AN ADAPTIVE FUZZY CONTROLLER WITH CMAC-BASED SCALING FACTORS

Liu Chaoying [*], Youichi Hirashima, Akira Inoue

Faculty of Engineering, Okayama Univ., 3-1-1 Tsushima-Naka, Okayama, 700-0082, Japan
[*] *Hebei Univ. of Science and Technology, Shijiazhuang, 050018, China*

Abstract: In this paper, a new adaptive fuzzy controller with CMAC-based adaptive scaling factors is proposed. By using the proposed method, scaling factors can be tuned on-line, and control performance can be improved. In addition, CMAC is used only for tuning scaling factors, so that it is easy to implement and other fuzzy parameters can be designed by conventional method. Furthermore, robustness for modeling error can be improved as compared to conventional method. Finally, simulation results are shown to demonstrate that the proposed method has better dynamic and static property and has better robustness than conventional method. *Copyright © 2002 IFAC*

Keywords: adaptive fuzzy control, fuzzy hybrid system, CMAC, fuzzy parameter tuning, on-line learning

1. INTRODUCTION

Since the fuzzy control technique was first introduced in the early 1970's, it has been paid more and more attention and has been used to control a wide range of poorly understood plants. Their success was attributed to the fact that inherently nonlinear control strategies, expressed in a (restricted) natural language framework, could be obtained from human operator and then implemented as a fuzzy controller. One of its main merits is that it is very useful when the process models are nonlinear or too complex for analysis and synthesis by conventional control techniques (Knadelb and Langholz, 1994). However, it is also worth noting problem that it is impossible to design a fuzzy controller that need not assume anything about its environment. One can only strive to lessen its dependence and sensitivity to the parameters of its environment. The main weakness of conventional fuzzy systems is that the fuzzy algorithms and parameters are provided by experts, and they cannot be tuned in the process. Thus, many new types of fuzzy controllers have been developed by researchers. Adaptive fuzzy controller is a type of the new fuzzy controllers.

The first adaptive fuzzy controller was developed by Procyk and Mamdani at Queen Mary College (Procyk and Mamdani, 1979). Now, many types of adaptive fuzzy controllers have been developed to satisfy the difference process requirements. Adaptive fuzzy controller can be categorized into two kinds: the one has adaptive fuzzy rules; the other has adaptive fuzzy parameters. In the fuzzy rule adaptive controllers, the number of control rules is increased, decreased or their shapes are modified according to change of plant parameters. In the fuzzy parameter adaptive controller, fuzzy parameter, for example scaling factor, are modified according to change of plant parameters. In general, the fuzzy parameters adaptive controllers are simpler than the fuzzy rule adaptive controllers are both in the application and the design (Brown and Harris, 1994).

In this paper, a new fuzzy controller with adaptive-scaling factors is proposed. This fuzzy controller can tune scaling factors on-line by Cerebellar Model Articulation Controller (CMAC). Although tuning scaling factor is simplest in all fuzzy parameters, it has similar effects as the change of mapping relationship to the control input and output variables, and the change of shape and size for membership function (Brown and Harris, 1994). Furthermore, CMAC can tune scaling parameters quickly and control performance can be improved, because of its learning capability of non-linear functions, small computational complexities of the learning algorithm and generalization capability. Therefore, the proposed method is also effective for plants existing modeling errors.

This paper is organized as follows. In section 2, scaling factors in fuzzy controller that effect to system performance are introduced briefly. In section 3, a new scheme of adaptive fuzzy controller is proposed. CMAC network, its learning algorithm and the principle of adaptive fuzzy control with CMAC-based scaling factor are given. In section 4, two-order plant is considered as including varying parameters, and computer simulations are conducted to show effectiveness of the proposed method. In section 5, properties of the proposed method are summarized.

2. THE RELATION TO SCALING FACTORS AND SYSTEM PERFORMANCE

The characteristics of fuzzy controller depend on fuzzy rule base, membership function, scaling factors. After rule base are determined by the experience of experts or operators, in fuzzy controller, parameters that can be tuned mainly are membership functions, scaling factors and scaling gains.

Scaling factors are used to map the real input to the normalized input space. In general, scaling factor is defined as (Brown and Harris 1994)

$$K_e=n_e/e_{max}\ , \qquad (1)$$
$$K_{ec}=n_{ec}/ec_{max}\ , \qquad (2)$$

where K_e is the scaling factor of the error, K_{ec} is the scaling factor of the error change rate, n_e n_{ec} are the numbers of scaling grades of error and error change rate respectively, e_{max}, ec_{max} are the maximum range of the real error and error change rate respectively.
K_e , K_{ec} are multiplied by the corresponding input variable, thus altering the domain of interest for the respective variable. After fuzzy rule base is fixed, we can alter the scaling factors to modify effectively the distribution (the shape and size) of all of the membership functions defined on the appropriate domain, that is, changing the performance of the overall system.

Many researchers have performed an extensive series of the experiments that show how the process dynamic and static characteristics change when scaling factors of fuzzy controller are altered (Procyk and Mamdani, 1979; Brown and Harris, 1994; Albus 1975). These experiences are summarized in Table 1.

Besides, many results show that the type, shape and size of fuzzy membership function determine mapping relation between input and output parameters, and system characters (Brown and Harris, 1994). The shape of error membership function affects sensibility of system directly. When the curve shape of the system error membership function is flatter, system sensibility becomes inferior, and when the shape of the membership function for the rate of system error change is the flatter, system has better stability. Hence, in order to improve system performance, uneven distribution of membership function can be effective, that is, when system error is smaller, the sharper shape of error membership function should be use and when system error is larger, the flatter shape of the curve should be use. As for the membership function of system error change rate, when system error change rate is smaller, the shape of membership function should be flatter and when system error is larger, the curve shape should be sharper. To realize similar effects, CMAC-based adaptive scaling factor is proposed in the followings.

3. CONTROL SYSTEM STRUCTURE AND LEARNING ALGORITHMS

The structure and learning algorithm for adaptive fuzzy controller with CMAC-scaling factors are now given.

3.1 Control system structure

The final form of the fuzzy controller to be defined is shown in Fig.1. In the figure, the differences with basic fuzzy controller are only scaling factors and additional learning part, which are implemented by CMAC network and corresponding update rule.

Table 1 Fuzzy controller parameter and system property change from small to big with constant scaling grade

	Definition	***Relationship of mapping***	***Static error***	***Overshoot***	***Transient time***
$\boldsymbol{K_e}$	$K_e=n_e/e_{max}$	e_{max} region reduce	Reduce	Increase	Reduce
$\boldsymbol{K_{ec}}$	$K_{ec}=n_{ec}/ec_{max}$	ec_{max} region reduce	Increase	Reduce	Increase
$\boldsymbol{K_u}$	$K_u= u_{max}/n_u$	u_{max} region increase	Reduce	Reduce	Reduce

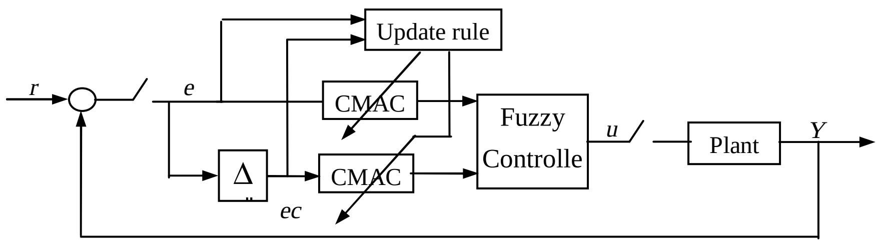


Fig.1 Block diagram of the proposed CMAC-Fuzzy control

3.2 Fuzzy controller

In Fig.1, the structure of fuzzy controller is a basic fuzzy controller with two-term input and single output. This fuzzy controller is constructed based on the following assumptions:

1) The fuzzy rule base is chosen as a linear one and shown in Fig.5 (Li, *et al.*, 1997);
2) The membership functions are chosen as the triangular or Gauss shape with even distribution;
3) Mamdani's max-min inference method is used in inference part (Procyk and Mamdani, 1979);
4) Defuzzification chose a center of gravity method (Brown and Harris, 1994), that is,

$$u = \Sigma\mu_u U / \Sigma\mu_u \tag{3}$$

where μ_u is membership function of *U*, and *U* is linguistic value of *u*.

3.3 CMAC Neural network

CMAC neural network was proposed by Albus in (Albus, 1975). The block diagram of CMAC is shown in Fig.2. From Fig.2, we can see that CMAC algorithm can be decomposed into three separate mapping, that is,

S—M—A—u ,

where

S = {input vectors};
M = {intermediate variables},
A = {association variables},
u = {output of CMAC},
f_1 *(S---M mapping)* is an input coding,
f_2 *(M---S mapping)* an address computing,
f_3 *(A---u mapping)* an output computing.

In this system, since CMACs are only used in tuning scaling factors, the number of memory locations in A is not so large in practice. Therefore, in this paper, the basic CMACs are used, which does not use any randomly mapping.

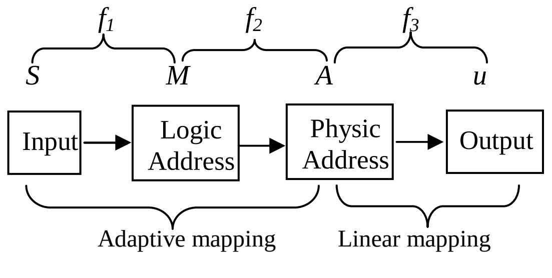


Fig. 2. The block diagram of CMAC

Fig.3 shows the CMAC without random mapping for the two-dimensional input and one-dimensional output consisting of 3 overlays and 12 basis functions.

The lattice cells are numbered from 1 to 16. Assume the input to the CMAC as $\boldsymbol{s} = (s_1, s_2)$ and the input space as:

$$S = \{ (s_1, s_2) \mid x_1 < s_1 < x_2, y_1 < s_2 < y_2 \}.$$

Then s_1 and s_2 are quantized with quantization intervals L_1 and L_2, respectively. In the first overlay, s_1 is quantized into *A* or *B*, and s_2 is quantized into *a* or *b*, respectively. The pairs of *aA*, *aB*, *bA*, *bB* express basis functions, and $aA(w_j)$ implies that the basis function *aA* has the weight w_j. When the input $\boldsymbol{s}$ is given to the 11th lattice cell, it specifies *aA*, *dA* and *fF*, and the CMAC output $w_1+w_8+w_{12}$.

Suppose that the desired signal for the input $\boldsymbol{s}$ is *d* and learning rate is *g*, the CMAC is then learned by the following correction factor δ to all weights corresponding to $\boldsymbol{s}$:

$$e = d - u$$
$$\delta = ge/k$$

Defining the threshold as e_l, the learning is continued until the following inequality is satisfied:

$$|e| < e_l,$$

Using such learning scheme, the CMAC can approximate a wide variety of nonlinear functions.

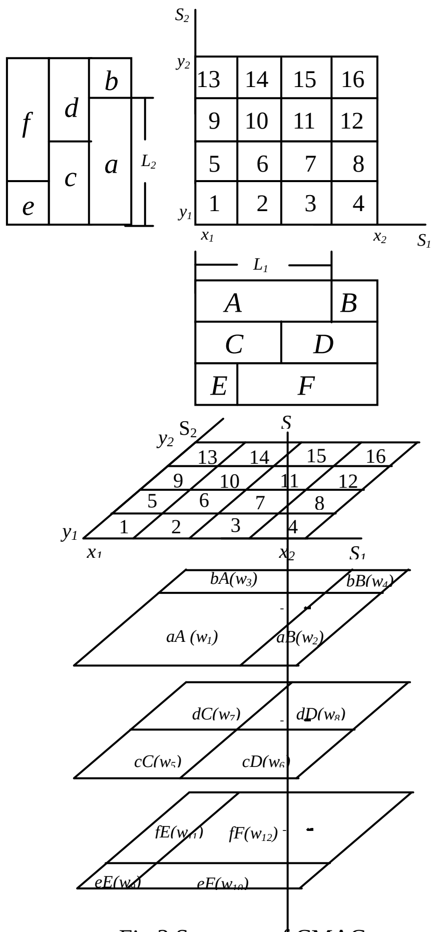


Fig.3 Structure of CMAC

3.4 Learning process in the proposed method

In Fig.1 the input of CMAC is system error or error change rate signal, that is, 1-input 1-output CMACs are used as the scaling factors. Here CMACs are trained by initialization process and on-line learning process. The initialization learning process is completed by off-line learning. The initialization and on-line learning processes are explained in the followings.

3.4.1 Initialization process

It is difficult to employ the controller constructed by an adaptive law or a learning law to the real plant without a suitable preparation. Therefore, the initialization learning process is very important for making controller work properly. The block diagram of the initialization learning process of the specified CMAC is shown in Fig4.

In Fig 4, *e(t), ec(t)* are system error and error change rate signal separately, that is, the input signal of the CMAC, defined as:

$$e(t) = r(t) - y(t), \qquad (4)$$
$$ec(t) = e(t) - e(t-1), \qquad (5)$$

where *r(t)* is system input signal, *y(t)* is system output signal, and $K_{e0}(e)$, $K_{ec0}(ec)$ are initialization functions, that is, the training signals. In the proposed method, two kinds of training signal are used:

（Ⅰ） conventional constant scaling factor

$$\boldsymbol{K_0} = [\, K_{e0} \quad K_{ec0} \,] = [\, n_e/e_{max} \quad n_{ec}/ec_{max} \,], \qquad (6)$$

(II) scaling factor function

$$\boldsymbol{K_0}(e,ec) = [\, K_{e0}(e) \quad K_{ec0}(e) \,] = [\, a_e \exp(-b_e|e|) \quad a_{ec}\exp(b_{ec}|ec|) \,], \qquad (7)$$

where

$$\begin{cases} a_e = n_e \exp(1)/e_{max}, \\ b_e = 1/e_{max}, \\ a_{ec} = 1/K_{ec0}, \\ b_{ec} = 1/ec_{max}, \end{cases} \qquad (8)$$

K_{e0}, K_{ec0}, $K_{e0}(e)$, $K_{ec0}(ec)$ are used in order to determine the initial weights in the on-line learning of the CMAC.

The weights *W* are updated so that $\boldsymbol{K_0}(t)$ approaches to $\boldsymbol{K_0}$, where the off-line learning is carried out based on the following algorithm:

$$W(t) = W(t) + g_1(\boldsymbol{K_0} - \boldsymbol{K_0}(t))/\rho, \qquad (9)$$

Where ρ denotes the total number of the selected weights in the CMAC and the g_1 is the learning rate.

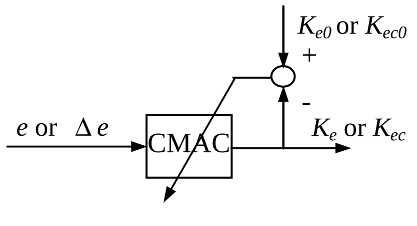


Fig.4 Block diagram of initialization learning

3.4.2 On-line learning process

After the initialization process is carried out enough, the learning phase is changed from the off-line to the on-line. The weights obtained in the initialization process are used as initial weights in the on-line learning. Fig.1 gives out the block diagram of the on-line learning process of the CMAC.

In on-line learning, the desired signal for tuning cannot be obtained. Therefore, the error *e(t)* and error change rate *ec(t)* of desired output signal *r(t)* and real output signal *y(t)* are introduced to on-line learning process, and the on-line learning is performed so that the system output *y(t)* approaches to *r(t)*, that is, the following algorithm is employed:

$$W(t) = \begin{cases} W(t) & : e(t) = 0,\ ec(t) = 0, \\ W(t) + g_2\,\mathrm{sign}(e(t))/\rho & : ec(t) = 0,\ e(t) \neq 0, \\ W(t) - g_2\,\mathrm{sign}(ec(t))/\rho & : e(t) = 0,\ ec(t) \neq 0, \\ W(t) - g_2\,\mathrm{sign}(e(t)*ec(t))/\rho & : otherwise, \end{cases} \qquad (10)$$

where g_2 is learning rate of the on-line learning.

According to the above algorithm, the CMACs are trained on-line.

4. SYSTEM PARAMETER DESIGN AND COMPUTER SIMULATION

In this section, to show the applicability of the proposed adaptive fuzzy controller, computer simulations are carried out, the results are described and compared with a conventional fuzzy controller.

4.1 Plant

The nominal plant to be controlled is a second-order process described as

$$G(s) = k/(T_1 s+1)(T_2 s+1)\,. \qquad (11)$$

where T_1, T_2, and k are 8,2 and 5, respectively. By selecting sample time as T_s=0.05, discrete representation of plant is obtained:

$$y(k) = a_1^* y(k-1) + a_2^* y(k-2) + b^* u(k-1), \qquad (12)$$

where a_1^*, a_2^*, b^* are nominal values of plant parameters, and their values are *1.9691, -0.9693, -0.0154,* respectively.

Assuming there exists modeling error in plant parameters, computer simulations are carried out for following cases:

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + bu(k-1)\,, \qquad (13)$$

case (a) $a_1 = a_1^*, a_2 = a_2^*, b = b^*$,
case (b) $a_1 = a_1^*, a_2 = a_2^*+0.0386, b = b^*$,
case (c) $a_1 = a_1^*, a_2 = a_2^*+0.0615, b = b^*$,
case (d) $a_1 = a_1^*, a_2 = a_2^*-0.1403, b = b^*$,
case (e) $a_1 = a_1^*, a_2 = a_2^*-0.2807, b = b^*$,
case (f) $a_1 = a_1^*+0.05901, a_2 = a_2^*, b = b^*$,
case (g) $a_1 = a_1^*+0.0299, a_2 = a_2^*+0.0386, b = b^*$,
case (h) $a_1 = a_1^*+0.0691, a_2 = a_2^*+0.0386, b = b^*$.

Also assuming the system input signal *r(t)* is step signal that has magnitude of 10.

4.2 The parameter design

The numbers of scaling grades of error, error change rate and controller output are set as 6-grades. k_u is selected as -16.234 according to conventional design method of scaling gain. The parameter regions are assumed as $[-e_{max} \quad e_{max}] = [-10 \quad 10]$, $[-ec_{max} \quad ec_{max}] = [-4 \quad 4]$.

4.2.1 Choosing of control rule base and membership function

In order to show universality of the system, linear fuzzy rule base is selected as shown in Fig.5, which has seven linguistic terms for each input and output. The input and output membership functions are chosen even distribution triangle-shaped with trapezoid-shaped as shown in Fig.6.

$EC=K_{ec}ec$

EC	NL	NM	NS	Z	PS	PM	PB
PB	Z	PS	PM	PB	PB	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PS	NM	NS	Z	PS	PM	PB	PB
Z	NB	NM	NS	Z	PS	PM	PB
NS	NB	NB	NM	NS	Z	PS	PM
NM	NB	NB	NB	NM	NS	Z	PS
NB	NB	NB	NB	NB	NM	NS	Z

$E=K_e e$

Fig .5 A two-dimensional linear rule base

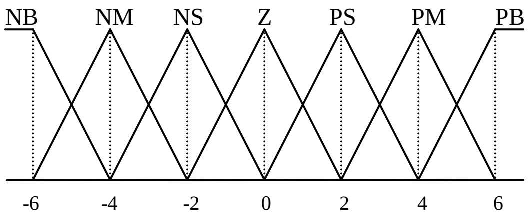


Fig.6 Figure of input and output membership function

4.2.2 Initialization training signal in the process

As we have mentioned in 3.3.1 the training signal for the initialization process of CMAC is determined from (6), (7), (8). Then we compared control performances of the following two cases:

(I) training signal is calculated based on the conventional scaling factor, that is:

$$\boldsymbol{K_0} = [K_{e0}\ K_{ec0}] = [6/10\ 6/4] = [0.6\ 1.5]$$

(II) calculated by

$$\boldsymbol{K_0}(e,ec) = [K_{e0}(e)\ K_{ec0}(ec)] = [\ 1.62\exp(-0.1|e|)\ 1.5\ \exp(0.25|ec|)]\ ,$$

where

$b_e=1/\ e_{max0}\ = 0.1,$
$a_e= n_e\ \exp(1)/e_{max0} = 1.62,$
$b_{ec}= 1/\ ec_{max0} = 0.25,$
$a_{ec}= K_{ec0} = 1.5.$

In addition, in order to show advantages of the proposed method, we conducted the third simulation,

(III) using a non-adaptive conventional fuzzy controller.

4.3 Simulation results

The simulation results are shown in Fig.7. In the figures:

(1) In Fig. 7 (a), rise times and maximum overshoots of methods (I), (II) are reduced as compared to method (III).

(2) The robustness of the proposed method becomes better than that of the conventional method. In Figs. 7 (c), (e), (f), with changing plant parameters, the results have become diverge in method (III), but the are kept stable in methods (I), (II).

(3) In Figs. 7 (c), (e), (f), method (II) shows better performance than method (I).
(4) Though steady state error are increased when plant parameters change, method (II) has smallest steady state error in the three methods in Figs.7 (b), (d), (g), (h).

Conventional fuzzy control theory is proven very effective. However, when plant parameters change, the property of conventional fuzzy controller becomes unfavorable. By using the proposed method, better control results is obtained, as compared to conventional fuzzy control, especially, in cases there exist modeling errors.

5. Conclusion

In this paper, the scheme of adaptive fuzzy controller with CMAC-based adaptive scaling factor has been presented. Then, learning phase of CMAC consists of two steps: initialization process and on-line learning process. For initialization process, two kinds of training signal have been given, that is, constant scaling factors and non-linear scaling factor functions. Simulation results show that by using the proposed method, control performance can be improved as compared to conventional fuzzy control. It is also demonstrated that the proposed methods have robustness for modeling error.

CMAC network is used only in tuning scaling factors, so that the design of fuzzy inference rule and membership function becomes more lenient. For example, in the simulations, we could select simple linear control rule and even distributing triangle membership function.

REFERENCES

Albus, J.S. (1975), A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC), *Trans. ASME. Jnl. Dyn. Sys. Meas. And Control, Vol. 63, No.3, 1975,* pp.220-227.
Brown, M. and C. Harris (1994), *Neurofuzzy adaptive modelling and control,* (Prentice Hall

International Limited (Ed)), pp. 218-257,409-430

Hirashima, Y., Y. Iiguni and N. Adachi (1997), An adaptive control system design using a memory based learning system, *INT. J. CONTROL. 1997. Vol. 68, No.5.* pp.1085-1102

Kandelb, A. and G. Langholz (1994), *Fuzzy control systems*,(CRC (Ed)), pp:236-142.

Lee, C. C. (1990), Fuzzy logic in control systems : Fuzzy logic controller – part I, *IEEE Trans. Syst., Man, Cybern.,Vol.20, No.2,* pp.404-417

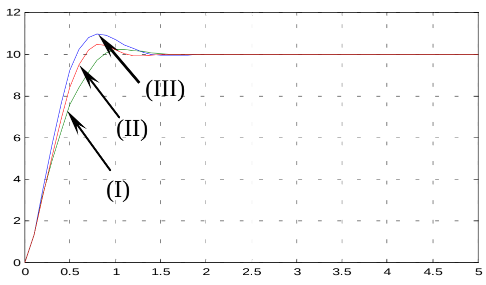
Li, H. X. (1999), Approximate model reference adaptive mechanism for nominal Gain Design of fuzzy control system, *IEEE Trans. Syst., Man, Cybern.,Vol.29,No.1,* pp.41-46.

Li, H. X., H. B. Gatland and A. W. Green (1997), Fuzzy Variable Structure Control, *IEEE Trans. Syst., Man, Cybern., Part B: Cybernetics, Vol.27, No.2,* pp.306-312.

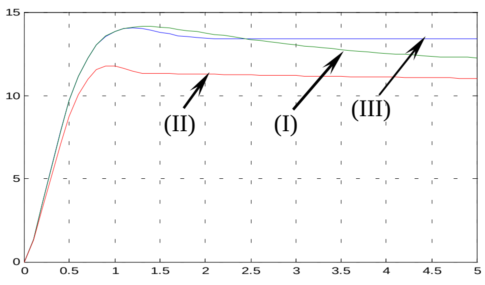
Procyk, T.J. and E.H. Mamdani (1979). A Linguistic Sefl-Organising Process Controller, *Automatica, Vol.15*, pp. 15-30

Sutto, R. and I.M. Jess (1991), A Design Study of a Self-organizing Fuzzy Autopilot for Ship Control, IMEchE, *Proc.Instn.Mech. Engrs., Vol.205*,pp.35-47.

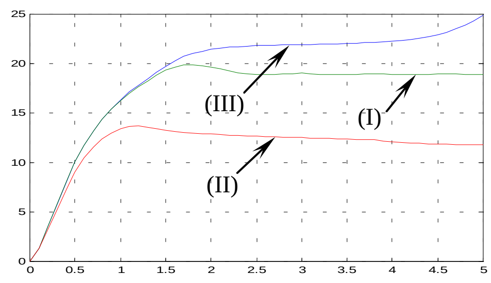
Yamamoto, T. and M. Kaneda (1999), Intelligent Controller Using CMACs with Self-Organized Structure and Its Application for a Process System, *IEICE TRANS. Fundamentals, Vol.E82-A,No.5, May,* pp.856-860



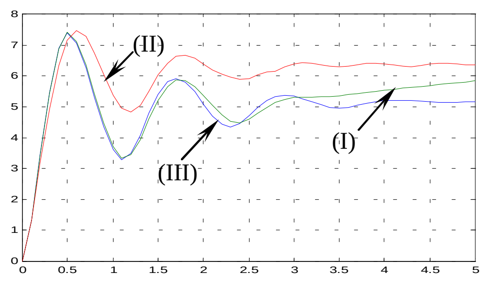
(a) $y(k)=a_1y(k-1)+a_2y(k-2)+bu(k-1)$



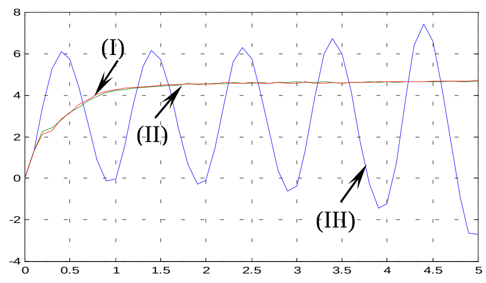
(b) $y(k)=a_1y(k-1)+(a_2+0.0386)y(k-2)+bu(k-1)$



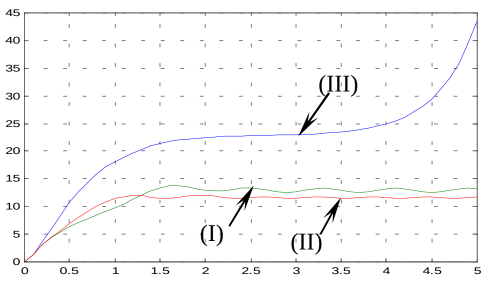
(c) $y(k)=a_1y(k-1)+(a_2+0.0615)y(k-2)+bu(k-1)$



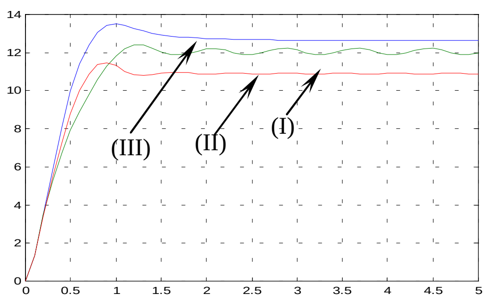
(d) $y(k)=a_1y(k-1)+(a_2-0.1403)y(k-2)+bu(k-1)$



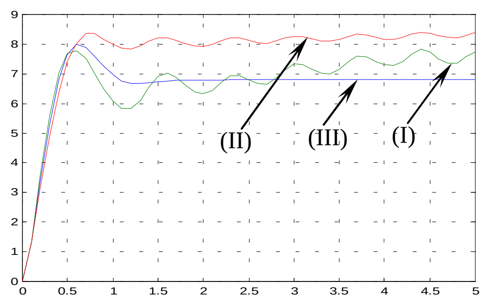
(e) $y(k)=a_1y(k-1)+(a_2-0.2807)y(k-2)+bu(k-1)$



(f) $y(k)=(a_1+0.05901)y(k-1)+a_2y(k-2)+bu(k-1)$



(g) $y(k)=(a_1+0.0299)y(k-1)+a_2y(k-2)+bu(k-1)$



(h) $y(k)=(a_1+0.0691)y(k-1)+a_2y(k-2)+bu(k-1)$

Fig.7. Simulation results