2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Study on Fuzzified CMAC Control for Ship Steering based on Eligibility

Zhipeng Shen and Chen Guo

Abstract—**A fuzzified CMAC controller with eligibility (FCE) is proposed. The introducing of eligibility can predict the uncertainty of controlled system, and improve the system stability. The structure of FCE system is presented, and its learning algorithm is deduced. To make the algorithm fit to on-line control ,the efficient implementation of FCE method is also given. Applying the FCE controller in a ship steering control system, the simulation results show that the ship course can be properly controlled in case of the disturbances of wave, wind, current. It is demonstrated that the proposed algorithm is a promising alternative to conventional autopilots.**

I. INTRODUCTION

CMAC is an acronym for Cerebellar Model Articulation Controller, which was first described by Albus in 1975 as a simple model of the cortex of the cerebellum, see [1-2]. Since then it has been extended in many different ways, and used in a wide range of different applications. Despite its biological relevance, the main reason for using the CMAC is that it operates very fast, which makes it suitable for real-time adaptive control. Eligibility is an idea described firstly by Croft ,and it has been used for many years as part of the reinforcement learning paradigm[3]. Many scholars are doing research on it, and achieve a lot beneficial result. Thereinto, Sutton had made systematic and future work on it in his doctoral thesis[4].

Combining eligibility into Fuzzy CMAC (FCMAC) neural network [5-8], a Fuzzy CMAC controller with Eligibility (FCE) is proposed in the paper. The basic idea is that each weight in the FCMAC neural network is given an associated value called its "eligibility". When the weight is used (i.e. when its value makes a contribution to the network output) the eligibility is increased. Thereafter it decays toward zero. Training should change all eligible weights by an amount proportional to their eligibility, i.e. the input error signal is only effective at integrating the weight while the eligibility is nonzero. Eligibility can improve the system in two ways. First, it can increase the system stability by achieving a "braking" effect, and reduce the oscillations. The second improvement is that cause and effect are better associated in the system, and the system state may be able to greatly anticipate the change.

Applying the FCE controller to ship steering control, the parameters of controller are on-line learned and adjusted. And the FCE algorithm will be formally derived by considering how to optimize an error function of the controlled system's inputs and outputs. To make the algorithm fit to on-line control ,the efficient implementation of FCE method is also given. Simulation results show that the ship course can be properly controlled in case of the disturbances of wave, wind, current. It is demonstrated that the proposed algorithm is a promising alternative to conventional autopilots.

II. WEIGHT ELIGIBILITY

Eligibility is an idea that has been used for many years as part of the reinforcement learning paradigm.The basic idea is that each weight in a neural network is given an associated value called its "eligibility". When the weight is used (i.e. when its value makes a contribution to the network output) the eligibility is increased. Thereafter it decays toward zero. Figure 1(a) shows a normal neural network in which a weight multiplies some input to get some output. The weight is usually the integrated value of some error signal, which is somehow derived from the global error signals given to the network. Figure 1(b) shows how the network is modified for eligibility. The input is filtered to obtain an eligibility signal ζ, which is multiplied by the error and the result integrates the weight. Many different forms of the eligibility filter have been proposed to solve specific problems but it is usually a low pass filter of some description.

Figure 2 shows how eligibility is implemented in a two-input FCMAC. In FCMAC all the network inputs will be either zero or not. All weights along the input trajectory will have some degree of eligibility. Training should change all eligible weights by an amount proportional to their eligibility ,i.e. the error signal is only effective at integrating the weight while the eligibility is nonzero.

Considering the principle of adding eligibility to the

Manuscript received September 20, 2004. This work was supported in part by the Specialized Research Fund for the Doctoral Program of Higher Education of P.R.C. under grant (20040151007), in part by the Ministry of Communications of P.R.C. under grant (200432922504)

Zhipeng Shen is with the School of Automation and Electrical Engineering, Dalian Maritime University. (e-mail: s_z_p@ 263.net).

Chen Guo is with the School of Automation and Electrical Engineering, Dalian Maritime University. (e-mail: guoc@ dlmu.edu.cn).

FCMAC weight, eligibility can improve the system in two ways. First, it can increase the system stability (reduce the oscillations) by increasing the region over which error acts, achieving a "braking" effect. Figure 3(a) shows the system state approaching and overshooting the reference. At any point the feedback-error signal influences a small area of the trajectory corresponding to the FCMAC local generalization region. Figure 3(b) shows that this area expands when eligibility is used, because weights influencing the older trajectory have remained eligible for modification by the error signal. The result is that on the next training iteration the braking force arising from error is able to be applied earlier, which is more appropriate for reducing the overshoot.

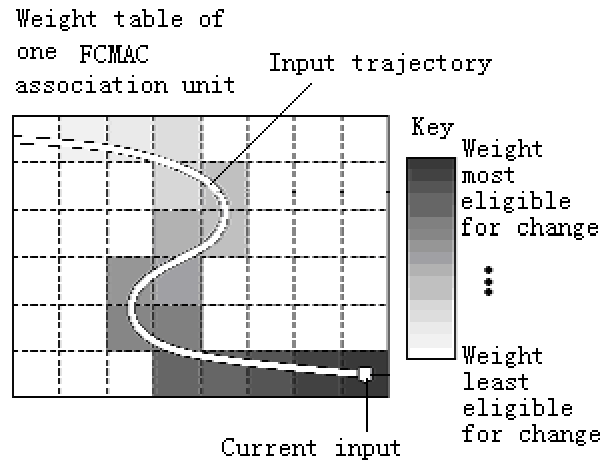

Fig.2 How eligibility is implemented in a two input FCMAC

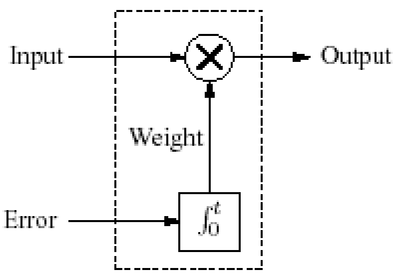

Fig.1 (a) A normal neural network

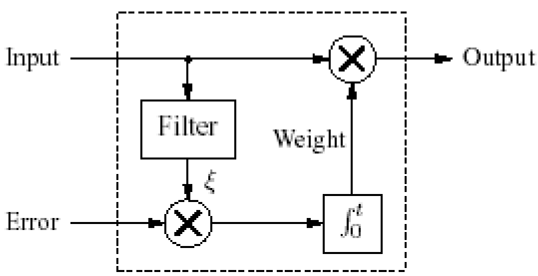

Fig.1 (b) A modified neural network which has weight eligibility

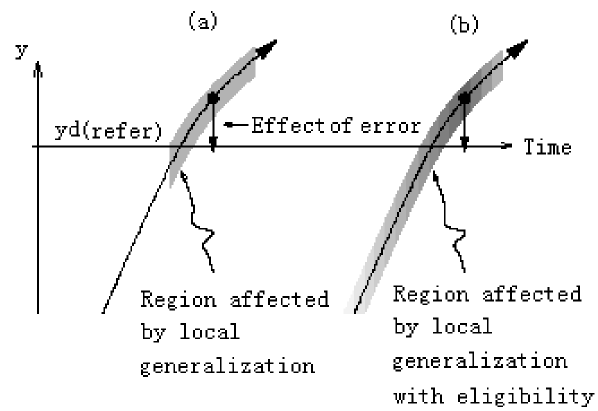

Fig.3 How adding eligibility to the standard FCMAC local generalization increases the region affected

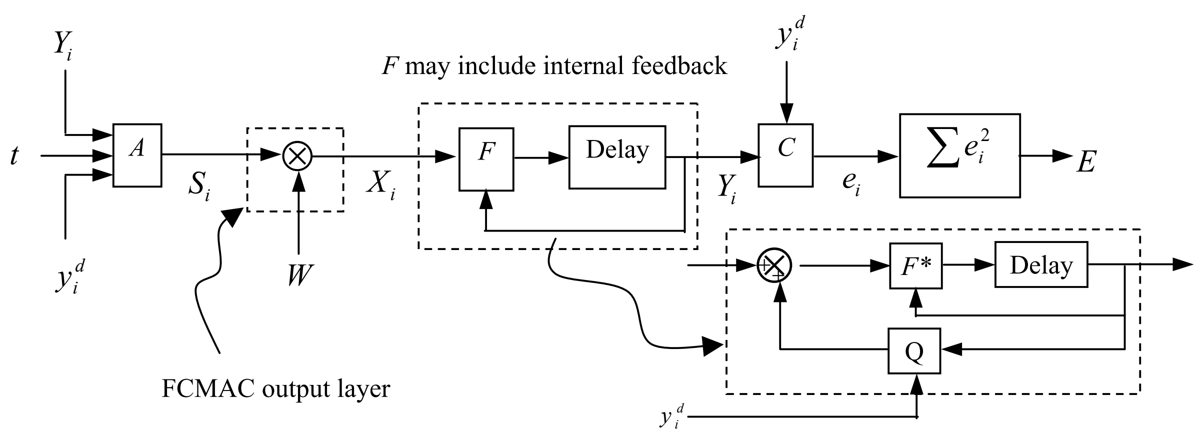

Fig.4 FCE controller and controlled system

III. FCE SYSTEM STRUCTURE

The basic FCE system shown in figure 4.The block F is the controlled system that implements:

$$Y_{i+1} = F(Y_i, X_i) \qquad (1)$$

As shown in the figure the block F can also incorporate any additional feedback controller Q that the basic system might have. The FCMAC has been split into two parts in the figure:

(1) The block A which represents the input layers (or association unit transformations). At each time step this block reads the system state Y_i (one expected value y_i^d or perhaps the current time t) and encodes it in the "sensor" vector S_i.

(2) The output layer which multiplies the vector S with the weight matrix W to get the output X ($X = SW$).At each time step the "critic" block C computes a scalar error value e_i that depends on the current state. The squares of these error values are summed over time to get the global error value E.

If the system's desired behavior is to track a scalar reference position y_i^d ,thus

$$e_i = y_i^d - CY_i \qquad (2)$$

Here C is a matrix that selects whatever element of y corresponds to a position. Then the total error E is given by:

$$E = \sum_{i=1}^{T} e_i^2 \qquad (3)$$

The critic function is chosen so that when E is minimized the system achieves some desired behavior.

IV. FCE LEARNING ALGORITHM

The purpose of the FCE learning algorithm is to modify W using gradient descent so that the error E is minimized. The gradient descent process for this system is:

$$w_j \leftarrow w_j + \alpha \frac{dE}{dw_j} \qquad (4)$$

Where w_j is an element of the matrix W and α is a scalar learning rate. This equation gives the modification made to weight w_j as a result of one iteration of learning. Now, from the chain rule:

$$\frac{dE}{dw_j} = \sum_{i=1}^{T-1} \left[2 \sum_{k=i+1}^{T} e_k \frac{de_k}{dX_i} \right] \cdot \frac{dX_i}{dw_j} \qquad (5)$$

Note that (T-1) is the limit on the outer sum, not T , because X_T has no effect on e_T. Here de_k / dX_i can be calculated using forward propagation ,i.e.:

$$\frac{de_{i+k}}{dX_i} = \frac{\partial e_{i+k}}{\partial Y_{i+k}} \cdot \left[\frac{\partial Y_{i+k}}{\partial Y_{i+k-1}} \cdots \frac{\partial Y_{i+2}}{\partial Y_{i+1}} \right] \cdot \frac{\partial Y_{i+1}}{\partial X_i} \qquad (6)$$

Now for a key step to determine the meaning of equation (6), and to derive a practical algorithm, F is approximated by a linear system F^*:

$$Y_{i+1} = AY_i + BX_i \qquad (7)$$

$$e_i = CY_i - y_i^d \qquad (8)$$

Combining equation (7) and (8) with equation (6):

$$\frac{de_{i+k}}{dX_i} = CA^{k-1}B \qquad (k > 0) \qquad (9)$$

thus

$$\frac{dE}{dw_j} = 2\sum_{k=2}^{T} e_k C \left[A^{k-i-1} B \hat{S}_i^j \right]$$

$$= 2\sum_{k=2}^{T} e_k C \xi_k^j \qquad (10)$$

where $\xi_k^j = \sum_{i=1}^{k-1} A^{k-i-1} B \hat{S}_i^j$, $\hat{S}_i^j = \frac{\partial X_i}{\partial w_j}$

$\hat{S}_i$ is all zero except for the element whose corresponding neural weight w_j is excited. And here ξ_k^j is called the *eligibility* signal. Based on the above equations, the FCE learning algorithm can be deduced:

$$\xi_1^j = 0 \qquad (11)$$

$$\xi_{i+1}^j = A\xi_i^j + B\hat{S}_i^j \qquad (12)$$

$$w_j \leftarrow w_j + \alpha \sum_{k=2}^{T} e_k C \xi_k^j \qquad (13)$$

Note that a factor of 2 has been combined into α .Every FCMAC weight w_j requires an associated eligibility vector ξ^j .The order of the eligibility model is the size of the matrix A. There is a relationship between the two constants α and C : if the magnitude of C is adjusted then α can be changed to compensate. Because of this the convention will be adopted that the magnitude of C is always set to one ($|C| = 1$) and then the resulting α is the main FCE learning rate.

V. .THE EFFECIENT IMPLEMETATION OF FCE LEARNING ALGORITHM

A naive implementation of the training equations is very simple—just update the eligibility state for every weight during each time step. Consider a FCMAC with n_w weights and n_a association units. To compute the FCMAC's output without training (in the conventional way) requires one set of computations per association unit, so the computation required is O(n_a) per time step. But if eligibilities must be updated as well then one set of computations per weight is needed, so the time rises by O(n_w). A typical FCMAC has n_w

>> n_a (e.g. n_a = 10 and n_w = 1000), so the naive approach usually requires too much computation to be practical in an online controller.

As follows, the FCE algorithm performs the same computations using a far more efficient approach which eliminates the recalculation of redundant information. FCE performs just O(n_a) extra operations per time step, so the total computation time is still O(n_a)—the same order as the basic CMAC algorithm.

The algorithm described below requires the system F^* to have an impulse response that eventually decays to zero. This is equivalent to requiring that the eigenvalues of A all have a magnitude less than one. This will be called the "decay-to-zero" assumption. The next simulation part will explain how to get around this requirement in the ship steering system.

The weights is divided into three categories according to their values:

(1) *Active weights*: where the weight is one of the n_a currently being accessed by the FCMAC. There are always n_a active weights.

(2)*Inactive weights*: where the weight was previously active and its eligibility has yet to decay to zero.

(3) *Retired weights*: where the weight's eligibility has decayed sufficiently close to zero, so no further weight change will be allowed to take place until this weight becomes active again.

Figure 5 shows how a weight makes the transition between these different states. FCE does not have to process the retired weights because their values do not change (their eligibilities are zero and will remain that way) and they do not affect the FCMAC output. An active weight turns in to an inactive weight when the weight is no longer being accessed by the FCMAC (transition 1 in figure 5). An inactive weight turns in to a retired weight after σ time steps have gone past (transition 3 in figure 5).The value of σ is chosen so that after σ time steps a decaying eligibility value is small enough to be set to zero. At each new time step a new set of weights are made active. Some of these would have been active on the previous time step, others are transferred from the inactive and retired states as necessary (transitions 2 and 4 respectively in figure 5).

If a weight w is made inactive at time i_1 and it becomes active or retired at time i_2, its value (and the corresponding eligibility value) must be corrected to account for the interval $i_1 \ldots i_2$ during which it was not modified.

If weight w_j goes inactive at time i_1 step then $\hat{S}_i^j = 0 \quad (i \ge i_i)$,so:

$$\xi_{i+1}^j = A\xi_i^j \qquad (i \ge i_i) \qquad (14)$$

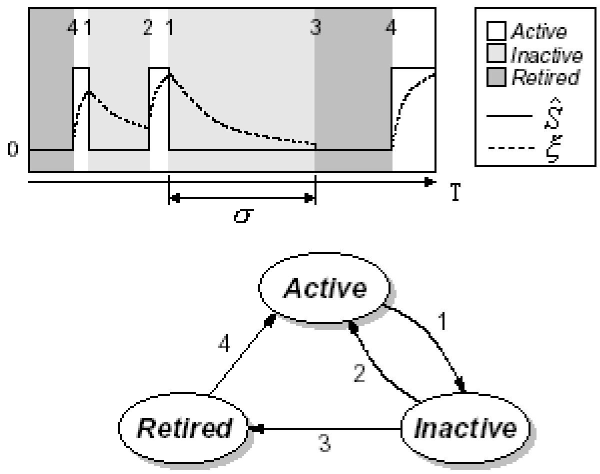

Fig. 5 The three states of FCE weight, and the transition between them

If w_j is made inactive (or "frozen") at time i_1, to find its true value at time i_2 the following must be computed:

$$w_j \leftarrow w_j + \alpha \sum_{i=i_1}^{i_2} e_i C \xi_i^j$$
$$= w_j + \alpha(\lambda_{i_2} - \lambda_{i_1-1}) A^{-i_1} \xi_{i_1}^j \qquad (15)$$

where

$$\lambda_k = \sum_{i=1}^{k} e_i C A^i \qquad (16)$$

Thus if the value λ is accumulated for each time step, any inactive weight can be activated or retired with a fast calculation (equation 15). Values of λ must be kept for at least the last σ time steps. Note that only one calculation of λ has to be made for the entire system, not one per weight.

There is one remaining complication: the exponentially decreasing A^i factor in equation 16 causes λ to quickly converge to a steady value. As time increases, equation (16) multiplies an exponentially increasing value (A^{-i_1}) by an exponentially decreasing difference $(\lambda_{i_2} - \lambda_{i_1-1})$. The result will quickly lose numerical precision.

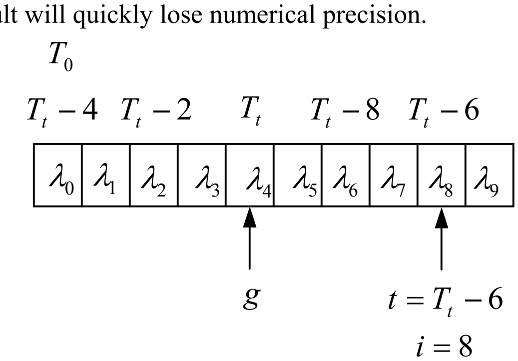

Fig.6 λ buffer (δ=10)

This can be rectified by periodically resetting λ to zero and then accounting for the discontinuities. This idea is implemented in the "trace" algorithms. The trace algorithm maintains a buffer of δ past λ values (where δ = σ + 2), as shown in figure 6. Here *g* is the index of the λ value for the "current" time T_t. Each step increases *g* and deposits a new λ value (*g* wraps around to zero at the end of the buffer). The buffer size limits the value of $T_t - t$ to at most δ-2, or σ. The procedure shows that the λ values are set as follows:

$$\lambda_i = \sum_{j=0}^{i} e_{T_0+j} C A^j \qquad (0 \le i \le g) \tag{17}$$

$$\lambda_i = \sum_{j=0}^{i} e_{T_0-\delta+j} C A^j \qquad (i > g) \tag{18}$$

VI. SIMULATION

Figure 7 shows the ship steering control system applying FCE controller. Its input are course error $\Delta\psi=\psi_r(k)-\psi(k)$ and fore turning angular velocity $r(k)$. Its output is the rudder angle $\delta(k)$. $\Delta\psi$ varies between (-20˚,20˚), r between (-0.9˚/sec,0.9˚/sec),and δ is (-35˚,35˚).

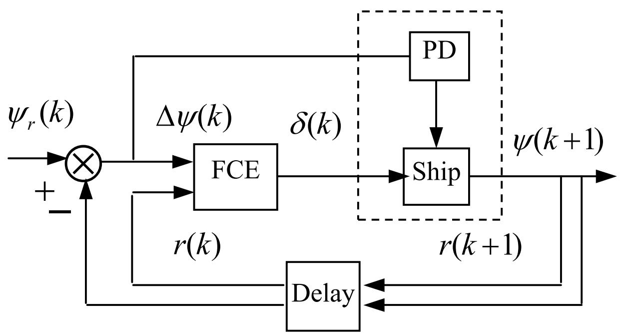

Fig 7 Ship steering control system

Ship motion can be described either in state space mode or by input-output model. The former can deal with multivariable problem of ship steering control and the disturbances caused by waves, wind and currents directly and more accurately, but the computation burden is more heavy. The latter is also called response model, it omits the sway velocity but grasps the main characteristics of ship dynamics: $\delta \rightarrow \dot{\psi} \rightarrow \psi$, and the obtained differential equation can still preserve the nonlinear. The disturbances of wind, waves can even be converted to a kind of equivalent disturbance rudder angle as an input signal. In fact, response model is an extension of the linear Nomoto model. The second-order Nomoto model is

$$\ddot{\psi} + \frac{1}{T}\dot{\psi} = \frac{K}{T}\delta \tag{19}$$

To some unstable ship, $\dot{\psi}/T$ must be replaced with a non-linear term and $H(\dot{\psi}) = a\dot{\psi} + \beta\dot{\psi}^3$ So the second-order non-linear ship response model is expressed

$$\ddot{\psi} + \frac{K}{T}H(\dot{\psi}) = \frac{K}{T}\delta \tag{20}$$

parameters a, β and K, T is related to ship's velocity.

The FCE algorithm described above requires the system F^* to have an impulse response that eventually decays to zero. So a PD feedback control element is joined, then the ship state model is changed:

$$X = AX + Bu \tag{21}$$

$$Y = CX \tag{22}$$

where

$$X = [\dot{\varphi}, \varphi]^T, A = \begin{bmatrix} -(1+\mathrm{Kkd})/\mathrm{T} & -\mathrm{Kkp/T} \\ 1 & 0 \end{bmatrix}$$

$$B = [K/T, 0]^T, C = [0,1], u = \delta$$

Transfer the state matrix into discrete format, the eligibility curve can be attained as fig.5 shown. Here, $K = 0.36, T = 230,\quad K_p = 1.2$, $K_d = 15$ .The eligibility decays to zero about 80s from fig.8, so the eligibility decay parameter can be selected as $\sigma = 100$.

Fig.9 shows the control curve result when set course is 10˚, wind force is Beaufort 5 and wind direction is 30˚.While Fig.10 and Fig 11 show the control curve result when set course is 5˚ ~15˚~20˚. From the compared curves, the proposed FCE control has better real-time quality and fast tracking speed. In term of course, it has no over-training results and has satisfied tracking effect; as to rudder angle, at beginning the bigger angle is accelerated to start up ,then regained to stable angle needed. The curves indicate that the course tracking is fast, control action reasonable and meet the performance of ship steering. The control result is partial satisfied.

VII. CONCLUSION

A fuzzified CMAC controller with eligibility (FCE) is proposed. The eligibility can predict the uncertainty of controlled system, and improve the system stability. The structure of FCE system is presented, and its learning algorithm is deduced. To make the algorithm fit to on-line control ,the efficient implementation of FCE method is also given. Applying the FCE controller in a ship steering control system, the simulation results show that the ship course can be properly controlled when changeable wind and wave exist. It is demonstrated that the proposed algorithm is a promising alternative to conventional autopilots.

References

[1] Albus,J.S. (1975). A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller(CMAC). *Trans ASME-J Dyn Syst Meas Contorl,* 97, ,pp. 220~227

[2] Albus,J.S. (1975). Data Storage in the Cerebellar Model Articulator Control (CMAC). *Trans ASME-J Dyn Syst Meas Contorl*, 97,. pp.228~233

[3] Rainer Malaka and Martin Hammer.(1996). Real-time models of classical conditioning. In Proceedings of the International Conference on Neural Networks (ICNN'96), Washington, *volume 2*, pp 768–773. IEEE Press, Piscataway, N.J.

[4] Sutton,R.S. (1984). Temporal credit assignment in reinforcement learning. PhD thesis, University of Massachusetts, Amherst,MA.

[5] Deng,Z.d., Sun,Z.Q.(1995) A Fuzzy CMAC Network. *,ACTA AUTOMATICA SINIC,* China, VOL.21, No.3,288~293.

[6] Nie,J.H. and D.A.Linkens. (1994). FCMAC: a Fuzzified Cerebellar Model Articulation controller with Self-organizing Capacity. *Automatica*, Vol.30,No.4, pp.655~664

[7] Lin,C.S. and Chris,K. (2002). A Fuzzy Cerebellar Model Approach for Synthesizing Multichannel Recordings. *AES 113TH Covention, LOS ANGELES ,* CA, USA, OCTOBER 5-8.

[8] Sun,Z.Q.,Deng,Z.D.(1996) A Similar CMAC Network and its Application in Control.,Journal of Tsinghua University China, VOL.36,No.5,17~23.

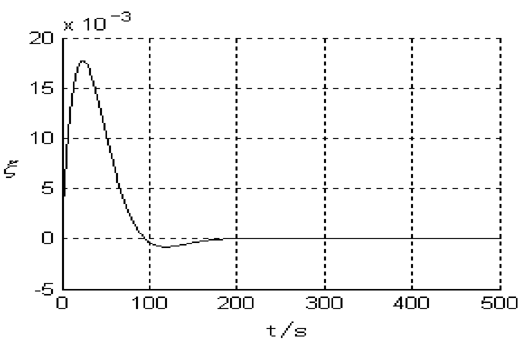

Fig.8 eligibility curve

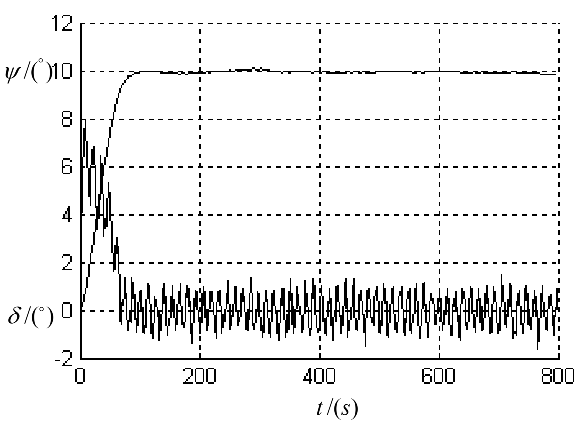

Fig.9 control curve, course 10°

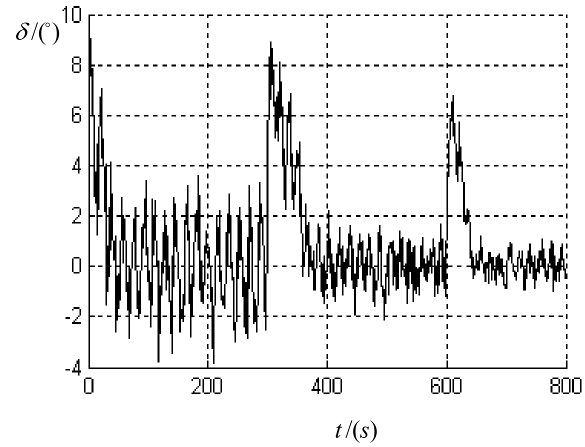

Fig.10 control curve, course 5° ~15°~20°

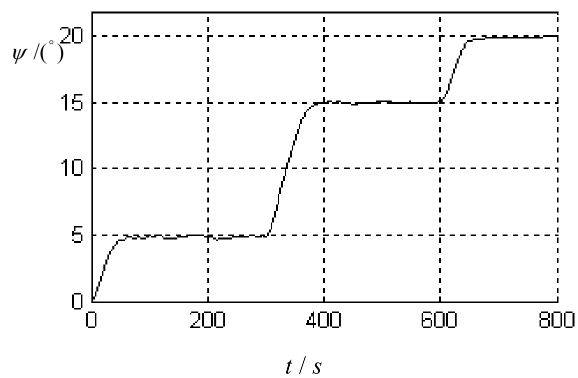

Fig.11 control curve, course 5° ~15°~20°