

Design and Experimental Evaluation of a Human Skill-Based PID Controller

Yuntao Liao*

*Graduate School of Engineering, Hiroshima University,
Hiroshima Japan*

Toru Yamamoto

*Faculty of Engineering, Hiroshima University
Hiroshima Japan*

*E-mail: liao-yuntao@hiroshima-u.ac.jp, yama@hiroshima-u.ac.jp
www.hirosima-u.ac.jp*

Abstract

Nowadays less and less people are willing to do hard works. Which makes that the controller based on human skill is needed. Cerebellar model articulation controller (CMAC) is a kind of neural networks (NNs) that can easily solve problems of nonlinear system. Compared with other NNs the advantage of CMAC is that it takes shorter learning time. Moreover because of the widely used of PID controller a human skill-based PID controller using CMACs has been proposed in this paper.

Keywords: Cerebellar model articulation controller, Neural network, Human skill, PID controller.

1. Introduction

In the last decades, the automatic controller has been widely used in many fields. But still there are a lot of fields that human skill cannot be taken place by automatic controllers. However, the skilled workers become less and less due to the change of modern thoughts. As the result these specific skills will not be taken to the next generations. Therefore, to develop human skill based controller is necessary.

Human skill is a kind of nonlinear system. In order to deal with such nonlinear system, it is effective that some neural networks (NNs) are utilized. So far many NNs have been proposed and parts of them have been successfully used to design controllers¹⁻⁴. But there is a problem that the conventional NNs need large learning

time to adjust weights of neurons, which makes it impossible for NNs to be applied widely. However the appearance of cerebellar model articulation controller (CMAC) that was proposed at 1975 by Albus partly solves this problem. CMAC has simple structure and like other NNs weights of CMAC also need to be adjusted, but this kind of adjustment has generalization ability. This kind of ability brings CMAC good real-time performance and some CMAC controllers have already been proposed⁵. Due to the above reasons using CMAC to design a controller based on human skill is feasible.

Furthermore PID controller has been widely used in various control systems. And the experienced human experts have been inevitably required in tuning PID gains. For the above proposal a human-skill based PID controller using CAMCs have been proposed.

2. Design of a Human Skill-Based Controller

2.1. CMAC

The following figure is a sample of simplified 2-dimensional input CMAC model to explain the structure of CMAC.

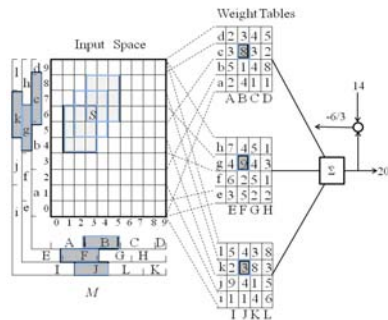


Fig. 1. 2-dimensional input CMAC model

In Fig.1, it is a CMAC with 3 weight tables and 4 labels. The set of input signals S (3,6) is mapped to B, F, J and c, g, k as set of labels M. Then, from each weight table 8,9 and 3 are selected. The sum of the selected weights is 20. Assume that the wished output value is 14, there will be a $-6/3$ added to all referred weights in the tables because of the difference between obtained value and wished value is -6 and the number of selected weight tables is 3. As explained in the example CMAC usually works as follows: First the set of input signals S is mapped to the set of labels M. Then, according to the set of tables weights will be selected from each table. At last, output signals will be performed as the sum of the weights. From the explanation, when CMAC updating weights for one input signal the weights for the nearest input signals are also being updated, this kind of ability makes CMAC can obtain approximately wished value in shorter learning time than conventional NNs.

2.2. Controller design

In this paper, the following PID structure is considered:

$$\begin{aligned}
 u(t) = & u(t-1) \\
 & + K_p (y(t-1) - y(t)) \\
 & + K_i e(t) \\
 & + K_d (2y(t-1) - y(t) - y(t-2))
 \end{aligned} \tag{1}$$

where K_p , K_i and K_d denote the proportional gain, the integral gain and the differential gain. Moreover $r(t)$ denotes the target value and $e(t)$ means the control error signal. The block diagram that shows the process how proposed method works is Fig.2.

© The 2015 International Conference on Artificial Life and Robotics (ICAROB 2015), Jan. 10-12, Oita, Japan

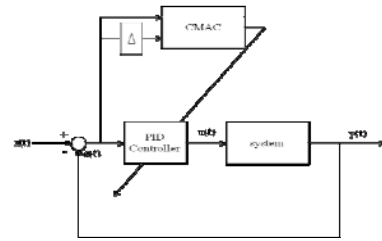


Fig.2. Block diagram of proposed method

In figure human skill data has already been learned by the CMAC and PID gains that tuned by these CMACs are used to control the system. The CMAC here sets $e(t)$ and $\Delta e(t)$ as the input space to calculate weights. And $\Delta e(t)$ is the signal defined as:

$$\Delta e(t) := e(t) - e(t-1) \tag{2}$$

PID parameters are tuned by CMACs according to the following equations:

$$\begin{aligned}
 K_p &= \sum_{h=1}^K W_{p,h}(t) \\
 K_i &= \sum_{h=1}^K W_{i,h}(t) \\
 K_d &= \sum_{h=1}^K W_{d,h}(t)
 \end{aligned} \tag{3}$$

Where $h=1,2,\dots,K$ and K denotes the total number of the selected weights in CMAC. From above the human skill based controller is designed through PID gains. And the performance of the controller is strongly depended on these PID gains.

Then the process of how CMACs learn human-skill data is shown in Fig.3.

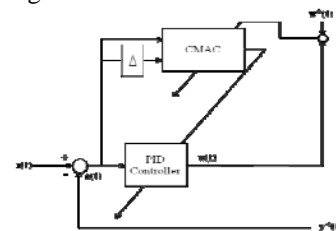


Fig.3. Block diagram of learning process

From Fig.3 there are two signals that attached asterisk (*) which means they are human-skill data. And the human-skilled signals are set as the teacher signals. The learned result $u(t)$ approaches teacher signal $u^*(t)$ by updating weights $W_h(t)$. The CMAC weight tables are modified by steepest descent method and shown in Eq.4. In the equation $g(t)$ means the gradient that used to update weights, J means the error criterion. And they are expressed as Eq.5 and Eq.6. In Eq.5 a , b and c are the appropriate positive constants set by designer.

$$W_{p,h}^{new}(t) = W_{p,h}^{old}(t) - g(t) \frac{\partial J}{\partial K_p} \frac{1}{K}$$

$$W_{i,h}^{new}(t) = W_{i,h}^{old}(t) - g(t) \frac{\partial J}{\partial K_i} \frac{1}{K}$$

$$W_{d,h}^{new}(t) = W_{d,h}^{old}(t) - g(t) \frac{\partial J}{\partial K_d} \frac{1}{K}$$
(4)

$$g(t) = \frac{1}{c + a * \exp(-b |u^*(t) - u(t)|)}$$
(5)

$$J := \frac{1}{2} \varepsilon(t)^2$$
(6)

$$\varepsilon = (u^*(t) - u(t))$$
(7)

Moreover, the partial differential of Eq. (4) is developed as follows:

$$\begin{aligned} \frac{\partial J}{\partial K_p} &= \frac{\partial J}{\partial \varepsilon(t)} \frac{\partial \varepsilon(t)}{\partial u(t)} \frac{\partial u(t)}{\partial K_p} \\ &= \varepsilon(t)(y(t) - y(t-1)) \\ \frac{\partial J}{\partial K_i} &= \frac{\partial J}{\partial \varepsilon(t)} \frac{\partial \varepsilon(t)}{\partial u(t)} \frac{\partial u(t)}{\partial K_i} \\ &= -\varepsilon(t)e(t) \\ \frac{\partial J}{\partial K_d} &= \frac{\partial J}{\partial \varepsilon(t)} \frac{\partial \varepsilon(t)}{\partial u(t)} \frac{\partial u(t)}{\partial K_d} \\ &= \varepsilon(t)(y(t) - 2y(t-1) + y(t-2)) \end{aligned}$$
(8)

3. Experimental Evaluation

3.1. Experimental Equipment

In order to illustrate the effectiveness of the proposed method, an experiment has been performed by using a beam and ball control model. The experiment equipment is shown in Fig.4. As the figure shows, there is an input manipulator for human to operate. It is set to record the data of human skill. Also there is a distance sensor on beam to read output data of the system. Moreover there is an Arduino board to transfer data between computer and system. And the ball on the beam is the target to be controlled.

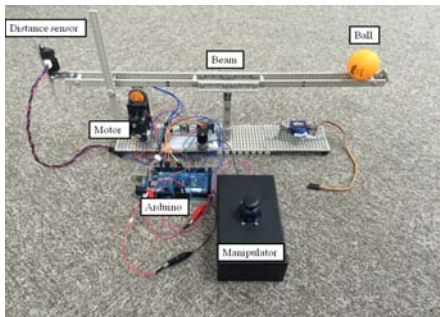


Fig. 4. Beam and Ball control Equipment

3.2. Simulation and Experiment Result

The aim of the experiment is to control the ball staying at the middle of the beam, where $r=0$. Position of ball will be considered as output y and in this experiment the initial y is 20, it means at the beginning the ball stays at the right side of the beam. And angle of the beam is considered as the input of the system and the initial input is 0. The limit of u is set as:

$$-45^\circ \leq -u(t) \leq 45^\circ$$

And the output y is limited as:

$$-20[cm] \leq y(t) \leq 20[cm]$$

Human skill data u^* and y^* are obtained by skilled operator and they are shown in Fig.5. And based on the human skill data the learning process of CMACs is performed. In this design the CMACs with 4 weight tables and 3 labels are chosen. And the coefficients of learning ratio are set to be $a=10^2$, $b=10^2$, $c=10^2$. In addition, the initial values of the weights were fixed PID gains obtained by pole placement method based on the model of the system. And these PID gains are $K_p = 0.0056$, $K_i = 0.0019$, $K_d = 0.0045$.

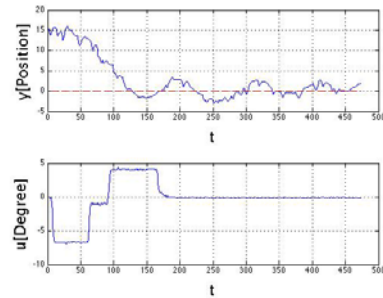


Fig.5. Control result by skilled operator

And the performance of learning process is evaluated by the following integrated squared error (ISE) equation:

$$ISE = \sum_{t=1}^{30} (u^*(t) - u(t))$$
(9)

As Fig.6 shows, the learning process stops when the integrated squared error is less than 0.00056.

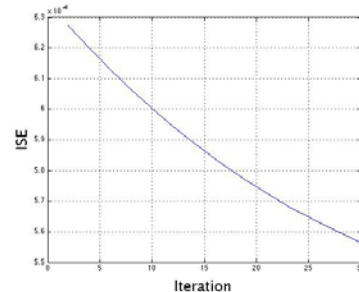


Fig.6. Performance of learning process

Fig.7 shows the comparison between learning result u and teacher signal u^* , where 30 iterations were performed. Fig.8 shows the PID gains trajectories corresponding to Fig.7. By using the PID gains that shown in Fig.8 an experiment has been done and the result is shown in Fig.9. From figure y is close to y^* shows that by using proposed method human skill can be learned.

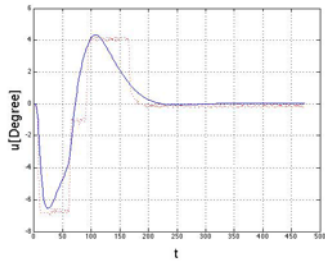


Fig.7. Comparison between u and u^*

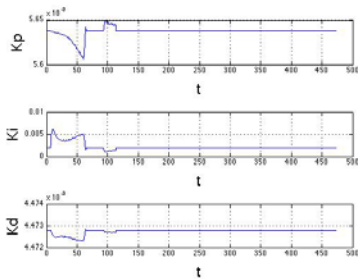


Fig.8. PID gains corresponding to Fig.7

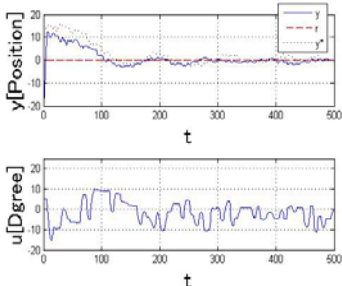


Fig.9. Control result by proposed method

3.3. Human Skill Analysis

Fig.13 shows the trajectories of kc , Ti and Td they are got by following equations:

$$kc = Kp \quad Ti = \frac{kc}{Ki} Ts \quad Td = \frac{Kd}{kc} Ts \quad (10)$$

where Ts is sampling time and $Ts=0.02$. From the figure human skill is analyzed as follows: At first kc and Ti become smaller and Td becomes bigger because manipulator tries to move the ball to target stably. And then kc and Ti becomes bigger and Td becomes smaller shows human thinks the ball is going to reach the target

and he is trying to stop moving the ball. Next kc and Ti become smaller again and Td becomes bigger because his prediction was wrong the ball is not reach the target.

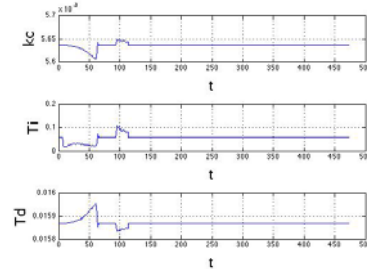


Fig.13. Trajectories of PID parameters

Then kc and Ti becomes bigger and Td becomes smaller because the manipulator predicts the ball is going to reach the target again and after that the change of kc , Ti and Td is not obvious shows human is maintain his input to move the ball to target. At last, kc and Ti become smaller and Td becomes bigger to stop the ball at the target after that there is no change of kc , Ti and Td shows human thinks the ball stops at the target. From analysis, the manipulator wants to control the ball stably approach to the target without overshoot.

4. Conclusions

In this paper, a human skill based PID controller using CMAC has been proposed. And the effectiveness of method has been experimentally examined on a beam and ball model. According to the control results the proposed method can replace human skill and based on the PID parameters human skill can be analyzed.

5. References

1. S.Liu and H.Asada "Adaptive Control of Deburring Robots Based on Human Skill Models," *Proc.of IEEE Conf. on Decision and Control, Brighton*, pp.348-353,1991.
2. K.S.Narendra and K.Parthasarathy "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans.on Neural Networks*, Vol.1,No.1,pp.4-27,1990.
3. S.Omatu, K.Marzuki and Y.Rubiyah "Neuro-Control and Its Applications," *Springer-Verlag*, 1995.
4. M.Kato, T.Yamamoto and S.Fujisawa "A Skill-Based PID Controller Using Artificial Neural Networks," *Proc.of Int. Conf. on Computational Intelligence for Modeling, Control and Automation, Vienna*, pp.702-705,2005.
5. K.Koiwai, K.Kawada and T.Yamamoto, "Design of a Human-Skill Based PID Controller using CMACs," *Int. Conf. on Control, Automation and Systems*, 2008