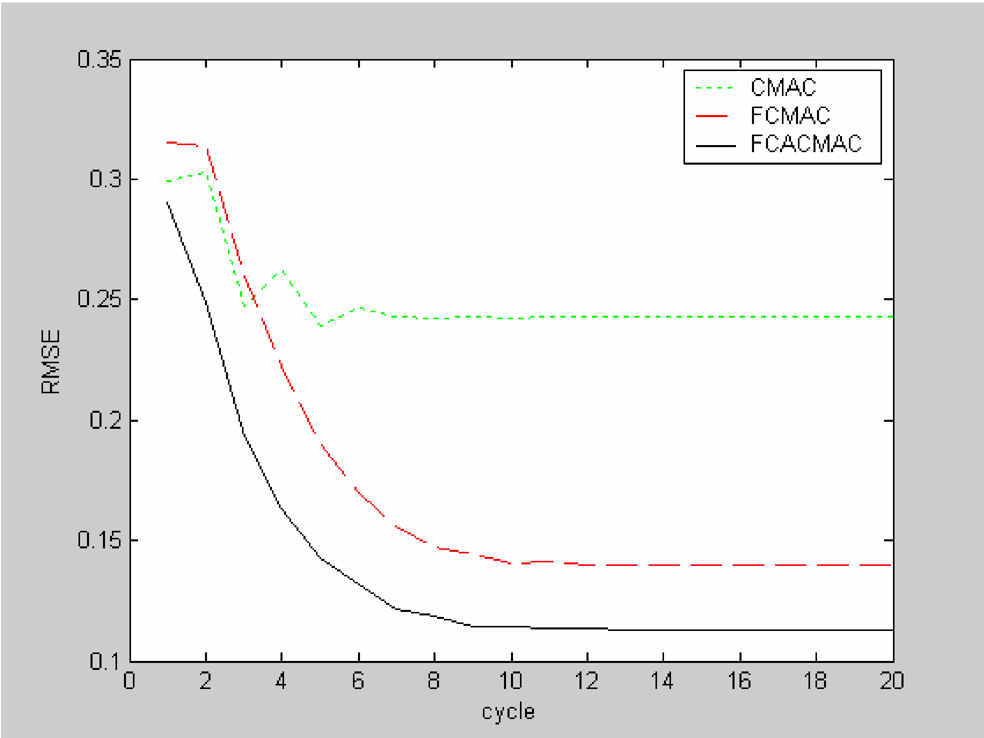

Figure 4 two-dimensional $RMSE$ in CMAC, FCMAC and FCACMAC

Ⅳ. Conclusion

In this paper, the FCMAC self-organizing competing algorithm and the credit assignment-based weight updating algorithm are combined, and a credit assignment-based fuzzy CMAC neural network learning algorithm is presented. Simulation results showed that compared with the conventional CMAC and FCMAC, FCACMAC improved evidently in both the learning speed and accuracy. For the dynamic nonlinear system identification, credit assignment-based fuzzy CMAC neural network modal is a better online learning tool.

References

[1] HuangD.S., Radial basis probabilistic neural networks, Model and application, International Journal of Recognition and Artificial Intelligence, Vol.13, No.7, 1999, pp.1083-1101.

[2] Huang D.S. The local minima free condition of feedforward neural networks for outer-supervised learning, IEEE Trans on SMC, Vol.28, No.3, 1998, pp.477-480.

[3] Albus J.S., A new approach to manipulator control: The cerebellar model articulation controller(CMAC), ASME J. Dynamic Systems, Measurement, Control, 1975, pp.220-227.

[4] Albus J.S., Data storage in cerebellar model articulation controller(CMAC), ASME J. Dynamic Systems, Measurement, Control, 1975, pp.228-233.

[5] Wong Y.F. and Sideris A., Learning convergence in cerebellar model articulation controller, IEEE Trans. Neural Networks, vol.3,No.1, 1992. pp. 115-121.

[6] Lin C.S. and Chiang C.T., Learning convergence of CMAC technique, IEEE Trans. Neural Networks, vol.8, no.6, 1997, pp. 1281-1292.

[7] Yiu-fai W., Athanasios S., Learning convergence in the cerebellar model articulation controller, IEEE Trans. On Neural Networks, Vol.5, No.6, 1992, pp.1393-1401.

[8] Nie J. and Linkens D.A., FCMAC: A fuzzified cerebellar model articulation controller with self-organizing capacity, Automatica, vol.30, no.4, 1994, pp. 655-664.

[9] Geng Z.J. and Mccullough C.L., Missile control using fuzzy cerebellar model arithmetic computer neural networks, J. Guid., Control, Dyn., vol. 20, no.3, 1997, pp. 557-565.

[10] Hana-Ming Lee., Chin-Ming Chen., A self-organizing HCMAC neural network classifier, IEEE Trans. On Neural Networks, Vol.14, No.1, 2003, pp.15-27.

[11] Huang D.S., Systematic Theory of neural networks for Pattern Recognition, Publishing House of Electronic Industry of China, Beijing, 1996.

[12] Shun-Feng S., Ted T., and Hung T.H., Credit assigned CMAC and its application to online learning robust controllers, IEEE Trans. On Systems, Man, and Cybernetics----Part B: Cybernetics, vol. 33, no.2, 2003, pp. 202-213.

Daqi Zhu was born in Anhui province, China, 1965. He received the B.S. degree in Physics, in 1992, from HuaZhong University of Science and Technology, and Ph.D. degree in electrical Engineering, in 2002, Nanjing University of Aeronautics and Astronautics. He is currently a professor in Research Centre of Control Science and Engineering Southern Yangtze University. His current research interests include neural networks and machine learning, Fault diagnosis .

Min Kong was born in Shandong province, China, 1983. He received the B.S. degree in electrical Engineering, in 2006, from Southern Yangtze University. She is currently a teacher in Research Centre of Control Science and Engineering Southern Yangtze University. Her current research interest is neural networks.