A control system based on the fuzzy neural network for a robot joint

Huailin Zhao[1](zhao_huailin@yahoo.com) Masanori Sugisaka[2]

[1]School of Mechanical and Automation Engineering, Shanghai Institute of Technology, China

[2]Mec. & Ele. Course, Nippon Bunri University, Japan

Abstract: To the robot joint actuated by two McKibben muscles, a new model is supposed, and a control algorithm based on fuzzy CMAC is designed. The new model of the robot joint supposes that there are two independent inputs so that the stiffness control of the robot joint becomes possible. The control algorithm based on both fuzzy logic and CMAC is designed. The fuzzy logic fuzzifys the relationships among the blocks of CMAC so that the performance of the CMAC is improved. At last the simulation is done.

Key word: robot joint, control algorithm, fuzzy CMAC.

1 Introduction

We have done some study on the intelligent control algorithm for the robot joint actuated by a pair of Mckibben muscles[1]. The study aimed only at the joint posture. In another word, the study mainly concerned the angle of the robot joint. But the stiffness of the robot joint is important too in practice. Based on human feeling, to the same joint and the same joint angle, the stiffness can be different. For example, the stiffness should be different when the load changes if the joint wants to keep the same joint angle. So both the angle and the stiffness of the robot joint are important.

In the previous study, we used the robot joint model as Fig.1. Based on the model, the pressure changes of the two Mckibben muscles are related with each other. The pressures of the two Mckibben muscles will change at the same time. In this way we can only control the joint angle. The stiffness of the robot joint can't be adjusted, because the pressures of the two Mckibben muscles can't be adjusted independently. In this study, we improve the control algorithm based on the previous research which only emphasized the joint angle. In the position control, when the joint angle is changed from one value to another one, not only the joint angle should be got to the required value, but also the joint stiffness may be adjusted too. In this study, a special adjusting algorithm of the pressures of the two Mckibben muscles is designed. In the new algorithm, the pressures of the two Mckibben muscles can changes independently and a special constraint condition limits their pressure changes of the two Mckibben muscles when the robot joint works.

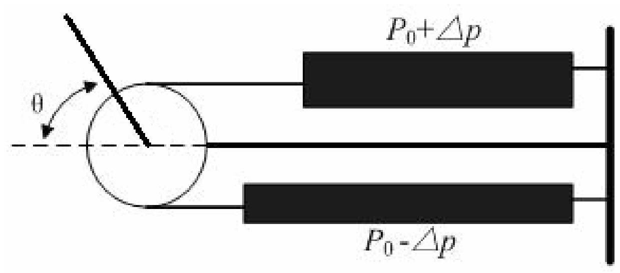


Fig.1 The old model

2 The robot joint actuated by McKibben muscles

In this study, a new robot joint model is designed which is shown in Fig.2. The structure is as a little same as the old one. But the pressures of the two Mckibben muscles change in different way. Both P_1 and P_2 change independently. Based on this model, the angle and the stiffness of the robot joint can be controlled independently.

When the robot joint working, the two Mckiben muscles will be input by two gas pressures P_1 and P_2 which are independent, and they will output pulling forces F_1 and F_2 independently. We use formula (1) to calculate the static forces[2]:

$$F = c_1 P \tfrac{b^2}{4\pi n^2}\{3[(1-c_2 R_c)\tfrac{L}{b}]^2 - 1\} \quad (1)$$

This is a revised force model based on experiments. Where, P is the pressure of the Mckibben muscle, R_c is its contraction rate, L is its current length, b is the length of the fiber in its outer layer, *n* is the number of winding turns of a single fiber, and c_1 and c_2 are two constants decided by the experiments.

Then we establish the dynamic model as the following:

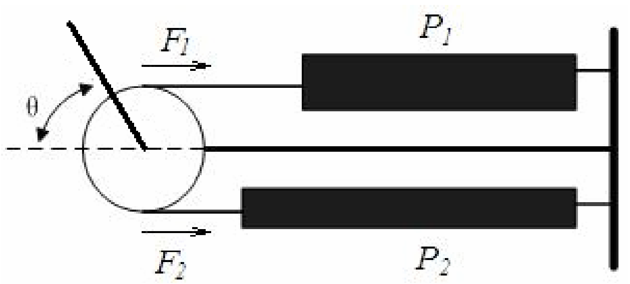
$$M = J\frac{d^2\omega}{dt^2} \tag{2}$$

$$M = M_s - M_v \tag{3}$$

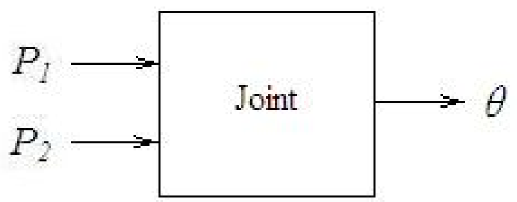
$$M_s = R(F_1 - F_2) \tag{4}$$

$$M_v = c(P_1 + P_2)\frac{d\omega}{dt} \tag{5}$$

Where, M is the total moment of the robot joint, M_s is its static moment, M_v the moment item related with joint viscous damping, J is the rotation inertia, *R* is the radius of the joint cylinder, and c is the viscous damping factor.



(a)



(b)

Fig.2 The new model of the robot Joint

3 The control system based on the fuzzy neural networks

In previous studies, we designed the control algorithms based on the neural networks, which including the pure CMAC network and the compound control algorithm based on both CMAC and PID. To those algorithms, there are some problems not resolved. One problem is that the pure neural network controller is not absolutely reliable. Another one is referred to the compound control algorithm. The compound controller consists of a CMAC controller and a PID controller which are parallel to each other. The problem is that we must determine the PID parameters before the controller working. The previous study concluded that the PID parameter values should be within a proper field. So we must try the PID parameters before the control system begins to work. And there is a risk that the compound control system may fail if the robot joint or its load changes too much. Summarizing the above analysis, we design a new control algorithm based on both fuzzy logic and CMAC networks, shown in Fig.3. This algorithm can solve the two problems at the same time.

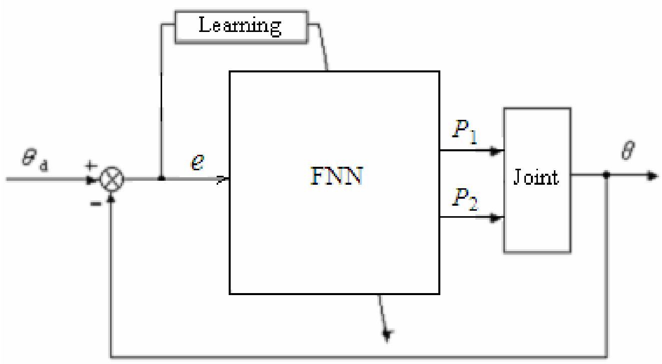


Fig.3 The fuzzy neural network control system

In Fig.3, θ_d is the desired angle of the robot joint, θ is the real angle, *e* is the difference between θ_d and θ, and FNN is the controller.

To a same robot joint angle, there are numberless pairs of P_1 and P_2 corresponding to it. So here we add a constraint to them, which is the formula (6).

$$(\Delta P_1)^2 + (\Delta P_2)^2 = \min \tag{6}$$

Where, ΔP_1 and ΔP_2 are the changes of P_1 and P_2 respectively.

The structure of the fuzzy neural network controller is shown in Fig.4.

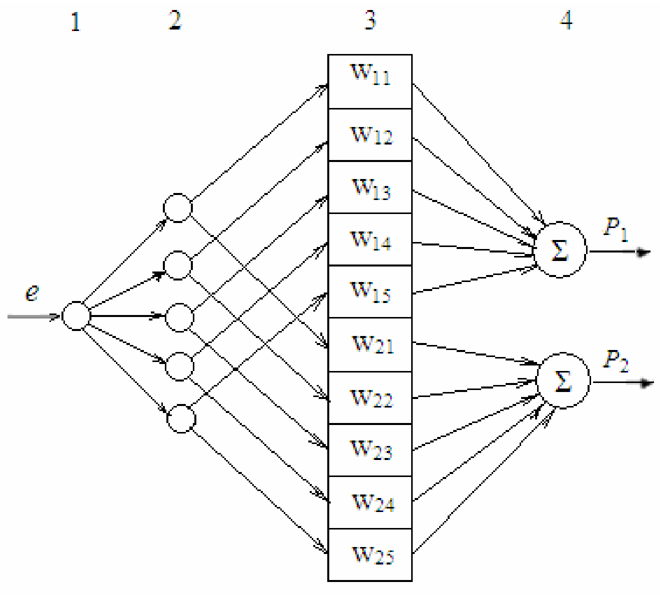


Fig.4 The fuzzy neural network controller

In Fig.4, we divide the controller into 4 layers. Layer 1 introduces the input of the controller. Its output and input has the relationship as formula (7).

$$O^{(1)} = I^{(1)} = e \qquad (7)$$

Where, $O^{(1)}$ and $I^{(1)}$ are respectively the output and input of the layer 1.

In layer 2, the input of the controller is fuzzified. Here we divide the whole input space into 5 blocks and define 5 Gaussian functions as the membership functions of the 5 blocks:

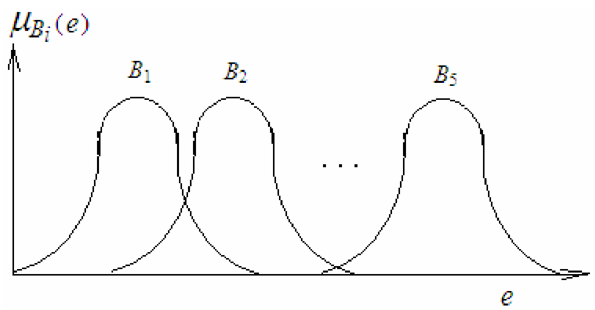


Fig.5 The Gaussian functions and their distribution

$$\mu_{B_i}(e) = e^{-\left(\frac{e-\sigma_i}{\upsilon_i}\right)^2} \qquad \text{(i=1,2,3,4,5)} \qquad (8)$$

Therefore, the relationship between the input and the output in layer 2 should be:

$$O_i^{(2)} = I_i^{(2)} = \mu_{B_i}(e) \qquad \text{(i=1,2,3,4,5)} \qquad (9)$$

In the layer 3, the output of the layer 2 activates the associative intensity in the associative units, and the output is calculated by the formula (8).

$$O_{ji}^{(3)} = O_i^{(2)} \times w_{ji} = \mu_{B_i}(e) \times w_{ji}$$
$$\text{(i=1,2,3,4,5,} \quad \text{j=1,2)} \qquad (10)$$

The layer 4 sums the outputs of the associative units and then calculates the output of the controller. The calculation is the formula (11).

$$P_j^{(4)} = \sum_{i=1}^{5} O_{ji}^{(3)} = \sum_{i=1}^{5}(O_i^{(2)} \times w_{ji}) = \sum_{i=1}^{5}(\mu_{B_i}(e) \times w_{ji})$$
$$\text{(i=1,2,3,4,5,} \quad \text{j=1,2)} \qquad (11)$$

4 The leaning algorithm of the fuzzy CMAC controller

The learning of the controller should include the associative intensity w_{ji}, central value σ_i and width υ_i of the Gaussian function parameters. Here we suppose P_d and P are the desired and real outputs of the controller, and define the objective function of the error as the formula (12). And the network learns by δ algorithm[3].

$$E = \tfrac{1}{2}(P_d - P)^2 \qquad (12)$$

$$\Delta w_{ji}(k) = -\eta_1 \frac{\partial E}{\partial w_{ji}} = -\eta_1 \frac{\partial E}{\partial P} \bullet \frac{\partial P}{\partial w_{ji}} = \eta_1 (P_d - P) O_i^{(2)}$$
$$\text{(i=1,2,3,4,5,} \quad \text{j=1,2)} \qquad (13)$$

$$\Delta \sigma_i(k) = -\eta_2 \sum_{i=1}^{5} \left[\frac{\partial E}{\partial P} \bullet \frac{\partial P}{\partial O_i^{(2)}} \bullet \frac{\partial O_i^{(2)}}{\partial \sigma_i}\right]$$

$$= 2\eta_2 (P_d - P) \sum_{i=1}^{5} \left[O_{ji}^{(3)} \bullet \frac{e-\sigma_i}{\upsilon_i^2}\right]$$
$$\text{(i=1,2,3,4,5,} \quad \text{j=1,2)} \qquad (14)$$

$$\Delta\upsilon_i(k) = -\eta_3 \sum_{i=1}^{5}[\frac{\partial E}{\partial P} \bullet \frac{\partial P}{\partial O_i^{(2)}} \bullet \frac{\partial O_i^{(2)}}{\partial \upsilon_i}]$$

$$= 2\eta_3(P_d - P)\sum_{i=1}^{5}[O_{ji}^{(3)} \bullet \frac{(e-\sigma_i)^2}{\upsilon_i^3}]$$

(i=1,2,3,4,5, j=1,2) (15)

Where, η_1, η_2 and η_3 are the learning rates, and each of them should belong to (0,1). Therefore the three parameters should be calculated as

$$w_{ji}(k+1) = w_{ji}(k) + \Delta w_{ji}(k)$$

(i=1,2,3,4,5, j=1,2) (16)

$$\sigma_i(k+1) = \sigma_i(k) + \Delta\sigma_i(k)$$

(i=1,2,3,4,5) (17)

$$\upsilon_i(k+1) = \upsilon_i(k) + \Delta\upsilon_i(k)$$

(i=1,2,3,4,5) (18)

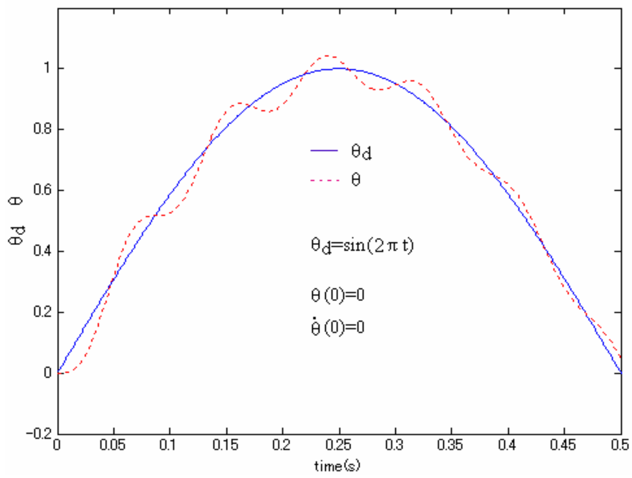
5 Simulation

Based on both the model of the robot joint and the control algorithm, we simulate the control system by MATLAB. There is no load, the initial conditions are that both $\theta(0)$ and $\omega(0)$ are zero. The desired track of the robot joint $\theta_d(t)$=sin($2\pi t$). The sampling time is 0.001 second. And we add a disturbance item to the joint which is M_d=sin($10\pi t$).

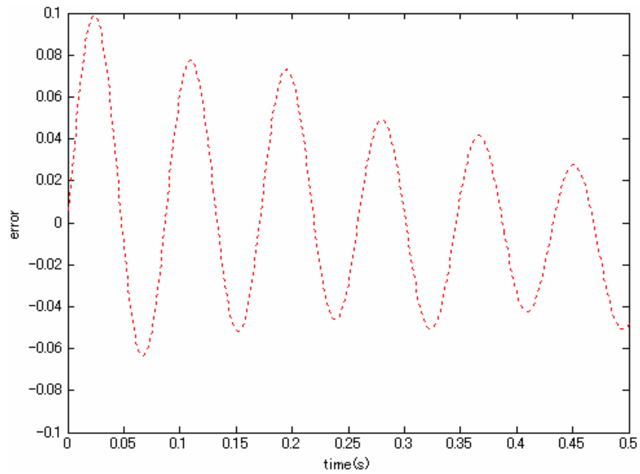
The simulation results are shown in Fig.6, which indicates that the control system is stable and it can resist the outer disturbance.

6 Conclusion

To the nonlinear system like the robot joint actuated by the McKibben muscles, a new model with two inputs and one output is supposed. The changes of the pressures of the two Mckibben muscles are independent. It is possible to realize the stiffness control based on the model. The control algorithm based on Both fuzzy logic and CMAC is designed. The simulation shows that the control algorithm is effective and the control system is stable. The control system can resist the outer disturbance too.



(a) The tracking lines



(b) The tracking error

Fig.6 Tracking and error of the control system

Acknowledgement

This study is supported by the project from Shanghai Institute of Technology, China (YJ2008-12).

Reference:

[1] Huailin Zhao, Masanori Sugisaka, Simulation study of CMAC control for the robot joint actuated by McKibben muscles, Applied Mathematics and Computation, 203(2008), pp457-462

[2] Huailin Zhao, Masanori Sugisaka, A Model of the Mckibben muscle actuator based on experiments, Systems science, Vol. 34(2008), No. 2, pp83 - 88

[3] Yaonan Wang, Intelligent control engineering for robots, Beijing: Science Publishing House, 2004