15th Triennial World Congress, Barcelona, Spain

MODEL-REFERENCE ADAPTIVE CONTROL USING ASSOCIATE MEMORY NETWROK

Xiang-Jie Liu, Felipe Lara-Rosano

Centro de Instrumentos UNAM, Mexico City, D.F. 04510, Mexico

C.W Chan

Department of Mechanical Engineering, The University of Hong Kong

Abstract: Model-reference adaptive control with neurofuzzy methodology is derived in this paper. Associate memory network(AMN) is investigated in detail to be the possible implementation as the direct self-tuning nonlinear controller. The essence of the neurofuzzy controller has been discussed and the local stability of the system is reached. The performance of the model-reference adaptive neurofuzzy controller is illustrated by examples involving both linear and nonlinear systems. *Copyright©2002IFAC*

Keywords: model-reference adaptive control, neural networks, fuzzy control, nonlinear

1. INTRODUCTION

Model-reference adaptive control(MRAC) has been an important research branch in self-tuning control area (Ioannou, and Sun, 1996). MRAC uses the error between the reference model output and the real plant output to adjust the control gain in order to force the plant to follow the desired response of the reference model. The control value u(t) could be linear combination of the states of the system(Fig.1).

$$u(t) = \theta^T(t)\omega(t) \quad (1)$$

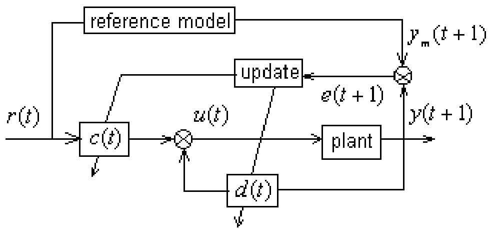

Fig.1 Model reference adaptive control system

$\omega \cong [r(t), r(t-1), \cdots, y(t), y(t-1), \cdots]$ is signal state vector and $\theta^T \cong [c_1, c_2, \cdots, d_1, d_2, \cdots]$ is weigh vector. With cost function:

$$J = \frac{1}{2}e^2 = \frac{1}{2}(y - y_m)^2 \quad (2)$$

The parameter θ^T could be changed in the direction of the negative gradient of J, i.e.

$$\dot{\theta} = -g\frac{\partial J}{\partial \theta} = -ge\frac{\partial e}{\partial \theta} \quad (3)$$

$g > 0$ is called the self-tuning factor determining the rate of decrease.

MRAC has so far been proved to be an effective method over linear plant and thus found applications in many aspects of process control. Comparing with conventional fixed parameter PID controller, it can better adaptive to the parameter change of the plant and thus has drawn much attention in the field of control engineering. For nonlinear system, linear MRAC works on the linearised model about a local operating point. The working effect depends heavily on the nonlinearity of the plant.

Neural networks have been used in the adaptive control of nonlinear systems(Narendra, and Parthasarathy, 1990). Indirect adaptive control using neural networks is presented by Bittanti and Piroddi

(1994). Direct MLP model reference adaptive control comparing with Lyapunov method has been presented by Lightbody and Irwin (1995). Also fuzzy theory have been combined with adaptive control(Takagi, and Sugeno, 1985) which is aimed at solving the problem of uncertainty and thus introduce nonlinearity. The combination of neural network and fuzzy control constitutes the neurofuzzy networks (Brown, and Harris, 1994), in which fuzzy rules could easily express the expert knowledge in linguistic form while neural networks posses the learning ability which could approximate nonlinear functions with arbitrary accuracy. Previous work include fuzzy neural networks for nonlinear systems modelling (Zhang, and Morris, 1995), GMV controller based on the neurofuzzy networks with a simplified recursive least squares method (Chan, *et al.,* 2000). Among the several neurofuzzy network structure, AMN(associate memory network) provides a direct link between artificial neural networks and the fuzzy systems. B-spline network could be one form of lattice AMN whose univariate basis functions represent fuzzy linguistic statements. These networks therefore embody both a qualitative and a quantitative approach, enabling heuristic information to be incorporated and inferred from neural nets, and allowing fuzzy learning rules to be derived.

The purpose of this paper is to analyze a kind of direct model reference adaptive control using neurofuzzy networks. As nonlinear controller, the structure of AMN is analyzed. The essence of the neurofuzzy controller and the local stability are discussed. Two case studies involving linear steam-boiler system and nonlinear system respectively are presented to illustrate the implementation and the performance of the MRAC neurofuzzy controllers.

2. NEUROFUZZY NETWORK IMPLEMENTATION OF MRAC

For the following input-output model of the nonlinear plant (Bittanti, and Piroddi, 1994):

$$\begin{aligned} y(t+1) &= a_0 y(t) + \cdots + a_{n-1} y(t-n+1) \\ &+ f[u(t), u(t-1), \cdots, u(t-m+1)] + e(t) \end{aligned} \quad (4)$$

u(t), y(t) are the control and the output of the system. Variable m, n are respectively the orders of control and output of the system, which are assumed known, and e(t) is a sequence of independent random variables with common variance σ^2. Assuming that f[.] is a smooth nonlinear function such that a Taylor series expansion exists. And:

$$\frac{\partial f}{\partial u(t+1)} = 0 \qquad \frac{\partial f}{\partial u(t)} \neq 0 \quad (5)$$

to ensure that the plant Jacobian exist.

By choosing the state variables as

$$\left.\begin{aligned} x_1(k) &= y(k-n+1) \\ &\vdots \\ x_{n-1}(k) &= y(k-1) \\ x_n(k) &= y(k) \end{aligned}\right\} \quad (6)$$

the state-space equation of the nonlinear system is:

$$\begin{aligned} X(k+1) &= AX(k) + Bf[u(t), \cdots, u(t-m+1)] \\ y(k) &= CX(k) \end{aligned} \quad (7)$$

$$A = \begin{bmatrix} 0 & 1 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & & \cdots \\ 0 & \cdots & \cdots & 0 & 1 \\ a_0 & a_1 & \cdots & \cdots & a_{n-1} \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 1 \end{bmatrix}$$

where $X = [x_1 \ x_2 \cdots x_n]^T \in R^n$ is the state vector.

2.1 AMN based model reference adaptive controller

The constitution of neurofuzzy MRAC system share the common architecture, just that the controller is composed of a the B-spline AMN (Fig.2). The state variables $x(t) = [y(t) \cdots y(t-n_y+1), r(t) \cdots r(t-n_r+1)]$ are the inputs of the neurofuzzy network. The B-spline network is initially designed to specify the shape(order) of each of the univariate basis functions, and this implicitly determines the number of basis functions mapped to for a particular network input. There exist recurrence relationship for evaluating the membership of a univariate B-spline basis function of order k with r inner knots(Brown, and Harris, 1994)and they have several desirable properties:

1. The basis functions are defined on a bounded support and the output of the basis function is positive on its support , i.e. $\mu_k^j(x) = 0, \ x \notin [\lambda_{j-k}, \lambda_j]$ and $\mu_k^j(x) > 0, \ x \in (\lambda_{j-k}, \lambda_j)$
2. The basis functions form a partition of unity. For any network input, the sum of the outputs of the basis functions is always one, i.e $\sum_j \mu_k^j(x) \equiv 1, \ x \in [x_{\min}, x_{\max}]$.

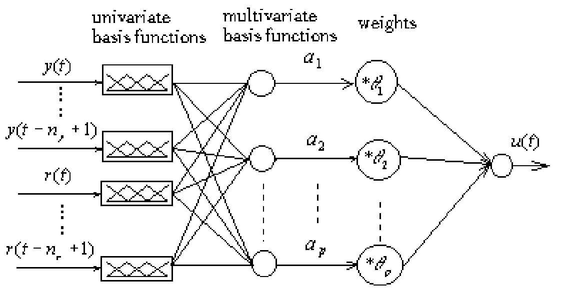

Fig. 2: B-spline neurofuzzy network controller

It is obviously that the univariate B-spline basis function can be used to represent the fuzzy membership functions which implement the fuzzy linguistic terms. The product operator combining the univariate basis functions represents a fuzzy conjunction. Therefore the neurofuzzy network shown in figure 2 could be expressed as a set of fuzzy production rules :

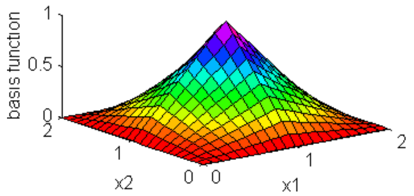

Fig.3 Multivariate basis function

Rp: IF $y(t)$ is negative large and $\cdots, y(t-n_y+1)$ is negative medium and $r(t)$ is positive medium and $\cdots, r(t-n_r+1)$ is positive medium

THEN $u(t)$ is positive medium

The total number of rules is given by

$$p=\prod_{i=1}^{n}(R_i+k_i) \qquad (8)$$

The multivariate basis function is the tensor products of the outputs of the univariate B-spline basis functions, i.e.

$$a_i(x(t))=\prod_{k=1}^{n}\mu_{A_k^i}(x_k(t)) \qquad (9)$$

where $i=1,2,\cdots,p$ and $n=n_y+n_r$ is the dimension of the input vector x(t). Thus the desirable properties of the univariate B-spline basis functions are all extended in a natural way to the multivariate basis functions. They are defined on hyperrectangles of size $(k_1\times k_2\times\cdots\times k_n)$ and therefore possess a bounded support. The output is positive inside this domain and zero outside. Fig.3 shows the multivariate basis function formed from two ,order 2, univariate basis function.

The MRAC control law can be expressed by:

$$u(k)=\sum_{i=1}^{p}\theta_i a_i(x(t))=\sum_{i=1}^{p}\theta_i\prod_{k=1}^{n}\mu_{A_k^i}(x_k(t)) \qquad (10)$$

where $\theta=[\theta_1\theta_2\cdots\theta_p]$ is the weight vector and x(t) is the input vector .

2.2 Local change property

In the neurofuzzy controller, the input space is separated into q regions given by

$$q=(R_y+1)^{n_y}(R_r+1)^{n_r} \qquad (11)$$

Only one region is activated each time by the input, whilst elements in the other regions are zero. The number of non-zero elements is given by

$$p'={k_y}^{n_y}{k_r}^{n_r} \qquad (12)$$

In figure 4, the language variable of two inputs to the network are "positive large", "positive medium", "positive small", "zero", "negative small", "negative medium", "negative large" for x_1; and "positive large", "positive medium", "positive small", "negative small", "negative medium", "negative large" for x_2. In this case, $k_1 = k_2 = 2$, $R_1 = 5$ and $R_2 = 4$, the number of regions q is 30. The total memory storage p is 42. Each time there are four fuzzy rules activated. The antecedent of this four rules could be expressed as: if x_1 is "negative small" and x_2 is "negative small"; if x_1 is "negative small" and x_2 is "positive small" ; if x_1 is "zero" and x_2 is "negative small" ; if x_1 is "zero" and x_2 is "positive small". Fig. 5 shows the membership function of the operator regions.

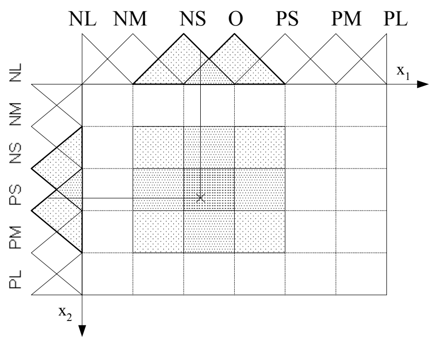

Fig. 4:Local change property of neurofuzzy networks

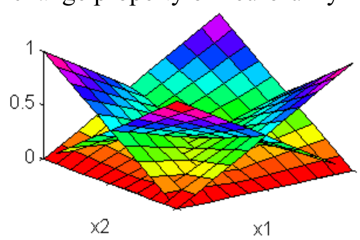

Fig.5 Membership function of the operator regions

3. ESSENCE OF THE NEUROFUZZY CONTROLLER AND ITS STABILITY

The proposed neurofuzzy controller could be decomposed into two part: a static, nonlinear, topology conserving map and an adaptive linear mapping so that it is in fact a fuzzy controller with its output weights learned by neural network. Originally, fuzzy controller refers to the two-dimensional controller with error (between the setpoint and the output) and change of error as its inputs. This idea is from the human experience in real-time control representing the following step-respond characteristic "if the plant output is far from the setpoint and it is moving away from the setpoint, then increase the control value to force the output back". Fundamentally, a two-dimensional fuzzy controller with linear control rules is equivalent to the sum of a global two-dimensional multilevel relay and local nonlinear PI controller (Ying, 1993). A three-dimensional fuzzy controller with linear control rules is the sum of a global three-dimensional multilevel relay and local nonlinear PID controller(Liu, *et al.,* 1997). Functionally, neurofuzzy controller could be nonlinear self-tuning regulator. Ching *et al.* (1995)shows that the output of fuzzy controller with multiple inputs can be represented by a linear parametric function of the inputs.

While fuzzy controller is combined with neural network, the constituted neurofuzzy network is an adaptive fuzzy controller. The state-space equation (7) could be express as:

$$X(k+1) = AX(k) + Bf[\sum_{i=1}^{p}\theta_i \prod_{k=1}^{n} \mu_{A_k^i}(x_k(t)),\ \sum_{i=1}^{p}{}^{(-1)}\theta_i \times \prod_{k=1}^{n} \mu_{A_k^i}(x_k(t-1)), \cdots, \sum_{i=1}^{p}{}^{-(m-1)}\theta_i \prod_{k=1}^{n} \mu_{A_k^i}(x_k(t-m+1))] \quad (13)$$

where the left superscripts $(-1),\cdots,-(m-1)$ of weights θ_i represent the corresponding values at previous updating steps. Let the setpoint $r=0$, and $\overline{X}=0$ to be the equilibrium point of the system. Choose the Lyapunov candidate V{X(k)} on the compact set S as:

$$V\{X(k)\} = \frac{1}{2}E^T E \quad (14)$$

where $E = X - \overline{X} = X$, then

$$V\{X(k)\} = \frac{1}{2}X^T X = \frac{1}{2}\{y(k-n+1)^2 + \cdots + y(k)^2\} \quad (15)$$

is positive define, and

$$V\{X(k+1)\} = \frac{1}{2}\{y(k-n+2)^2 + \cdots + y(k+1)^2\} \geq 0 \quad (16)$$

For $|\theta_i| \leq d, \quad \forall d \geq 0$, the derivative in Lyapunov function could be expressed as follows:

$$\begin{aligned}\dot{V}\{X(k+1)\} &\cong \sum_i \frac{\partial V\{X(k+1)\}}{\partial \theta_i}\Delta\theta_i \\ &= y(k+1)\sum_i \frac{y(k+1)}{\partial u(k)}\frac{\partial u(k)}{\partial \theta_i}\Delta\theta_i\end{aligned} \quad (17)$$

Sicne $$\Delta\theta_i(k) = -ge\frac{\partial e}{\partial \theta_i} \quad (18)$$

$$\begin{aligned}&\dot{V}\{X(k+1)\} \\ &\cong (-g)y(k+1)^2\sum_i \frac{y(k+1)}{\partial u(k)}\frac{\partial u(k)}{\partial \theta_i}\frac{\partial y(k+1)}{\partial u(k)}\frac{\partial u(k)}{\partial \theta_i} \\ &= (-g)y(k+1)^2\sum_i \left(\frac{y(k+1)}{\partial u(k)}\right)^2\left(\frac{\partial u(k)}{\partial \theta_i}\right)^2 \\ &= (-g)y(k+1)^2\sum_i \left(\frac{y(k+1)}{\partial u(k)}\right)^2 \left(a_i(x(t))\right)^2 \leq 0 \quad (19)\end{aligned}$$

where $\partial y(k+1)/\partial u(t) = \partial f/\partial u(t)$ is the plant Jacobian which has been supposed in the nonlinear model (4) to be exist. In such a way, the state will converge to the equilibrium point and the local stability of the system around the equilibrium point is reached.

4. CASE STUDIES

Example 1-Local linear steam-boiler system:

In general industry process, there exist almost no purely linear model. But under certain fixed working condition, linear model could be acquired within certain local operation region. Fuel combustion system is the main part of a boil-steam generation which produce high pressure steam to drive generator making power. For a typical load disturbance, while the coal fuel is constant, the control valve μ_T is opened as quickly as possible to yield an increase of output power (Fig.6). The steam flow increases and steam pressure P_b decreases with no time delay. Finally, steam pressure stay in a relatively lower level (fig.7).

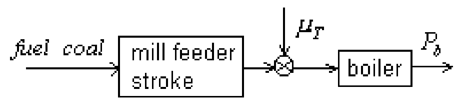

Fig.6 Steam pressure plant

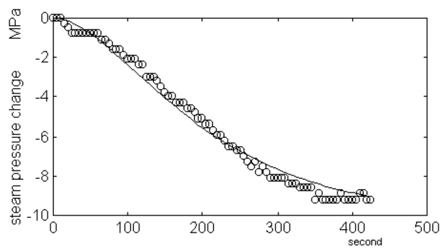

Fig.7 Fitting curve of steam pressure

Considering the output "steam pressure P_b" with the control input μ_T, the dynamic of the disturbance process could be considered to be a second order linear model transfer function.

$$G(s) = \frac{P_b(s)}{\mu_T(s)} = \frac{K_p}{(T_1 s+1)(T_2 s+1)} \quad (20)$$

The dynamic process posses different time constant under different load condition(Liu,1999). Under 250MW load condition, the identification parameter could be: $T_1 = 120s \quad T_2 = 100s$. K could be decided by the static parameter: $K = P_{\Delta b}/\mu_{\Delta T} = 0.01 kg/s/\%$

In order to get the second-order close-loop system, the conventional model reference adaptive controller should be chosen as:

$$u = \theta_1 r - \theta_2 y - \theta_3 \dot{y} \quad (21)$$

The reference model could be chosen a typical second order transfer function form

$$G_m(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (22)$$

Let $\xi = 0.7$ $\omega_n = 2$, the parameters updating equation could be depicted as:

$$\frac{d\theta_1}{dt} = -ge\frac{r}{s^2 + 2\xi\omega_n s + \omega_n^2}$$
$$\frac{d\theta_2}{dt} = ge\frac{y}{s^2 + 2\xi\omega_n s + \omega_n^2} \qquad (23)$$
$$\frac{d\theta_3}{dt} = ge\frac{sy}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

The control valve μ_T is changed in square wave mode, and initial condition of the controller is $\theta_{10} = \theta_{20} = \theta_{30} = 0.1$, $g = 1$. The closed-loop response curve of the output steam pressure is shown by the dashed line in Fig.8. With continuously updating the parameters, the transient process reaches a good tracking result. In practical situation, it could satisfy the demand of the real steam-boiler process.

The MRAC neurofuzzy controller is implemented with the same input as that in traditional MRAC. Each input is fuzzified by two triangular basis functions, representing the language variable "small" and "large". In this case $k_y = k_{\dot{y}} = k_r = 2$, and $R_y = R_{\dot{y}} = R_r = 0$. The number of weights of the neurofuzzy network p is: 2^3=8. The range of y(t) $\dot{y}(t)$ and r(t) is chosen to be between –1 and 3. The weights $\hat{\theta}(0)$ are set the same as that of the MRAC controller. In Fig.8 the dotted line shows the tracking of the set-point of the MRAC neurofuzzy controller. Also the controller forces the output steam pressure to follow the desired reference model, but the initial overshot is high and the parameter updating process takes longer time. In usual situation, the tranditional linear MRAC controller performs better on linear model than that of neufuzzy controller. This is because nonlinear controller is more complicated and there are more parameters to be updated. Nevertheless, nonlinear control has the potential to outperform linear methods. Similar control effect to that of linear controller with neurofuzzy controller could be acquired by optimization its membership function.

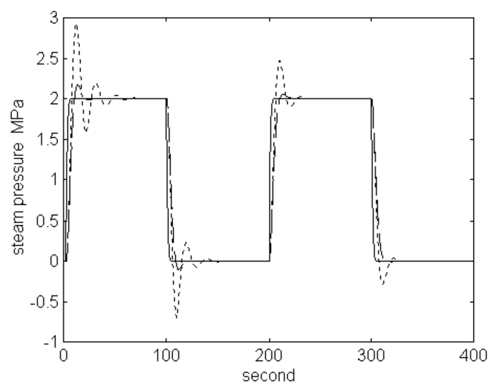

Fig8: Response of linear system

Example 2 Nonlinear model:

Consider the following nonlinear model (Bittanti and Piroddi, 1994):

$$y(t) = 0.3y(t-1) + 0.6y(t-2) + [u(t-k)]^{1/3} + e(t) \qquad (24)$$

where e(t)~N(0,0.1).

Choose variable $x(t) = [y(t), y(t-1), y(t-2), r(t)]$ to be the input of the MRAC, and reference signal to be the square wave, the parameters updating equation could be depicted as:

$$\frac{d\theta_1}{dt} = -ge(t)r(t)[u(t-1)]^{-\frac{2}{3}}$$
$$\frac{d\theta_2}{dt} = -ge(t)y(t)[u(t-1)]^{-\frac{2}{3}}$$
$$\frac{d\theta_3}{dt} = -ge(t)y(t-1)[u(t-1)]^{-\frac{2}{3}} \qquad (25)$$
$$\frac{d\theta_4}{dt} = -ge(t)y(t-2)[u(t-1)]^{-\frac{2}{3}}$$

The initial condition is chosen as $\theta_{10} = \theta_{20} = \theta_{30} = \theta_{40} = 0.1$, $g = 0.2$. The closed-loop response curve is shown in fig.9 which exhibits large peaks and oscillations. It reaches convergence just because the plant nonlinear is accounted for by adaptation of the parameters of a linear controller.

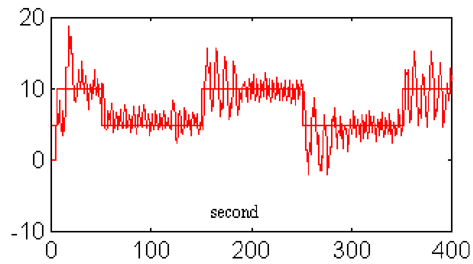

Fig.9: nonlinear system with linear MRAC

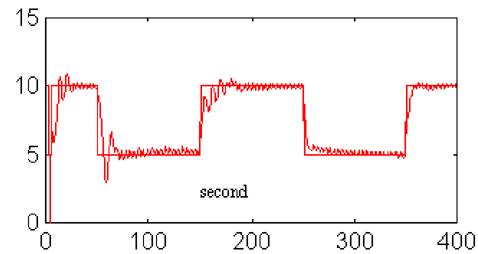

Fig.10 Nonlinear system with neurofuzzy controller

The MRAC neurofuzzy controller is implemented with the same input as that in MRAC. Each input is fuzzified by two triangular basis functions, representing "small" and "large". In this case $k_y = k_{y-1} = k_{y-2} = k_r = 2$, and $R_y = R_{y-1} = R_{y-2} = R_r = 0$. The number of weights of the neurofuzzy network p is: 2^4=16. The range of y(t) , $y(t-1)$, $y(t-2)$ and r(t) is chosen to be between 0 and 12. The weights $\hat{\theta}(0)$ are initially set the same as that of the MRAC controller. The parameters updating equation could be depicted as:

$$\frac{d\theta_i}{dt} = -ge(t)a_i(x(t))[u(t-1)]^{-\frac{2}{3}} \qquad (26)$$

Where $i = 1,2,\cdots,16$. Fig.10 shows the good tracking of the set-point with the MRAC neurofuzzy controller. In this case it no longer exhibits large peaks but shows small oscillations in stable state. To make the improvement, the local region division is changed, with each input fuzzified by five triangular basis functions, representing the language variable: "positive large", "positive medium", "zero", "negative medium" and "negative large". With typical B-spline network, this would require 5^4=625 storage locations and each input would activate 2^4=16 basis functions. To solve this problem, the controller is additively decomposed, such that it is a linear combination of two two-dimensional subnetworks(Fig.11):

$$u(t) = s_1[y(t), y(t-1)] + s_2[y(t-2), r(t)] \qquad (27)$$

In such a way, the memory requirements reduce to 50 and 8 basis functions are activated for each input. Fig.12 shows the improved tracking of the set-point with the MRAC neurofuzzy controller. In stable state the oscillations almost disapeare.

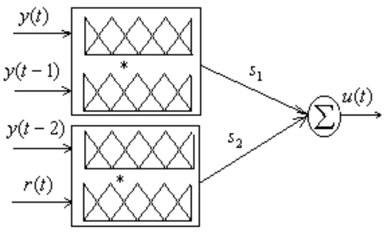

Fig.11 : Lower-dimension neurofuzzy network controller

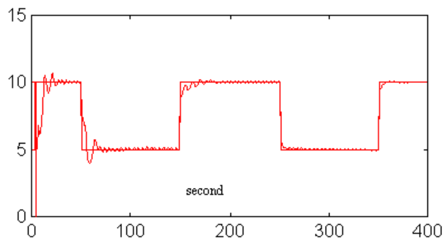

Fig.12: Neurofuzzy controller by improved result

5. CONCLUSION

A MRAC neurofuzzy controller is derived in this paper. AMN is chosen to introduce the nonlinearity and act as nonlinear direct adaptive controller. The effectiveness of the AMN exist in that the fuzzy control is realized by neural network. The essence of the neurofuzzy controller has been discussed and the local stability of the system has been achieved. From the two examples, the performance of the MRAC neurofuzzy controller for the linear system could be no better than that of the traditional MRAC controller, but is superior to traditional MRAC controller for the nonlinear system. While the trait of the neurofuzzy controller is further improved by choosing more univariate basis functions, the computational burden is increased exponentially. Decomposing way is introduced to solve the problem of "curse of dimensionality" to some extent.

REFERENCES

Brown, M. and Harris, C. J. (1994). Neurofuzzy adaptive modelling and control. Englewood Cliffs, NJ:Prentice Hall.

Bittanti, S.and Piroddi, L.(1994). GMV technique for nonlinear control with neural networks. *IEE Proc.-Control Theory Appl.*, **141**(2), 57-69.

Chan, C. W., CHEUNG, K. C., and YEUNG, W. K.(2000). A computation-efficient on-line training algorithm for neurofuzzy networks. *Int. J. Sys. Sciences*, **31**(3), 297-306.

Ching-Chong Wong et al. (1995). Studies on the Output of Fuzzy Controller with Multiple Inputs. *Fuzzy Sets and Systems,* **71**, 113-129.

Ioannou, P.A., and Sun, J. (1996). Robust adaptive control. Prentice-Hall, Englewood Cliffs, NJ,.

Lightbody, G.and Irwin,G.W. (1995). Direct neural model reference adaptive control. *IEE Proc.-Control Theory Appl.*, **142**(1), 31-43.

Narendra, K. S. and Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Trans.Neural Networks*, **1**, 4-27.

Tanaka,K., Sano,M., & Watanabe,H.(1995). Modeling and control of carbon monoxide concentration using a neuro-fuzzy technique. *IEEE Tran. fuzzy system,* **3**(3), 271-279.

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its application to modelling and control. *IEEE Trans*.SMC, **15**(1), 116-132.

Xiang-Jie Liu, Yi-Min Peng & Xiaoxin Zhou(1999). Identification of Boiler Models and Its Fuzzy Logic Strategy. Proc.the 14th IFAC World Congress. Volume O (pp.149-154).Beijing China.

Xiang-Jie Liu, Huan-Shui Zhang and Tian-You Chai. (1997). Structure analysis of three-dimensional fuzzy controller and its relationship to PID controller. Proc. the 36th IEEE Conference on Decision and Control (pp.3350-3351). San Diego, California USA.

Xiang-Jie Liu, Xiaoxin Zhou. Structure Analysis of Fuzzy Controller with Gaussian Membership Function," Proc.the 14th IFAC World Congress. Volume K (pp.201-206).Beijing China.

Ying H.(1993). A nonlinear fuzzy controller with linear control rules is the sum of a global two-dimensional multilevel relay and a local nonlinear proportional-integral controller. *Automatica,* **29**, 499-503.

Zhang, J. and Morris, A.J. (1995). Fuzzy neural networks for nonlinear systems modelling. *IEE Proc.-Control Theory Appl.*, **142**(6), 551-561.