Fault-tolerant Control of Nonlinear System Using Credit Assign Fuzzy CMAC[1)]

ZHU Da-Qi KONG Min

(*Research Centre of Control Science and Engineering, Southern Yangtze University, Wuxi* 214122)

(E-mail: zdq367@yahoo.com.cn, zhizi0708@sohu.com)

Abstract The adaptive fault-tolerant control scheme of dynamic nonlinear system based on the credit assigned fuzzy CMAC neural network is presented. The proposed learning approach uses the learned times of addressed hypercubes as the credibility, the amounts of correcting errors are proportional to the inversion of the learned times of addressed hypercubes. With this idea, the learning speed can indeed be improved. Based on the improved CMAC learning approach and using the sliding control technique, the effective control law reconfiguration strategy is presented. The system stability and performance are analyzed under failure scenarios. The numerical simulation demonstrates the effectiveness of the improved CMAC algorithm and the proposed fault-tolerant controller.

Key words Credit assigned fuzzy CMAC, fault diagnosis, fault-tolerant control, nonlinear system

1 Introduction

During the last two decades, extensive research activities have focused on developing fault-tolerant control (FTC) to maintain the system stability and to avoid losses under various failure scenarios. Reference [1∼3] provided excellent overviews of recent research work on FTC. However, links between fault diagnosis and FTC techniques are still lacking[1], and some recent results on the integration of fault diagnosis with FTC can be found in [4∼7]. From these recent research results, two key components must be researched further: 1) the online fault diagnosis module consisting of the fault detection mechanism and online fault estimator; and 2) the controller module consisting of nominal controller and a fault-tolerant controller.

The fault information generated by diagnosis procedure can be very useful to FTC, and accuracy and speed of fault diagnosis are very important for active fault-tolerant control. With development of artificial neural network, it has strong superiority in fault diagnosis, especially for the cases that cannot be expressed by formula, and in complex nonlinear situation with strong accommodation. In various neural networks, the BP (back-propagation) using multi-layer feedforward neural networks (MFNN) is always the first candidate to deal with nonlinearity fault. However, owing to the gradient descent nature of BP, the learning process of an BP may need to iterate too many times so as to converge to an acceptable error level, or even diverge. So BP can hardly be used for online learning. Another kind of learning approaches termed as cerebellar model articulation controllers (CMAC) was proposed in [8,9], in which several advantages including local generalization and rapid learning convergence have been demonstrated. Although the conventional CMAC is much faster than BP, it still is not good enough for online learning. Several approaches have been proposed to improve the learning performance of CMAC in [10,11], they introduced the fuzzy concept into the cell structure of CMAC. As expected, the fuzzy kind of interpretation capability can indeed increase accuracy of representation of the stored knowledge. However, the speed of convergence still cannot be improved. Recently, Shun *et al.*[12] presented a credit assigned CMAC learning approach (CA-CMAC), and its on-line learning speed can be improved. However, for on-line fault-tolerant control of nonlinear dynamic systems, the convergence speed still cannot satisfy the requirement for real-time applications. This paper introduces the credit assignment concept into fuzzy CMAC (FCMAC) weight adjusting and proposes a credit assignment-based fuzzy CMAC (FCA-CMAC) learning algorithm, so that the learning of the network is more rational and effective. The simulation result shows that FCACMAC has a relative better learning speed and accuracy.

1) The Natural Science Foundation of Jiangsu Province (BK200402), Key Project of Chinese Ministry of Education (105088)

Received August 30, 2005; in revised form November 21, 2005

In this paper, fault diagnosis and fault-tolerant control are linked together after the unknown fault is estimated online by using the improved CMAC neural network (FCA-CMAC). In order to improve the system performance and stability, the nominal controller is reconfigured by the discrete-time sliding mode control law(DSMC) to compensate the fault effect. An integrated fault-tolerant control scheme of dynamic nonlinear system is designed.

2 Credit assignment-based fuzzy CMAC (FCA-CMAC)

2.1 Conventional CMAC

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

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

In (1), w_j is the weight of the jth storage hypercube. If $a_j(\boldsymbol{x})$ is addressed, then its values is 1, else is 0. There are only N_L storage hypercubes that have affection to the output.

On the stage of weight learning and adjusting of the network, the conventional CMAC equally distributes the amounts of correcting errors into all addressed hypercubes. For storage hypercube j which is addressed by a certain input $\boldsymbol{x}$, its weight adjusting rule is

$$w_j^k = w_j^{k-1} + \beta_1 (y_d - \sum_{j=1}^{N_L} w_j^{k-1} a_j(\boldsymbol{x}))/N_L, \quad j = 1, 2, \cdots, N_L \tag{2}$$

where y_d is the desired value, β_1 is the learning step, and is a constant in the conventional CMAC. Then the error is equally distributed into N_L addressed hypercubes.

2.2 Fuzzy CMAC neural network

In order to improve the accuracy and real time character of CMAC, on the network output calculating stage, [10] and [11] absorbed fuzzy self-organization competing algorithm to restructure the conventional CMAC neural network. The following definitions are made here.

Definition 1. Assume that the N_L hypercubes 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 the 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 to enable a certain generality ability of the network.

Definition 2. Assume that input $\boldsymbol{x} \in R^n$ and association field $\Psi_j (j = 1, 2, \cdots, N_L)$ is centered at z_j, with radius of δ. If each hypercube is manifested by a vector $\boldsymbol{a}_j$ which has the same dimension with the input, then the association index will be:

$$\boldsymbol{a}_{fj} = \begin{cases} \dfrac{\delta - \|\boldsymbol{a}_j - \boldsymbol{x}\|}{\delta}, & \|\boldsymbol{a}_j - \boldsymbol{x}\| \leqslant \delta \\ 0, & others \end{cases}$$

Base on the definition of the association field, a fuzzed association vector $\boldsymbol{a}_{fj}(\boldsymbol{x}) = (\boldsymbol{a}_{f1}, \cdots, \boldsymbol{a}_{fN_L})^{\mathrm{T}}$ will be gained, and the output of FCMAC will be

$$y_i = \sum_{j=1}^{N_L} w_j \boldsymbol{a}_{fj}(\boldsymbol{x}), \quad i = 1, 2, \cdots, m, \quad j = 1, 2, \cdots, N_L \tag{3}$$

If $a_{fj} = 1$, $j = 1, \cdots, N_L$, and $a_i = 0$ for other situation, then $\boldsymbol{a}_{fj}(\boldsymbol{x})$ becomes a binary vector $\boldsymbol{a}_j(\boldsymbol{x})$, and we can see that the conventional CMAC is a particular situation of FCMAC.

At the same time, because of the absorption of association field, the discreting, quantizing, coding, and 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. Reference [13] adopted the following algorithm to realize the weights for learning and adjusting of the network:

$$w_j^k = w_j^{k-1} + \beta_1 (y_d - \sum_{j=1}^{N_L} \boldsymbol{a}_{fj} w_j^{k-1}) \boldsymbol{a}_{fj} \Big/ \sum_{j=1}^{N_L} \boldsymbol{a}_{fj}, \quad j = 1, 2, \cdots, N_L \tag{4}$$

where the updating of weights learning is in proportion to the hypercube′s association index $\boldsymbol{a}_{fj}$, and the learning step β_1 is 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 is also to say, for addressed hypercubes of different updating times, after k times of learning, the reliability of their weights will be the same. Such a weight updating algorithm is the same as the conventional CMAC, completely violates the definition of credit assignment, and leads to a bad result that those hypercubes who should not be updated or updated slightly take constant learning and updating, while those who made great contribution to the errors and should undergo more times of updating cannot 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.

2.3 Credit assignment-based fuzzy CMAC (FCA-CMAC)

In order to improve the learning efficiency of CMAC, 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 be more responsible for 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. Reference [10] assumed that the more the hypercube updates, the more reliable the stored data. So the learning times of the hypercube is seen as a credit index. The more times the addressed hypercube has 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 propose a credit assignment-based fuzzy CMAC learning algorithm. Then (4) is changed into:

$$w_j^k = w_j^{k-1} + \beta_1 \frac{\boldsymbol{a}_{fj}}{\sum\limits_{j=1}^{N_L} \boldsymbol{a}_{fj}} \left\{ \frac{(f(j)+1)^{-1}}{\sum\limits_{i=1}^{N_L} (f(l)+1)^{-1}} \right\} (y_d - \sum_{j=1}^{N_L} \boldsymbol{a}_{fj} w_j^{k-1}) \tag{5}$$

In the equitation, $f(j)$ represents the former learning times of the jth hypercube. Considering that $f(j) = 0$ which is for the initial learning stage may cause the possibility that the denominator becomes zero, we use $(f(j)+1)$ to replace $f(j)$. N_L is the number of hypercubes 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. 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 hypercubes which have less former learning times and greater contribution to the errors, their weight reliability is low, and current updating of weight learning is more; conversely, for those who have 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 learning 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.

3 Fault-tolerant control law reconfiguration based on DSMC technique

3.1 The discrete-time sliding model control law

The discrete-time sliding model control law can be derived for a given discrete-time nonlinear dynamic system with unmatched uncertainties. A controllable nonlinear system with uncertainty can be represented as[14]:

$$y(k+1)=y(k)+\delta t\hat{f}(y(k))+\Delta t g(y(k))u(k)+\sum_{i=1}^{n}\beta_i(k-T_i)f_i(y(k)) \tag{6}$$

where $g(y(t))=\hat{g}(y(t))+\Delta g(\cdot)$, $\hat{f}(y(k))$, and $\hat{g}(y(k))$ are nonlinear functions of $y(k)$, Δg is uncertainty, Δt is the sampling period, $\beta_i(k-T_i)$ is a fault mode, T_i denotes the unknown fault-occurrence time, $f_i(\cdot)$ represents the dynamics of the failure mode i, and is assumed to be unknown due to the possible occurrence of unanticipated failures. The failure dynamics is an explicit function of past system outputs. Two typical faults, incipient faults and abrupt faults, are considered to be involved on-line. Abrupt failures represent the sudden change of the system dynamics due to catastrophic malfunction or failure of the system components and the incipient failures describe the time-varying effect of the system component-aging problem. We define $F(k)$ to represent the dynamics failure at time step k, and employ an online estimator $NF(k)$ to approximate the unknown failure dynamics $F(k)$ realized by

$$F(k)=NF(k)+\Delta NF(k) \tag{7}$$

$\Delta NF(k)$ can be treated as the remaining uncertainty of the failure dynamics and the on-line approximation error can be used to estimate the upper bound of $\Delta NF(k)$ in order to further improve the performance and increase the robustness property. Using the sliding mode control algorithm in [14,15], under different unknown failure modes, the effective control law $u(k)$ to accommodate the failures can be revised by adding a corrective control input $u_2(k)$, such that $u(k)=u_1(k)+u_2(k)$, where $u_1(k)$ represents the nominal control law:

$$u_1(k)=\frac{D(k)}{D(k)\hat{g}(y(k))}\left[-\hat{f}(y(k))+\frac{\Delta y_d}{\Delta t}\right]=\frac{1}{\hat{g}(y(k))}\left[-\hat{f}(y(k))+\frac{\Delta y_d}{\Delta t}\right] \tag{8}$$

The corrective sliding mode control law for the control problem is computed by

$$u_2(k)=\frac{K(k)}{D(k)g(\cdot)}sat\left(\frac{S(k)}{\phi(k)}\right)+U(k-T_c)\frac{-NF(\cdot)}{g(\cdot)} \tag{9}$$

where the first term on the right hand side of (9) is relevant to the online estimator error. The second term $U(k-T_c)\frac{-NF(\cdot)}{g(\cdot)}$ is the corrective control signal used to compensate for the nominal controller. $NF(\cdot)$ denotes the on-line estimator which tracks $\sum_{i=1}^{n}\beta_i(\cdot)f_i(\cdot)$. T_c denotes the specific time step at which the difference of the sum square approximation error of the on-line estimator during two consecutive windows, Ω is the parameter of on-line estimator, δ is below a pre-specified threshold, $\phi(k)$ is the boundary layer thickness, $K(k)$ is the controller gain.

$$\phi(k)=\Delta t[\eta(k)+\varepsilon],\quad K(k)=\eta(k)+2\varepsilon \tag{10}$$

where ε is an arbitrary positive number, $\eta(k)$ will be updated using the following equation.

$$\eta_{new}(k)=\begin{cases}\sup_L\{|D(k)\Delta NF(\cdot)|\}=\sup_L\left\{\left|D(k)(\sum_{i=1}^{n}\beta_i(\cdot)f_i(\cdot)-NF(k))\right|\right\}, & \Omega\leqslant\delta\\ \eta_{old}, & otherwise\end{cases} \tag{11}$$

$$S(k)=\frac{y_d(k)-y_d(k-1)}{\Delta t}-\frac{y(k)-y(k-1)}{\Delta t}+m(y_d(k)-y(k)),\ D(k)\approx\frac{S(k)-S(k-1)}{\tilde{y}(k)-\tilde{y}(k-1)}$$
$$\tilde{y}(k)=y_d(k)-y(k) \tag{12}$$

$y_d(k)$ is the desired system output at time step k. $\phi(k)$ is boundary layer thickness defined by the least upper bound of the remaining uncertainty, and identification error is defined as $\Delta f=\Delta NF(k)=\sum_{i=1}^{n}\beta_i(\cdot)f_i(\cdot)-NF(k)$ which is the remaining uncertainty of the failure dynamics. The design parameter

l represents a time period such that the least upper bound of the identification error is evaluated in every time period, $L = [k-l, k]$. (12) states that both the boundary layer thickness and the controller gain are automatically estimated and adjusted on-line by the estimator to further reduce control error. The on-line learning result is monitored and evaluated by the regulator using the following criteria.

$$SSAE0 = \sum_{k=k_0}^{k_0+l-1} (fy(k) - nfy(k))^2, \quad SSAE1 = \sum_{k=k_0+l}^{k_0+2l-1} (fy(k) - nfy(k))^2 \tag{13}$$

$$\Omega = |SSAE1 - SSAE0| \tag{14}$$

where $SSAE0$ and $SSAE1$ stand for the sum square approximation errors of the on-line estimator during two consecutive windows, $nfy(k)$ and $fy(k)$ are the output of the estimator and the difference between the measurement and the output of the nominal model at time step k, respectively. For certain threshold value, the on-line estimation result is considered to be accurate enough, and this implies that the on-line learning result cannot be further improved (*i.e.*, $\Omega \leqslant \Delta$).

3.2 Realization of fault-tolerant control

The intelligent on-line fault tolerant control scheme can be summarized as follows, with Step 1 being the off-line stage and Steps 2∼5 being the on-line stage:

Step 1. Obtain the nominal model, design the nominal controller, and test the performance of the controller with selected criteria (*e.g.*, mean square control errors, sum square errors). Determine the range within which the system is working under the nominal condition (*i.e.*, the fault detection threshold value) based on the testing results. A simple computationally cost-effective fault detection method used in the on-line simulation is as follows.

$$\psi = \frac{1}{\omega} \sum_{k=k_0}^{k_0+\omega-1} (y_d(k) - y(k))^2, \ \psi > \lambda, \quad failure\ alarm \tag{15}$$

Step 2. Set the initial upper bound η_0 for the unknown failure mode dynamics. Usually, the physical limitations of the system are useful information for deciding the upper bounds. For the alternative corrective control law, determine the threshold value δ for the convergence criterion of the on-line estimator and the design parameter l.

Step 3. Keep monitoring the system behavior after the system is working, and compare it with the nominal model response to decide if a fault has occurred. If the system is still in the nominal condition range, nothing has to be done. If a fault is detected, then initialize the on-line estimator to learn the failure mode dynamics by using the difference between the measured system output and the output of the nominal model as the desired target for on-line training.

Step 4. Add the corrective control signal (*i.e.*, $u(k) = u_1(k) + u_2(k)$). Evaluate Ω at every time period l. If $\Omega \leqslant \delta$, set T_c equal to the current time step k and adjust the upper bound of the remaining uncertainty.

Step 5. Collect the next training pattern from the measurement, keep on training the estimator, and observing the identification error of the estimator. Go back to Step 4 for the control process.

4 Simulation of the online fault diagnosis and fault-tolerant control

4.1 Nonlinear module

Consider the following SISO nominal plant.

$$y(k+1) = y(k) - \Delta t(a * y(k)^2 + b * y(k-1)) + \Delta t u(k) \tag{16}$$

where $y(k+1)$ and $u(k)$ represent the system output and control input at time step $k+1$ and k, respectively; Δt is the sampling period, a and b are the known system parameters. Under unexpected single failure mode, the system is represented by the following equation.

$$y(k+1) = y(k) - \Delta t(a * y(k)^2 + b * y(k-1)) + \Delta t u(k) + \beta(k-T) f(y(k), y(k-1)) \tag{17}$$

where $f(y(k), y(k-1)) = c \sin(y(k)) \times \cos(y(k-1))$ is unknown and c denotes an unknown constant gain. Abrupt fault scenarios will be considered, *i.e.*, $\beta(k-T) = U(k-T) = 1$, $\Delta t = 0.01$, $a = 5$,

$b = 100$, $c = 0.5$, $T = 100$, and the learning step $\beta_1 = 0.05$, $c = 0.5$. In order to track the desired trajectory $y_d(k+1)$, the nominal control input $u_1(k)$ is chosen by (16) as

$$u_1(k) = \frac{1}{\Delta t}\{-y(k) + \Delta t[a * y(k)^2 + b * (y(k-1))] + y_d(k+1)\} \tag{18}$$

The reference input: $r(k) = 0.2\sin\left(\frac{k\pi}{20}\right)$, desired output: $y_d(k+1) = 0.6y_d(k) + 0.2y_d(k-1) + r(k)$, and the sliding surface S is selected as

$$S(k) = \frac{y_d(k) - y_d(k-1)}{\Delta t} - \frac{y(k) - y(k-1)}{\Delta t} + 10(y_d(k) - y(k)) \tag{19}$$

The design parameters of the proposed intelligent control scheme are selected as follows: $\omega = 5$, $\lambda = 3 * 10^{-5}$, $\eta_0 = 0.5$, $\delta = 0.001$, and $l = 10$.

4.2 FTC results analysis

Fig. 1 shows the actual system output and the desired output within the 500 simulation time steps, when the nominal controller is applied alone at abrupt fault case. As is seen, the system performance degrades and a large deviation from the desired trajectory starts after the 100 time steps when the system suddenly experiences an abrupt fault. Fig. 2. shows the system response when the proposed intelligent control scheme is applied with the alternative corrective control law. As is clearly shown, the system performance is greatly improved. The controller successfully drives the output of the unknown faulty system back to the desired trajectory with a small range of error bounded by the estimated uncertainty. The fault is actually detected by the control regulator at time step 116. The control signal is adjusted by adding the corrective control signal, $u_2(k)$, and the FCA-CMAC neural network is initialized and learns the unknown failure mode dynamics on-line. At time step 127, the on-line identification error converges.

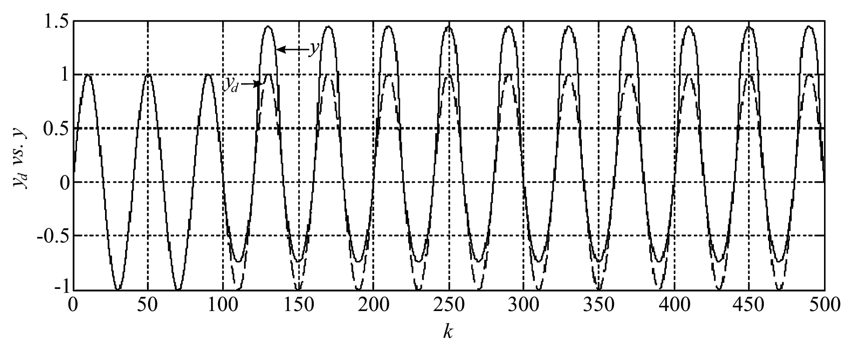

Fig. 1 The system response *vs.* desired output with nominal controller only

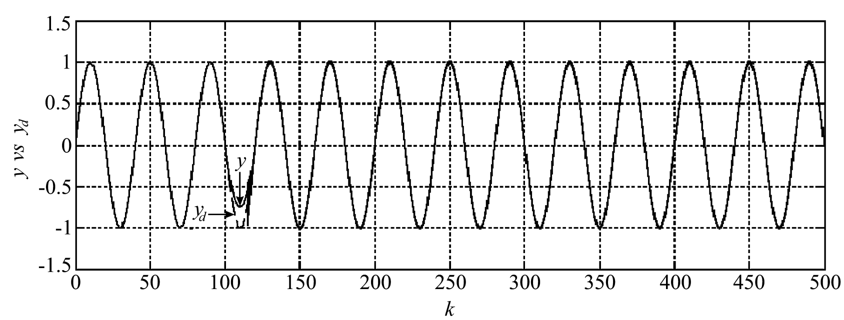

Fig. 2 The system response *vs.* desired output with the alternative corrective control law

Fig. 3. is the on-line estimator error plot. Note that there is no identification error defined before the 116th step because the learning of the neural network is initialized right after a fault is detected.

After step 127, the on-line estimator error becomes very little because the FCA-CMAC neural network has been converged. Fig. 4 shows the actual S function and the estimated boundary layer thickness. As is shown, the fault-tolerant controller adjusts the boundary layer thickness on-line. It is shown that the system is stable and that the fault-tolerant controller guarantees the boundary layer to be attractive. Because of the sliding mode control signal, the S function is confined within the boundary layer, as shown in Fig. 4.

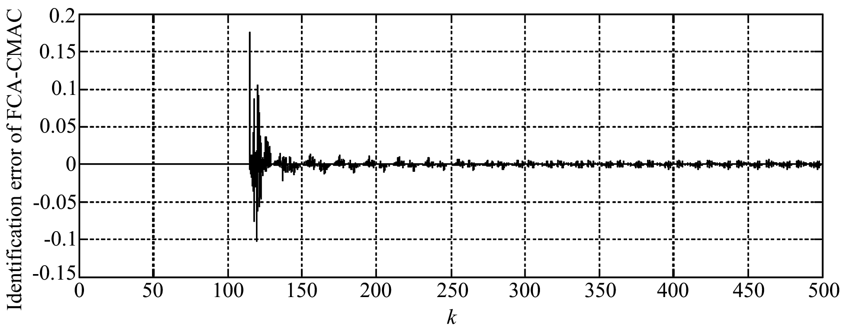

Fig. 3 The FCA-CMAC model identification error with the alternative corrective control law

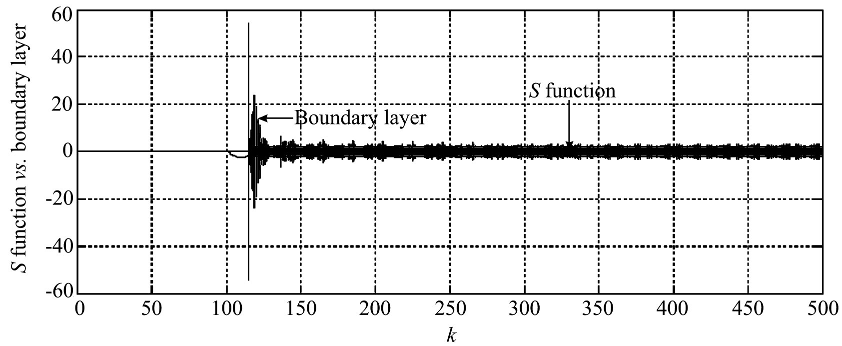

Fig. 4 The S function and the estimated boundary layer thickness

Fig. 5 shows the actual control signal at each time step. Before step 116, The actual control signal is the nominal control input, in the other time $u(k) = u_1(k) + u_2(k)$. The actual control signal is not relatively stable from step 116 to step 127, because the fault estimator FCA-CMAC has not been converged. After step 127, the fault estimator FCA-CMAC has been converged and $u(k)$ is stable.

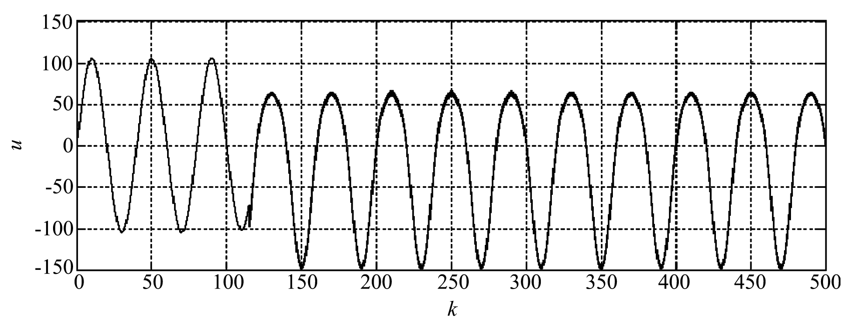

Fig. 5 The control input with the alternative corrective control law

5 Conclusion

In this paper, an efficient on-line fault diagnosis scheme and effective control law reconfiguration strategy are presented. The FCA-CMAC neural network is used in this research work as the on-line estimator to approximate the unknown failure dynamics. Based on the discrete-time sliding model control algorithm, the necessary and sufficient conditions to guarantee the system on-line stability and performance under various failure scenarios are greatly improved. The fault-tolerant controller successfully drives system back to the desired trajectory.

References

1 Patton R J. Fault-tolerant control: The 1997 situation (survey). In: Proceedings of International Federation of Automatic Control (IFAC) Symposium. Fault detection, supervision, and safety for technical processes, Hull, Oxford, England: Elsevier Science Ltd., 1997. 1029∼1052

2 Isermann R, Schwarz R, Stolzl S. Fault-tolerant drive-by-wire systems-concepts and realization. In: Proceeding of International Federation of Automatic Control (IFAC) Symposium. Fault detection, supervision, and safety for technical processes: Safeprocess, Budapest, Hungary: Elsevier Science Ltd., 2000. 1∼15

3 Zhou D H, Ye Y Z. Modern Fault Diagnosis and Fault-tolerant Control Technology. Beijing: Tsinghua University Press, 2000, 1∼23

4 Zhou D H, Frank P M. Strong tracking filtering of nonlinear time-varying stochastic systems with colored noise: application to parameter estimation and empirical robustness analysis. *International Journal of Control*, 1996, **65**(2): 295∼307

5 Chen S H, Tao G, Joshi S M. On matching conditions for adaptive state tracking control of systems with actuator failures. *IEEE Transactions on Automatic Control*, 2002, **47**(5): 473∼478

6 Chen S H, Tao G, Joshi S M. Adaptive control scheme for systems with unknown actuator failures. *Automatica*, 2002, **38**(8): 1027∼1034

7 Wang H, Huang Z J, Daley S. On the use of adaptive updating rules for actuator and sensor fault diagnosis. *Automatica*, 1997, **33**(2): 217∼225

8 Albus J S. A new approach to manipulator control: The cerebellar model articulation controller (CMAC). *Journal of Dynamic Systems, Measurement, and Control*, 1975, **97**(2): 220∼227

9 Albus J S, Albus J S. Data storage in cerebellar model articulation controller (CMAC). *Journal of Dynamic Systems, Measurement, and Control*, 1975, **97**(2): 228∼233

10 Nie J, Linkens D A. FCMAC: A fuzzy cerebellar model articulation controller with self-organizing capacity. *Automatica*, 1994, **30**(4): 655∼664

11 Geng Z J, Mccullough C L. Missile control using fuzzy cerebellar model arithmetic computer neural networks. *Journal of Guidance, Control, Dynamics*, 1997, **20**(3): 557∼565

12 Shun-Feng S, Ted T, Hung T H. Credit assigned CMAC and its application to online learning robust controllers. *IEEE Transactions on Systems, Man, and Cybernetics – part B: Cybernetics*, 2003, **33**(2): 202∼213

13 Shou S H, Yuan W. Robustness adaptive tracking control of nonlinear system based on self-organizing fuzzy CMAC neural network. *Acta Automatica Sinica*, 2002, **28**(6): 984∼989

14 Misawa E A. Discrete-time sliding mode control for nonlinear systems with unmatched uncertainties and uncertain control vector. *Journal of Dynamic Systems, Measurement, and Control*, 1997, **119**(4): 503∼511

15 Drakunov S, Utkin V. On discrete-time sliding modes. In: Proceeding of International Federation of Automatic Control (IFAC) Symposium on Nonlinear Control System Design, Oxford, England: Elsevier Science Ltd., 1988. 484∼489

ZHU Da-Qi Received the bachelor degree in physics from Huazhong University of Science and Technology in 1992, and Ph. D. degree in electrical engineering from Nanjing University of Aeronautics and Astronautics in 2002. He is currently a professor in Research Center of Control Science and Engineering at Southern Yangtze University. His research interests include neural networks, fault diagnosis, and fault-tolerant control.

KONG Min Graduate student in Research Center of Control Science and Engineering at Southern Yangtze University. Her research interests include neural networks, fault diagnosis, and fault-tolerat control.