ISSN 1009 - 3095 *Journal of Zhejiang University(SCIENCE)* V.1, No.3, P.311 - 316, July - Sep., 2000
http://www.chinainfo.gov.cn/periodical; http://www.zju.edu.cn;
http://lib.zju.edu.cn/English; http://www.zjupress.com

A RAPID FUZZY RULE EXTRACTION METHOD FOR FUZZY CONTROLLER*

YANG Jian-gang(杨建刚)†, WANG Ru-ming(王如明)
(*Dept. of Computer Science, Zhejiang University, Hangzhou, 310027, China*)
†Email: yangjg@cs.zju.edu.cn
Received Dec.26, 1998; revision accepted Dec.15, 1999

Abstract: Based on division of the three-dimensional space from data samples, the method proposed in this paper can rapidly extract fuzzy rules by using the fuzzy information of the samples. The principle of this approach is proved theoretically. Due to its simplicity this method can be used to extract fuzzy rules in real-time for an adaptive control system. Simulation results showed that this approach is effective and practical.

Key words: fuzzy control, rule extraction, space division, sample reappearance
Document code: A **CLC number**: TP183

INTRODUCTION

It is impractical to create a precise mathematical model for control engineering application to quite a few complex control objects. The fuzzy control technique, which with its characterizing features of robustness, stability, simple algorithm and high real-time response ability, does not need a precise mathematical model and so, is widely applied.

The key of the fuzzy control system design lies in the generation and definition of the fuzzy rules. But in many cases, it is hard to define the number of fuzzy rules and the specific form of each rule. However, by referring to the large number of data samples of the previous control operations, we can develop the fuzzy rules needed.

There have already been some methods to develop fuzzy rules from data samples. A relatively popular way is to use multi-layer feed-forward neural network to simulate the "fuzzification → fuzzy reasoning → defuzzification" process of the fuzzy system with the aid of neural network theory and transform the fuzzy rule generation process into the adjusting process of the network weights and limens of the neural network, taking advantage of the learning ability of the neural network (Yang Yupu et al., 1994; Li Gechen et al., 1997; Keller, 1992; Yang Jiangang, 1993).

These fuzzy control strategies, which are based on neural network theory, have problems such as the selection of the initial value of the network weights, the setting of the learning step in the learning process, the selection of the amount of the neurons in each network layer, which can lead to bad learning effect or even a failure if any selection is improper. More importantly, the learning speed of these methods is often very slow, for which reason only off-line learning is usually possible and the real-time adaptive control of fast systems is hard to be implemented.

This paper gives a method to extract specific fuzzy rules from data samples. Compared with the methods available (Jiao Licheng, 1993), it has the features such as the idea is natural, the calculation is simple, and it is fit for real-time fuzzy rules extraction and thus can meet the needs of on-line learning and make the adaptive control of the system possible. The result of simulation of a typical non-linear system has proved the reliability and practicability of the method.

PRINCIPLES OF THE ALGORITHM

1. Division of the domain

Many control problems can be attributed to

* Project(69775013) supported by Natural Science Foundation of China

the zero clearing process of the error and change-in-error of the controlled system. Therefore, without exception, the input data of the fuzzy controller should include error E and change-in-error E_C while the output data of the controller should include the control variable U. (Later, the variables E、E_C、U are also used to stand for their domain respectively). Here we should preferably assume that the fuzzy linguistic value in error E, change-in-error E_C and control variable U can be separated into 7 levels, that is {Negative Large(NL), Negative Medium(NM), Negative Small(NS), Zero(ZE), Positive Small (PS), Positive Medium(PM), Positive Large (PL)}. For the later's narrative convenience, the 7-level linguistic values are sequentially defined as $\{F_1, F_2, F_3, F_4, F_5, F_6, F_7\}$, and the usual form of fuzzy control rules is:

$$\text{if } (e = F_i \text{ and } e_c = F_j) \text{ then } u = F_k \quad i,j,k = 1,2,3,\cdots,7 \tag{1}$$

The shape of the 7-level fuzzy linguistic value's membership function is symmetric triangular and the overlapping degree of the neighbored fuzzy language is often set to 25%. As shown in Fig.1.

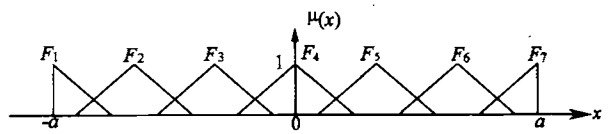

Fig.1 7-level fuzzy linguistic values' membership functions

For convenience, the domain of the input and output value is transformed into $[-a, a]$. If the actual domain of a varible is $[c, d]$, the transforming equation from the value x in the actual domain to the value y in $[-a, a]$ is

$$y = \frac{2a}{d-c}\left(x - \left(\frac{c+d}{2}\right)\right) \tag{2}$$

Definition 1 Any rule taking the form of if $(e = F_i$ and $e_c = F_j)$ then $u = F_k$ (in which F_i, F_j, F_k are respectively the fuzzy linguistic values in E, E_C, U) is called a virtual rule of the fuzzy control system.

Here the virtual rules of the fuzzy control system are summed up to $7^3 = 343$.

Each fuzzy linguistic value F_i in E determines an interval L_i on E accordingly (The interval determined by the neighbored fuzzy linguistic values in E is usually dealt with as adjacency relationship. A few overlaps are also acceptable. The union of the domains determined by the 7-level fuzzy linguistic value just covers the interval E), and to E_C, U, similar manipulations will be taken. This way, as far as a virtual rule in term of Equation (1) is concerned, according to the domain determined by the three fuzzy linguistic values in E, E_C, U respectively, a block in the 3-dimentional domain $E \times E_C \times U$ is determined.

To divide the domain according to virtual rules is based on the idea that: After input data are input to the fuzzy controller, through fuzzy reasoning, output control data are acquired. In the process of fuzzy reasoning, the minority parts of rules function decisively. The fuzzy linguistic values of these rules have relatively higher degreess of membership to input and output data. Therefore, we can regard the Cartesian product of the sensitive intervals (the intervals in which the degrees of membership are relatively high) of the three fuzzy linguistic values as the sensitive area the virtual rules correspond to.

Next we will discuss the strategy to settle the interval Li on corresponding domain according to each fuzzy linguistic value. The most direct way is to equally divide the domain according to the interval of the fuzzy linguistic values. (Here it is equally divided into 7 parts.). From left to right, each equal interval is set to be the corresponding interval of the fuzzy linguistic values F_1, F_2, F_3, F_4, F_5, F_6, F_7. As shown in Fig.2.

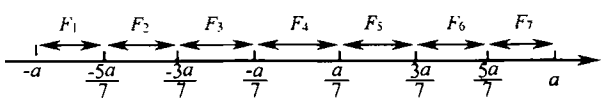

Fig.2 Equal division of the domain according to the interval of the fuzzy linguistic values

This paper adopts the following strategy to settle the corresponding interval of the fuzzy linguistic values.

Definition 2 Assume $f(x)$ is a non-negative function and its definition domain is set as A, then the set $\{x \mid x \in A,\ f(x) > 0\}$ is called the support of $f(x)$ in A, labeled S_{f_i}.

According to the definition of the membership function of the fuzzy linguistic value F in Fig.1, we label the membership function which corresponds to F_i as f_i, and call the support S_{f_i} as the corresponding interval of the fuzzy linguistic value F_i. As shown in Fig.3.

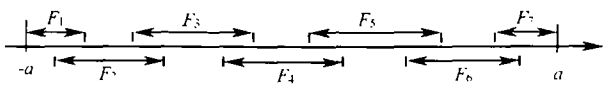

Fig.3 Definition of the membership function of the fuzzy linguistic value F

To the virtual rules in the form of if ($e = F_i$ and $e_c = F_j$) then $u = F_k$, we naturally take as the corresponding block of the virtual rules. The union of all the corresponding blocks of the virtual rules just covers the 3-dimentional domain $E \times E_C \times U$.

We are going to divide the 3-dimentional domain $E \times E_C \times U$ according to the virtual rules. The fuzzy information the samples carry is utilized to decode fuzzy rules. This will increase the speed of the rules extracting operation for the needs of on-line fuzzy rule extraction in the systems that require high real-time apacity. These are the fundamental principles of the fuzzy rule extracting method that this paper proposes.

2. Rule extraction

Definition 3 To a fuzzy control system derived from the rules extracted from samples, in the case that the system inputs equal input data of some sample, after the reasoning process of the fuzzy control system, the output control variable should equal or at least approximately equal the output data of the sample. This feature of fuzzy control system is called sample reappearance.

E, E_C, U above are regarded as continuous domain. However, in the process of fuzzy reasoning, discrete domain form is needed. Now we label the discrete domain of E, E_C, U as $\{x_1, x_2, \cdots, x_{N_1}\}$, $\{y_1, y_2, \cdots, y_{N_2}\}$, $\{z_1, z_2, \cdots, z_{N_3}\}$.

Theorem Assume (e_n, e_{cn}, u_n) to be a data sample, if it is required that the fuzzy control system derived from the data samples satisfy the feature of reappearance, then the fuzzy system must at least include the fuzzy rule if ($e = F_i$ and $e_c = F_j$) then $u = F_k$ while (1) $e_n \in S_{f_i}$; (2) $e_{cn} \in S_{f_i}$; (3) $u_n \in S_{f_k}$.

Proof Assume the input of the fuzzy control system is the input part (e, e_c) of the sample (e, e_c, u), and assume the reasoning process of the system follows such a strategy that each fuzzy rule is used in parallel and the subresults of each rule are merged into one. The actual reasoning process of each rule follows the strategy of decomposing reasoning.

Since data samples are accurate in value, 2-component fuzzy vectors are acquired after the operation of fuzzification.

Assume the fuzzy vector that $\boldsymbol{e}_n$ corresponds to is:

$$\boldsymbol{e}'_n = (0,0,\cdots,0,\underset{p^{\text{th}}}{1},0,\cdots,0) \tag{3}$$

The fuzzy vector $\boldsymbol{e}_{cn}$ corresponds to is

$$\boldsymbol{e}'_{cn} = (0,0,\cdots,0,\underset{q^{\text{th}}}{1},0,\cdots,0) \tag{4}$$

The fuzzy vector $\boldsymbol{u}_n$ corresponds to is

$$\boldsymbol{u}'_n = (0,0,\cdots,0,\underset{x^{\text{th}}}{1},0,\cdots,0) \tag{5}$$

Take any rule: if ($e = F_i$ and $e_c = F_j$) then $u = F_k$, and the corresponding fuzzy reasoning process is shown as follows:

$$\begin{aligned} u'_1 &= e'_n \times (F_i \times F_k) \\ &= (0,0,\cdots,0,\underset{p}{1},0,\cdots,0) \begin{pmatrix} \mu_{F_i}(x_1) \\ \mu_{f_i}(x_2) \\ \vdots \\ \mu_{F_i}(x_{N_1-1}) \\ \mu_{F_i}(x_{N_1}) \end{pmatrix} \cdot \\ &\quad (\mu_{F_k}(z_1), \mu_{F_k}(z_2), \cdots, \mu_{F_k}(z_{N_3})) \\ &= \mu_{F_i}(x_p) \wedge (\mu_{F_k}(z_1), \mu_{F_k}(z_2), \cdots, \mu_{F_k}(z_{N_3})) \end{aligned} \tag{6}$$

$$u'_2 = e'_{cn} \times (F_j \times F_k)$$

$$= (0,0,\cdots,0,\underset{q}{1},0,\cdots,0)\begin{pmatrix}\mu_{F_j}(y_1)\\ \mu_{F_j}(y_2)\\ \vdots\\ \mu_{F_j}(y_{N_2-1})\\ \mu_{F_j}(y_{N_2})\end{pmatrix}\cdot$$

$$(\mu_{F_k}(z_1),\mu_{F_k}(z_2),\cdots,\mu_{F_k}(z_{N_3}))$$

$$= \mu_{F_j}(y_q)\wedge(\mu_{F_k}(z_1),\mu_{F_k}(z_2),\cdots,\mu_{F_k}(z_{N_3})) \quad (7)$$

$$u' = u'_1 \wedge u'_2 = \min(\mu_{F_i}(x_p),\mu_{F_j}(y_q)\wedge (\mu_{F_k}(z_1),\mu_{F_k}(z_2),\cdots,\mu_{F_k}(z_{N_3})) \quad (8)$$

Here maximum and minimum reasoning operations are adopted in matrix operations. "$\wedge$" between vectors stands for the minimizing operation according to the corresponding element and when it is between a numerical value and a vector, it stands for the minimizing operation between the number and the elements of the vector. "$\mu_s(x)$" stands for the degree of membership of x to the fuzzy set S.

Negative approach is used. Assume that no fuzzy rule that meets the needs of the theorem is available in the fuzzy control system, that is, at least one of $e_n \notin S_{f_i}$, $e_{cn} \notin S_{f_j}$, $u_n \notin S_{f_k}$ is right.

1) if $e_n \notin S_{f_i}$ then $\mu_{F_i}(x_p) = 0$, so $u' = (0, 0,\cdots,0)$ in Equation (8)

2) if $e_{cn} \notin S_{f_j}$ then $\mu_{F_j}(y_q)$, for the same reason $u' = (0,0,\cdots,0)$ in Equation (8)

3) if $u_n \notin S_{f_k}$ then $\mu_{F_k}(z_r) = 0$, according to Equation (8), we have

$$u' = \min(\mu_{F_i}(x_p),\mu_{F_j}(y_q) \wedge (\mu_{F_k}(z_1),\cdots, \mu_{F_{r-1}}(z_2), \underset{r^{\text{th}}}{0},\mu_{F_{r+1}},\cdots,\mu_{F_k}(z_{N_3}))$$

From 1), 2), 3), we can see that the rth component of the sub-result u', which is derived from the data sample (e_n, e_{cn}, u_n) according to the fuzzy rules that do not meet the needs of the theorem, is constantly zero. Since we have assumed that there is no fuzzy rule which meets the needs of the theorem, the rth component of u', which is the very sum of all the sub-results of the fuzzy reasoning system, is still zero. If we make the fuzzy decision according to the maximum degree of membership, then the fuzzy control system will not output the control value u_n with the input (e_n, e_{cn}). Then the reappearance of data sample is impossible. So the assumption is wrong, that is, the fuzzy rule that meets the needs of the theorem is available, end proof.

For each virtual rule we define a weight to express the possibility the rule can be applied. Originally, all the weights of the virtual rules are set to zero. The goal of the algorithm is to determine the virtual rule that meets the needs of the theorem according to each sample in the sample database and increase the weight value of the virtual rule. Finally, the virtual rule whose weight exceeds the limen set in advance is selected to be the practical rule.

For each data sample, the weights of the virtual rules that meet the needs of the theorem should be corrected. This paper modifies the weights on the basis of the degree of membership of the data samples in the corresponding fuzzy set of the virtual rules the sample defines. This reflects the fuzzification quality of the sample itself and has been proved feasible in practice.

Assume the sample used to extract rules is (e_i, e_{ci}, u_i), $i = 1,2,\cdots, M$. Based on the theorem described above, we give the practical algorithm of fuzzy rule extraction as follows:

1) All the weights of the virtual rules are initialized to zero.

2) Select a sample (e_i, e_{ci}, u_i) from the sample database.

3) The respective supports of e_i, e_{ci}, u_i are deduced from the definition of membership function and support of the 7-level fuzzy linguistic value given above. For convenience, it is best that the numbers of support that e_i, e_{ci}, u_i respectively belong to are one or two. Here we should preferably consider the most complex circumstance, that is, assume there are two supports that e_i, e_{ci}, u_i respectively belong to and are assumed as, S_i^1、S_i^2, $i = 1,2,3$.

4) To each group (i, j, k), i, j, $k = 1$, 2, we calculate $d_{i,j,k} = \mu_{S_1^i} e \wedge \mu_{S_2^j}(e_c) \wedge \mu_{S_3^k}(u)$, then add $d_{i,j,k}$ to the weight of the virtual rule if ($e = S_1{}^i$ and $e_c = S_2^j$) then $u = S_3^k$. Since the supports correspond to fuzzy linguistic value one by one, here for the purpose of simplification, we use the supports to stand for the corresponding fuzzy linguistic values.

5) If there are still samples that have not

been learned, return to 2;

6) Sum up the weight of each virtual rule, get rid of the rules with relatively small weights when contradictory rules are met and select the virtual rules whose weights are greater than the limen set in advance as the practically useful fuzzy control rules.

ADAPTIVE CONTROL

With the help of the fuzzy rules extracting method mentioned above, an adaptive fuzzy control system is built up as shown in Fig.4.

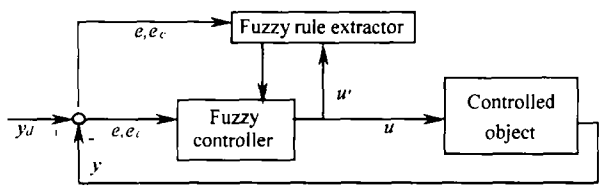

Fig.4 An adaptive fuzzy control system

In Fig.4, y_d is the ideal output, y is the actual output, e, e_c are respectively the error and the change-in-error, u is the output control value of the fuzzy controller. Originally, through the initial data sample we acquire the fuzzy control rules to control the system. In the running process of the system, the error e, change-in-error e_c and the correction u' are sent to the fuzzy rule extractor to learn. To ensure the stability of the system, every time after learning, only the rule with the highest weight is selected. If this rule is previously not available in the fuzzy controller, we get rid of the rules that are contradictory to this rule (if any) in the fuzzy controller, and add the rule to the fuzzy controller. Through the correction of rules like this, we have the system's adaptive control realized.

We can adjust u', the revised value of the output control variable, in several ways. A practical method is to use the following algorithm:

$$u' = u + \alpha e + \beta e_c \tag{9}$$

In which α, β are positive constants, valued according to actual controlled system. The greater the α, β value, the more powerful is the adaptive capacity of the system and the worse is its control stability. In practice, we can assign relatively small values to α, β and gradually increase their values later until the control system becomes not only more stable but also more adaptive.

RESULT OF SIMULATION

The object of simulation is a non-linear inverted pendulum, whose state equations are:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{F - m \cdot g \cdot \sin(x_3) + m \cdot l \cdot \sin(x_3) \cdot x_4^2 \cdot (1 - \cos(x_3))}{M + m \cdot (1 - \cos(x_3))} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \dfrac{(M + m) \cdot g \cdot \sin(x_3) - F \cdot \cos(x_3) + M \cdot l \cdot \sin(x_3) \cdot \cos(x_3) \cdot x_4^2}{M \cdot l \cdot \cos(x_3) + m \cdot l \cdot \cos(x_3) \cdot (1 - \cos(x_3))} \end{cases} \tag{10}$$

In the state equations, (x_1, x_2, x_3, x_4) corresponds to $(r, \dot{r}, \theta, \dot{\theta})$. Through the adjustment of the force F exerted on the trolley on which the inverted pendulum is put, the angle is adjusted. In the simulation experiment, we take the trolley mass $M = 1(\mathrm{kg})$, the inverted pendulum mass $m = 0.2(\mathrm{kg})$, the pendulum length $l = 0.2(\mathrm{m})$, and assume the trolley displacement to be r. Data samples are acquired from the PID controller, whose parameters have been adjusted right and whose control result is relatively good. Taking advantage of the method described in this paper, we obtained the fuzzy control rules after learning the data samples as Table 1.

The step response curve obtained from the controlling operation using the fuzzy control rules shown above is shown in Fig.5.

Table 1 Fuzzy control rules after learning the data samples

	NL	NM	NS	ZE	PS	PM	PL
NL				PM		PS	
NM				PS		PS	
NS			PS	PS	ZE	ZE	
ZE	PL	PM	PS		ZE	NS	NM
PS		PS	ZE		NS		
PM			ZE	NS	NM		
PL				NM			

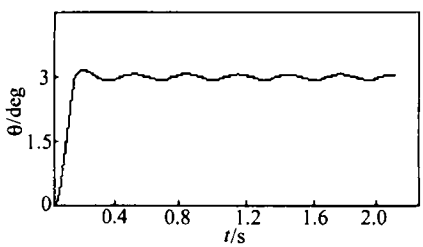

Fig.5 The step response curve

CONCLUSION

The fuzzy control rule extracting method proposed in this paper has the feature of direct and clear principles and simple calculation. The result of simulation shows the correctness of the method proposed in this paper. For controlled objects which require more powerful real-time response capability, we can use the method introduced in this paper to enable rapid rule extraction and adaptive control of the system through the adaptive adjustment of the control rules.

References

Yang Yupu, 1994. Extraction of fuzzy rules by using F – NN and the confidence interval estimation. *Pattern Recognition & Artificial Intelligence*, **7**:53 – 59

Li Gechen, 1997. Afuzzy controller based on new type neural network. Proc. of Fifith Robot Conference of China, Ha'erbing, p.461 – 466.

Keller J.M., 1992. Neural network implementation of fuzzy logic. *Fuzzy Set and System*, **45**(1):1 – 12.

Park Y.M., 1996. An optimal tracking neuro-controller for nonlinear dynamic system. *IEEE Trans on Neural Networks*, **7**(5):1099 – 1110.

Yang Jiangang, 1993. Real time dynamic control of a doubly inverted pendulum using ameliorated CMAC network: The 1st Congress of Post-Doctoral of China, National Defense Industry Press, Beijing, p.358 – 361.

Jiao Licheng, 1993. Application and implementatioin of neural network. Xidian Press, Xi'an, 580 p.