Design of Hierarchical Clustering CMAC Based PID Controllers

（階層型クラスタリング小脳演算モデルを

用いた制御システムの設計）

Graduate School of Engineering
Hiroshima University

Yuntao Liao

March 2018

Acknowledgment

My deepest gratitude goes first and foremost to Professor Toru Yamamoto, my supervisor. His guidance, encouragement and his rigorous attitude of research will influence me for my entire life.

Besides my supervisor, I would like to thank Professor Toshio Tsuji, Professor Ichiro Nishizaki, Professor Shuichi Ohno, Professor Masayoshi Nakamoto and Professor Shin Wakitani, for their insightful comments and helps.

I also want to show my great appreciation to Professor Kazushige Koiwai, for his continuous support of my research, for his patience and motivation.

Additionally, my gratitude goes to all the past and present members of the control systems engineering laboratory. Especially, I want to show my acknowledgment to Dr. Takuya Kinoshita, Mr. Guan Zhe and Mr. Rizwan Tariq, for their continuous support in my five years' laboratory life.

Finally, I would like to express my sincere thanks to my parents, for their supporting me spiritually throughout my writing of this thesis and five years' research life. I also would like to express my gratitude to my girlfriend, for her enlightening me when I facing difficulties in my research and life.

Contents

Chapter 1

Introduction

1.1 Background

The control theories develop fast these years, especially, some control theories aim to perform a control process by replacing workers and some control theories borrow ideas from biological systems become applicable. The control theories that achieve their control ability by emulating biological intelligence can be defined as ”intelligent control” [1]. Intelligent control contains many areas, main of them are fuzzy control, artificial neural networks, expert and planning system, and genetic algorithms. This dissertation will explain some intelligent controllers that developed based on a kind of artificial neural networks (ANN) in detail.

In reference [2], ANN is defined as ”A neural network is an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the interunit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns.”

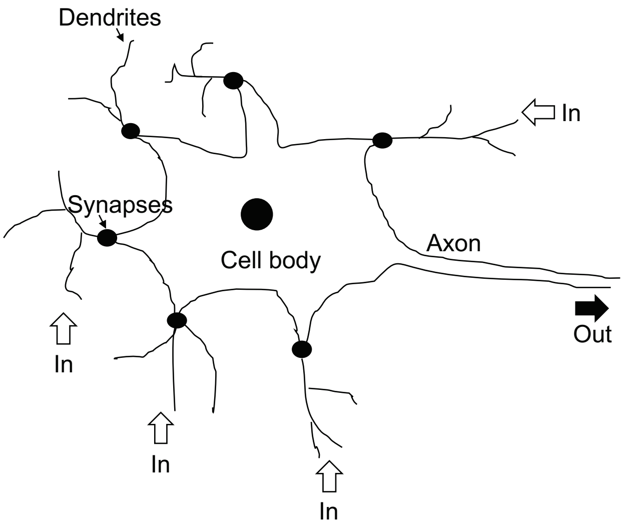

Figure 1.1: A simple structure of a neuron

From the definition, it is known that the ANNs are inspired by neurons. To understand the basic structure of ANNs, some simple neurobiology are explained according to Fig. 1.1.

In human brain, there are almost 10^{11} nerve cells or neurons [3]. There are about 10^{15} connections between these neurons, these connections are called synapses, the branches of neurons are referred to as dendrites. For a neuron, it receives thousands of incoming signals from other neurons, all these signals are finally reach cell body. All these signals are then integrated together in some way, and if the resulting signal exceeds some threshold, the neuron will be activated and its signal will be transferred to other neurons through axon.

Based on the above explanation, a single-input neuron, a multiple-input neuron and a multiple-layer network can be constructed [4] [5] [6]. A figure shows the structure of single-input neuron is given in Fig. 1.2. The input u is multiplied by *weight* w, in the *summer*, wu is added with *bias* b, through a

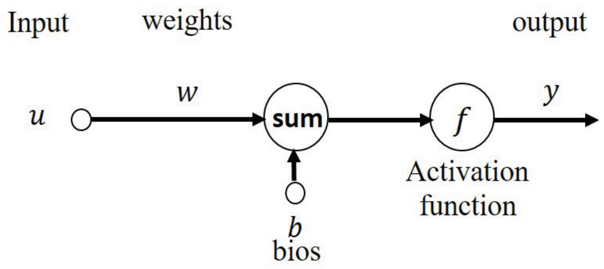

Figure 1.2: Structure of a single-input neuron

activation function the output value y can be calculated. y can be expressed as:

$$y = f(wu + b). \tag{1.1}$$

The *activation function* is decided by the users, particularly, log-sigmoid function which is given in (1.2) is a commonly used *activation function*.

$$\frac{1}{1+e^{-n}}, \tag{1.2}$$

where, n represents the order. This function takes the input into the range 0 to 1.

Multiple-input neuron is an extension of single-input neuron, it has more than one input. Each input is weighted by the corresponding weight, Fig. 1.3 shows the structure of a multiple-input neuron. The output of the neuron can be expressed as:

$$y = f(\boldsymbol{WU} + b), \tag{1.3}$$

where, $\boldsymbol{W}$ and $\boldsymbol{U}$ are as follows:

$$\boldsymbol{W} = [w_1, w_2, \cdots, w_n], \tag{1.4}$$

$$\boldsymbol{U} = [u_1, u_2, \cdots, u_n]^T. \tag{1.5}$$

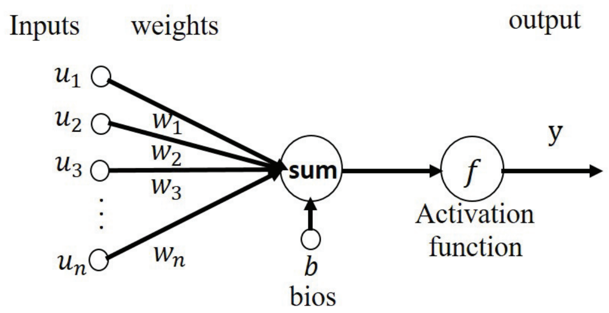

Figure 1.3: Structure of a multiple-input neuron

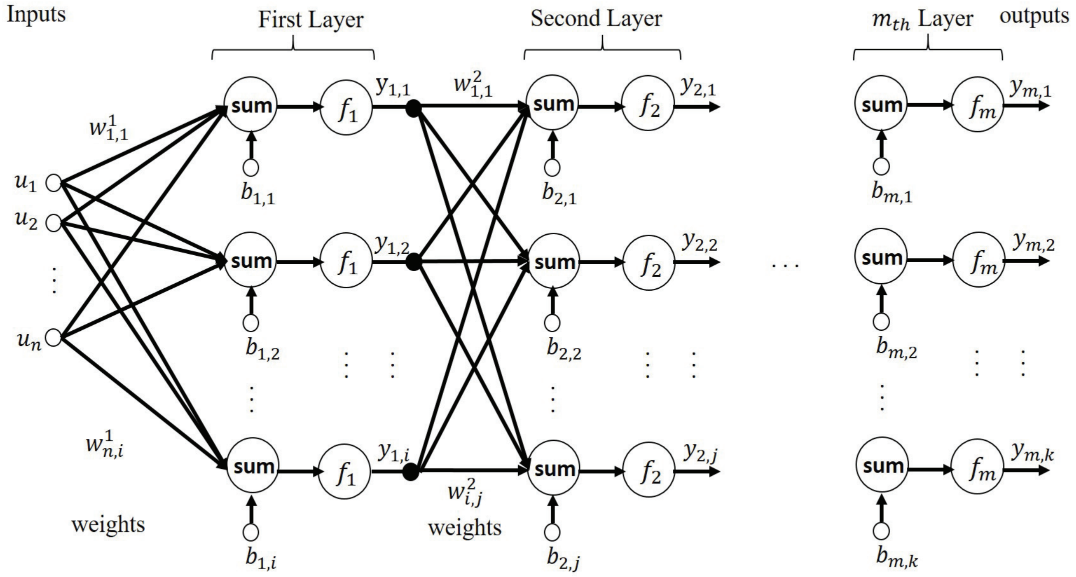

Figure 1.4: Structure of a multiple-layered neural network

A multiple-layered network is a neural network with several layers. Each layer has its weights, bias and activation function. As an example, a structure of a multiple-layer neural network is shown in Fig. 1.4. The number of neurons for each layer can be different, in this example, number of neurons for first layer, second layer and m_{th} layer are i, j and k. The weights of each layer can be expressed as a weight matrix, for the first layer, the weight matrix can be as:

$$\boldsymbol{W}_1 = \begin{bmatrix} w^1_{1,1} & w^1_{1,2} & \cdots & w^1_{1,i} \\ w^1_{2,1} & w^1_{2,2} & \cdots & w^1_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ w^1_{n,1} & w^1_{n,2} & \cdots & w^1_{n,i} \end{bmatrix}. \tag{1.6}$$

Bias can be expressed as a bias vector:

$$\boldsymbol{B}_1 = [b_{1,1}, b_{1,2}, \cdots, b_{1,i}]^T. \tag{1.7}$$

The output vector $\boldsymbol{y}$ of this multiple-layer neural network can be calculated according to the following equation:

$$\boldsymbol{y} = f_m(\boldsymbol{W_m} \cdots f_2(\boldsymbol{W_2} f_1(\boldsymbol{W_1 U} + \boldsymbol{B_1}) + \boldsymbol{B_2}) \cdots + \boldsymbol{B_m}). \tag{1.8}$$

The basic structure of ANNs is introduced from the above statement. Based on such structure, the ANNs achieve their functions through training of weights. However, for the conventional ANNs, there is a shortcoming that limits the application of them, the training of weights costs too much time.

1.2 Cerebellar Model Articulation Controller

Cerebellar model articulation controller (CMAC) is a kind of ANNs, it is utilized in modeling, classification and control fields; in control field, it is a popular choice, because it has a shorter learning time than other ANNs [7] [8] [9] [10] [11] [12].

CMAC is inspired by cerebellum, it proposed by Albus [13] [14] [15]. Albus proposed a cerebellum model, it is given in Fig. 1.5. In the figure, P denotes the *Purkinje cell*, S represents the *stellate b cell* and B stands for the *basket cell* [16]. For cerebellum, mossy fibers are one of the major inputs. Informations are transfered from mossy fibers to the granule cells, through the parallel fibers, finally they reach the Purkinje cells for processing [17].

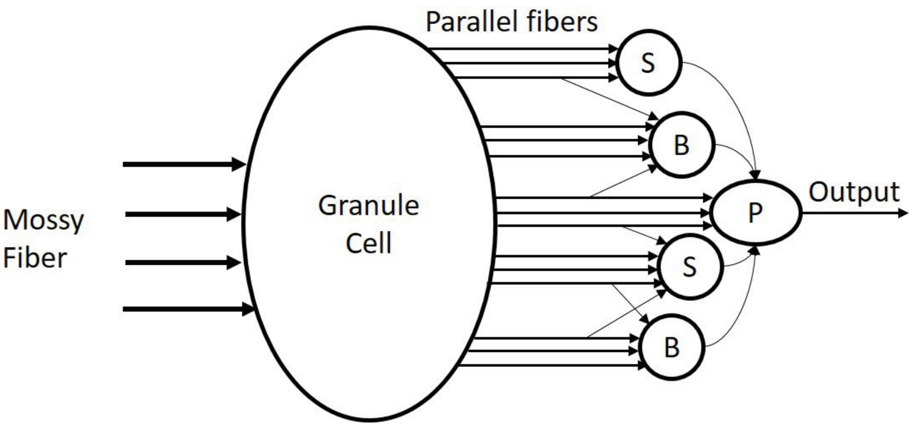

Figure 1.5: Albus' cerebellum model

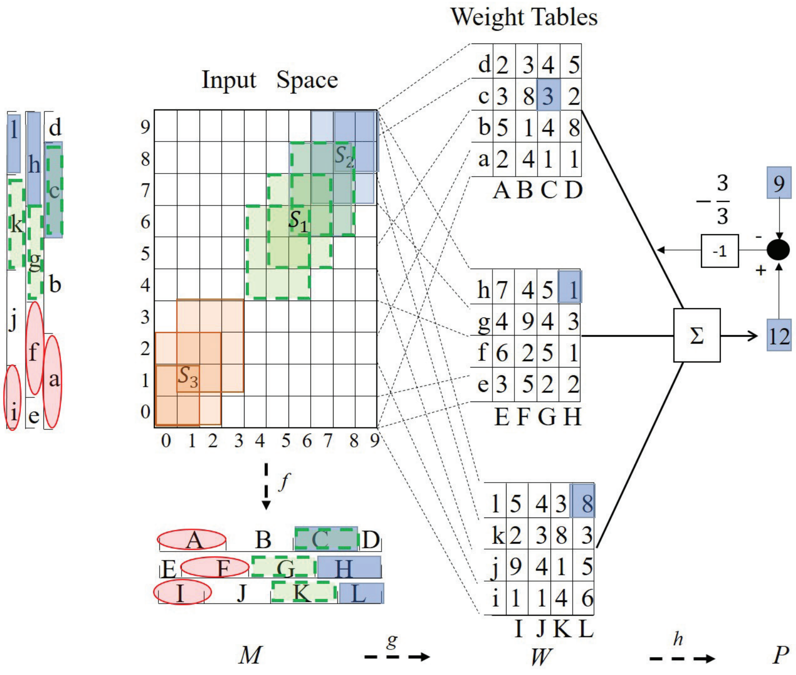

Figure 1.6: Two-dimensional CMAC structure

CMAC can be treated as a look-up table with partial learning ability and generalization ability, based on an example of CMAC which is shown in Fig. 1.6, these abilities are explained in detail. In the figure, S denotes the *input signals*, M represents the *Memory Cells*, W stands for the *Weight Tables* (actual memory) and P is the *response output*. The above structure can be formalized as:

$$\begin{cases} f : S \to M \\ g : M \to W \\ h : W \to P. \end{cases}$$

The partial learning ability and generalization ability of CMAC will be introduced based on Fig. 1.6. In the figure, there are three input signals, the learning process is going to be stated based on S_2, the partial learning ability of CMAC is going to be explained based on input signals S_2 and S_3, and the generalization ability of CMAC is going to be introduced according to input signals S_1 and S_2.

The weights of CAMC are updated through its learning process, from Fig. 1.6, S_2 is mapped to $[C, c]$, $[H, h]$ and $[L, l]$ in M, and the weights 3, 1 and 8 are select in A, the output of the system is sum of selected weights as: $\sum_{i=1}^{N} W_i$, in this equation, W_i denotes the selected weights. In this example, the output is 12, the reference value (R_v) is 9, compare with the output value (O_v) 12, the difference is 3, the number of weight tables (N_w) is 3, the selected weights are updated according to the following equation:

$$W_i^{new} = W_i^{old} - \frac{O_v - R_v}{N_w}, \tag{1.9}$$

therefore, there is a $-\frac{3}{3}$ added to the selected weights.

The partial learning ability is a feature of CMAC, in Fig. 1.6, S_2 and S_3 are two input signals, S_2 activates $[C, c]$, $[H, h]$ and $[L, l]$ and S_3 activates $[A, a]$, $[F, f]$ and $[I, i]$, on each coordinate, the projections of two input signals are dissimilar, thus, they are using different parts of CMAC and their updating of weights do not influence each other. This kind of ability is the partial learning ability of CMAC.

The generalization ability is another feature of CMAC, this ability results CMAC a fast learning ability. In Fig. 1.6, S_2 and S_1 are similar input signals, for S_2, $[C, c]$, $[H, h]$ and $[L, l]$ are selected, for S_1, $[C, c]$, $[G, g]$ and $[K, k]$ are chosen, they have a common activated unit $[C, c]$, that is to say, when the weights for S_2 are updated, part of S_1's weights is also being updated, this kind of ability is local generalization ability of CMAC.

However, CMAC has its drawbacks, such as the requirement of memory increases exponentially and the generalization ability decreases, as the learning accuracy increases. Hence, an optimized structure named as hierarchical-clustering(HC) CMAC is proposed.

In this dissertation, HC-CMAC is utilized as a "controller parameter tuner" for all the proposed schemes, the structure of HC-CMAC is going to be explained in detail in Chapter 2 [18][19][51].

1.3 Dissertation Outline

The thesis consists of five chapters, they are roughly introduced as follows.

In Chapter 2, a HC-CMAC is proposed and a HC-CMAC based PID controller is introduced.

The HC-CMAC focus on the improvement of conventional CMAC's shortcomings. In the conventional CMAC, as the learning accuracy increases the requirement of memory increases exponentially, simultaneously, the generalization ability decreases. To overcome such disadvantages, a HC-CMAC is proposed. The HC-CMAC can balance the requirement of memory, learning accuracy and generalization ability of a CMAC network. In this chapter, a PID controller that its gains are tuned on-line by using a HC-CMAC is designed. The effectiveness of the controller is verified through some simulations and experiments, the comparison between the proposed method and conventional CMAC based PID controller is also demonstrated.

In Chapter 3, a HC-CMAC PID controller using closed-loop data is designed.

In the proposed method, the fictitious reference iterative tuning (FRIT) it utilized to enable the controller gains are tuned in an off-line manner. By combining FRIT with HC-CMAC, the tuning of controller gains achieves in an off-line manner, and model of control objective is unnecessary. The advantage of the controller is shown by comparing the proposed method with a previous study through both simulations and experiments.

In Chapter 4, a HC-CMAC Performance Driven(PD) PID controller is proposed.

The controller improves control performances for transient state and steady state. For transient state, the controller gains are tuned to minimize the difference between fictitious output and output of closed-loop data. For steady state, a Minimum Variance Control based index (MV index) “Control Performance Assessment (CPA)” is proposed, and the controller gains are tuned to minimize the difference between CPA and fictitious CPA. Since then, the controller gains that improve control performance for both transient and steady state are learned. The effectiveness of the controller is shown by some simulation examples and experimental results, some comparisons between the proposed method and some previous works are also given.

In Chapter 5, the conclusion of the dissertation is stated.

Chapter 2

Design of Hierarchical Clustering(HC) CMAC Based Controllers

2.1 Introduction

As explained in the introduction of this thesis, cerebellar model articulation controller (CMAC) is a type of ANNs. Comparing CMAC with conventional ANNs, the CMAC and has a faster learning speed, since its local generalization ability. Thus, CMAC network is utilized to optimize control performance in some studies [20] [21] [22]. In the previous studies, CMAC is not only utilized to generate system input but also it is applied to tune controller parameters as a 'controller parameter tuner'. Furthermore, when CMAC is selected as a controller parameter "tuner", if the controller includes an integral element, for untrained signals, there exists an improvement of control performance, due to its generalization ability. Treating CMAC as a PID controller "tuner", some methods are proposed [23] [24] [25] [26] [27].

However, CMAC has its shortcoming. When a high learning accuracy

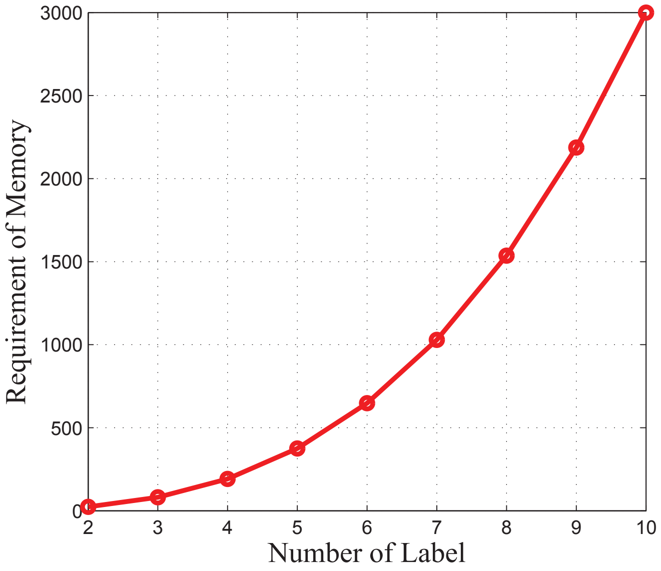

Figure 2.1: Relationship between memory requirements and the number of labels in each weight table for a three-dimensional CMAC

is required, a larger number of labels for each weight table is demanded, and then, for a conventional CMAC a higher quantization of its input space is needed, because the quantization of conventional CMAC is uniform. For a three-dimensional CMAC, a relationship between memory requirements and the number of labels in each weight table is demonstrated in Fig.2.1. From the figure, it is clear that as the requirement for accuracy increases, the requirement for memory increases rapidly.

Additionally, when the number of labels in each weight table increases, a degeneration of CMAC's generalization ability takes place. To improve the drawbacks mentioned above, a CMAC in which the number of labels for each weight table can be set individually is proposed. Such structure allows weight tables with more labels to boost the accuracy of the CMAC and weight tables with less labels to improve the generalization ability of the

CMAC. Thus, learning accuracy requirements, memory requirements and the generalization ability of CMAC can be balanced. In this novel CMAC, for each weight table, hierarchical clustering is employed to determine its labels, the number of labels for each weight table are specified by users. In this study, the newly proposed CMAC is called HC-CMAC.

In this chapter, the proposed controller considers to tune PID controller parameters using HC-CMAC in an on-line manner as a controller parameter "tuner" [51]. This controller focus on improving control performance for transient state, some simulation examples and experiments are demonstrated, and this controller is compared with a conventional CMAC based PID controller which is proposed in [25] to show its effectiveness.

2.1.1 Hierarchical Clustering CMAC

In a conventional CMAC, the input space is quantized uniformly. This means that if high-accuracy learning is required, the number of labels for each weight table grows. This leads to larger memory requirements and a decrease in generalization ability.

Therefore, a method in which each weight table can have a different number of labels for creating a CMAC is proposed. In this case, the weight tables with more labels boost the learning accuracy of the CMAC; while the weight tables with less labels improve the generalization ability of the CMAC and reduce memory requirements.

A CMAC generates similar outputs from similar inputs. Thus, it is important to separate similar inputs and dissimilar inputs into different groups. In the HC-CMAC, hierarchical clustering is utilized to achieve this separation [26]. By using hierarchical clustering, various clusters are created and the center points of these clusters are calculated. The center points are used to represent the clusters, and each center point represents a label in the HC-CMAC. During the learning and control processes, the current data for each input dimension selects the closest center point so that the appropriate weights can be selected. These centers are determined based on the following process :

[Step1]

Initial closed-loop data are collected by applying a fixed PID controller to a controlled objective. For each input dimension, the distances between each data are calculated. For each data point, n neighbors are selected based on the distance. n is a user specified parameter, it should be mentioned that $n \leq \frac{Nod}{Udnl}$. Nod is short for "Number of data", Udnl is short for "User desired number of labels"

[Step2]

The data processing is utilized to create processed data by calculating the average value of sum of each initial data point and its n neighbors. Because neighbors of a data has similar property with the data, thus, it is considered the processed data remain the property of initial data. As an example, in Fig.2.2, $pdata1$=$(data1 + data2)/(1 + n)$. In this case, $n = 1$.

[Step3]

Processed data are clustered using the ward method. The clustering process continues until the user specified number of clusters is achieved. The number of clusters is the user-specified number of labels. In Fig.2.2, the user-specified number of labels is two.

[Step4]

The process stops until the requirements for each weight table are satisfied.

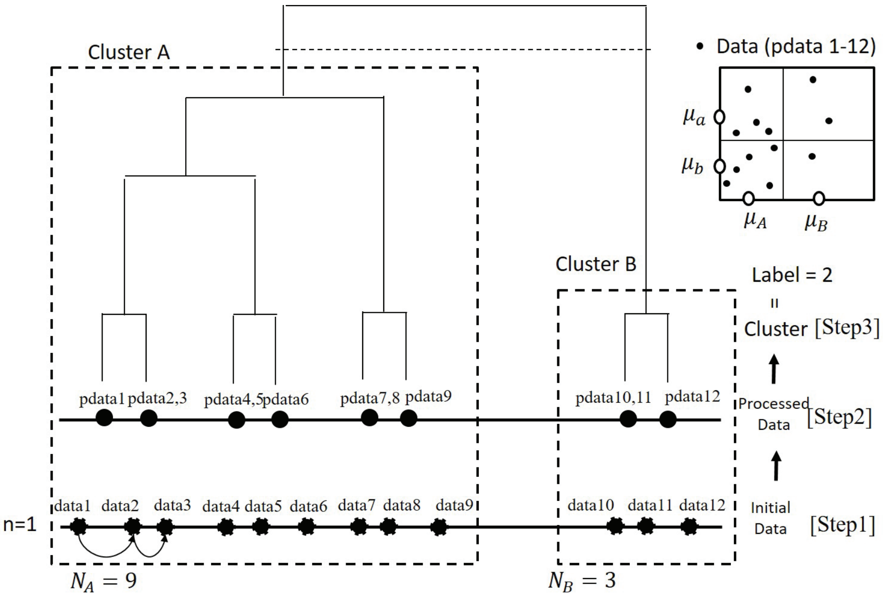

Figure 2.2: Label decision process

In [Step2], the initial data is considered to be processed because the hierarchical clustering technique may leads to the clustering results shown in Fig.2.3. The clustering results within the dashed circle provides an influence presented by the dashed circle in the input space. In the input space, the final labels of second and third layers are the same, which decreases the generalization ability of the CMAC. Therefore, the initial data need to be processed before the labels were created for each weight table.

For [Step3], the distance between two clusters is calculated using the following equation:

$$D(A, B) = \frac{N_A N_B}{N_A + N_B} d(\mu_A, \mu_B)^2, \tag{2.1}$$

where N_A and N_B denote the number of data points belonging to clusters A and B, μ_A and μ_B denote the centers of the two clusters. $D(A, B)$ represents

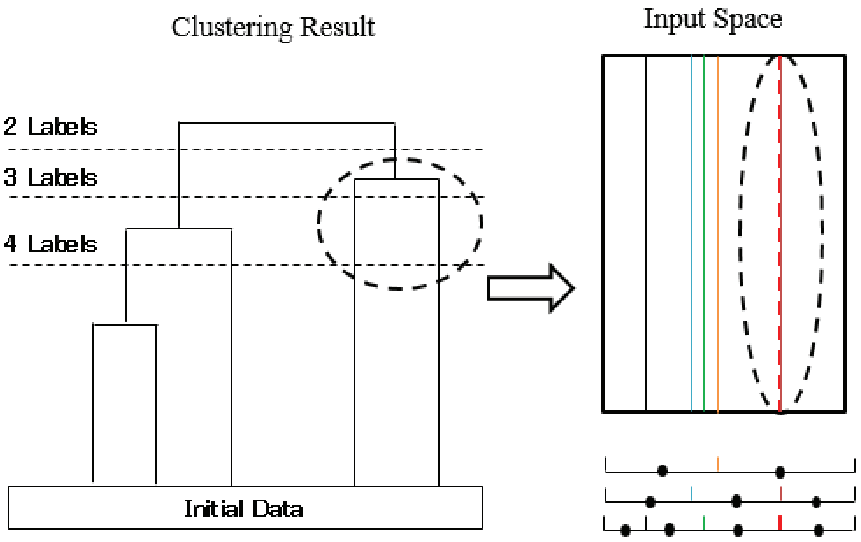

Figure 2.3: Clustering results that may appears when using the initial data

the distance between cluster A and cluster B, while $d(\mu_A, \mu_B)$ is the distance between the center of cluster A and the center of cluster B.

Through the steps above, the center points for each input dimension are created. The inputs can then be evaluated as similar or dissimilar by judging whether they are close to the same center points. As the center points for each input dimension are determined, the HC-CMAC structure is created. An example of a two-dimensional HC-CAMC is presented in Fig.2.4. The dots in the figure represent center points. The figure illustrates a situation where the collected data is concentrated around coordinates with small values. Thus, more labels are distributed in the part of the coordinate space with smaller values in Fig.2.4. In this example, the user specified two, three, and four as the numbers of labels for each weight table. In this case, the first weight table provides the network with better generalization

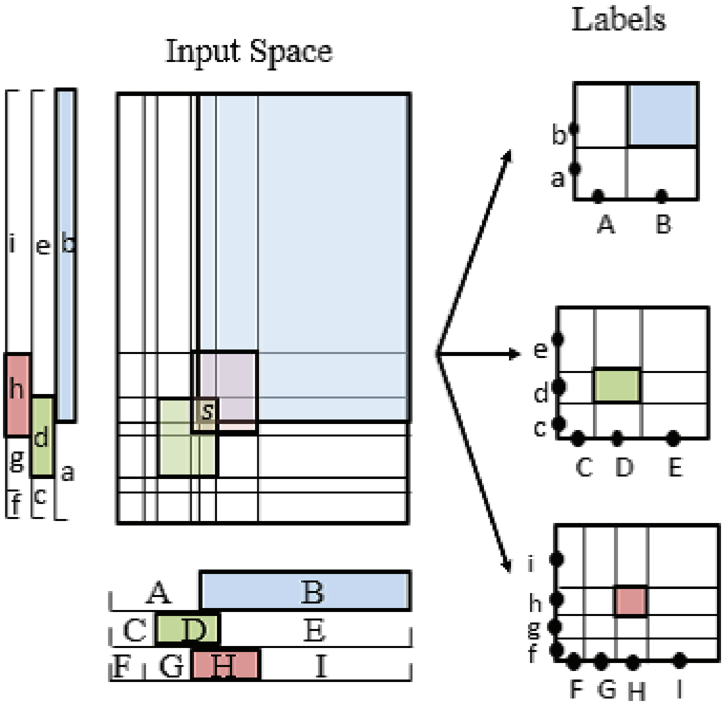

Figure 2.4: Example of a two-dimensional HC-CMAC

ability and the third weight table provides the network with better learning accuracy.

Because the HC-CMAC is utilized to tune PID gains in the proposed method, the following PID control structure is considered:

$$\begin{aligned}\Delta u(t) &= K_I(t)e(t) - K_P(t)\Delta y(t) - K_D(t)\Delta^2 y(t) \\ &= C(1)r(t) - C(z^{-1})y(t),\end{aligned} \tag{2.2}$$

$$e(t) := r(t) - y(t), \tag{2.3}$$

$$C(z^{-1}) = K_I(t) + K_P(t)\Delta + K_D(t)\Delta^2, \tag{2.4}$$

where $e(t)$ denotes the control error signal, $r(t)$ is the reference signal, $y(t)$ represents the signal of the system output, and $u(t)$ represents the signal of the system input. K_P, K_I, and K_D are the controller gains of the proportional, integral, and derivative, respectively. $C(z^{-1})$ denotes the controller. The controller parameters are the output of HC-CMAC:

$$\begin{aligned} K_P(t) &= \sum_{h=1}^{K} W_{P,h} \\ K_I(t) &= \sum_{h=1}^{K} W_{I,h} \\ K_D(t) &= \sum_{h=1}^{K} W_{D,h}, \end{aligned} \tag{2.5}$$

where h=1,2,...,K and K denote the total number of the weight tables in the HC-CMAC and $W_{PID,h}$ are the weights of the HC-CMACs.

2.2 Design of a HC-CMAC-PID controller

In the proposed method, when the following error criterion is minimized, the desired control performance can be obtained:

$$J(t) = \frac{1}{2}\varepsilon(t)^2, \tag{2.6}$$

$$\varepsilon(t) = y_m(t) - y(t), \tag{2.7}$$

where, y_m denotes a reference model, it is a desired system output.

This method is an improvement of some controller parameters that are already applied to controlled objective, thus, the initial weights $W^{ini}_{P,I,Dh}$ are determined as:

$$W^{ini}_{P,I,Dh} = K^{ini}_{P,I,D}\frac{1}{K}, \tag{2.8}$$

where, $K^{ini}_{P,I,D}$ is a set of PID gains that are applied to controlled object, they can be calculated from some methods or determined by experts.

The weights are updated on-line through steepest descent method, to minimize the error criterion:

$$W^{new}_{P,I,Dh} = W^{old}_{P,I,Dh} - \eta_{P,I,D}(t)\frac{\partial J(t)}{\partial K_{P,I,D}(t)}\frac{1}{K}, \tag{2.9}$$

where,

$$\begin{aligned}
\frac{\partial J(t)}{\partial K_P(t)} &= \frac{\partial J(t)}{\partial \varepsilon(t)}\frac{\partial \varepsilon(t)}{\partial y(t)}\frac{\partial y(t)}{\partial u(t)}\frac{\partial u(t)}{\partial K_P(t)} \\
&= -(y_m(t)-y(t))(y(t)-y(t-1))\frac{\partial y(t)}{\partial u(t)} \\
\frac{\partial J(t)}{\partial K_I(t)} &= \frac{\partial J(t)}{\partial \varepsilon(t)}\frac{\partial \varepsilon(t)}{\partial y(t)}\frac{\partial y(t)}{\partial u(t)}\frac{\partial u(t)}{\partial K_I(t)} \\
&= -(y_m(t)-y(t))e(t)\frac{\partial y(t)}{\partial u(t)} \\
\frac{\partial J(t)}{\partial K_D(t)} &= \frac{\partial J(t)}{\partial \varepsilon(t)}\frac{\partial \varepsilon(t)}{\partial y(t)}\frac{\partial y(t)}{\partial u(t)}\frac{\partial u(t)}{\partial K_D(t)} \\
&= -(y_m(t)-y(t))(y(t)-2y(t-1) \\
&\quad +y(t-2))\frac{\partial y(t)}{\partial u(t)}. \qquad (2.10)
\end{aligned}$$

In (2.10), $\partial y(t)/\partial u(t)$ is a system Jacobian, and the sign$(\partial y(t)/\partial u(t))$ is considered as constant. Consider a relationship: $x = |x|\text{sign}(x)$, thus, the system Jacobian can be expressed as:

$$\frac{\partial y(t)}{\partial u(t)} = |\frac{\partial y(t)}{\partial u(t)}|\,\text{sign}(\frac{\partial y(t)}{\partial u(t)}), \tag{2.11}$$

where, sign$(x) = 1(x > 0)$ and sign$(x) = -1(x < 0)$, $|\frac{\partial y(t)}{\partial u(t)}|$ is assumed as a constant and the learning coefficients $\eta_{P,I,D}$ include it.

Theorem 1. When the following situation is satisfied, $e(t)$ converges to zero gradually:

When sign$(\frac{\partial y(t)}{\partial u(t)}) = 1$,

$$0<\eta_{P,I,D}(t)<\frac{2}{\left|\frac{\partial y(t)}{\partial u(t)}\right|((\Delta y(t))^2+e(t)^2+(\Delta^2 y(t))^2)}. \tag{2.12}$$

When $\mathrm{sign}(\frac{\partial y(t)}{\partial u(t)}) = -1$,

$$\frac{-2}{\left|\frac{\partial y(t)}{\partial u(t)}\right|((\Delta y(t))^2 + e(t)^2 + (\Delta^2 y(t))^2)} < \eta_{P,I,D}(t) < 0. \tag{2.13}$$

The proof is demonstrated in Appendix in detail.

Remark In practice, to save calculation time, $\eta_{P,I,D}(t)$ are not calculated at each step, it can be set as a value which is small enough [27]. In Fig.2.5, control schematic figure is given. In this chapter, a HC-CMAC with 3-dimension is used, it is utilized as a 'PID parameter tuner', the structure of the 'tuner' is shown in Fig.2.6. There are 3 input signals of 'PID parameter tuner', they are the reference signal $r(t)$, the control error signal $e(t)$ and the difference of control error signal $\Delta e(t)$. Δ is a differencing operator, it is defined as:

$$\Delta := 1 - z^{-1}. \tag{2.14}$$

The reference model is designed from reference[30]:

$$G_m(z^{-1}) = \frac{z^{-1}P(1)}{P(z^{-1})}, \tag{2.15}$$

$$P(z^{-1}) := 1 + p_1 z^{-1} + p_2 z^{-2}, \tag{2.16}$$

$$\left.\begin{array}{l} p_1 = -2\exp(-\frac{\rho}{\mu})\cos(\frac{\sqrt{4\mu-1}}{2\mu}\rho) \\ p_2 = \exp(-\frac{\rho}{\mu}) \\ \rho := \frac{T_S}{\sigma} \\ \mu := 0.25(1-\delta) + 0.51\delta \end{array}\right\} ., \tag{2.17}$$

where, T_s represents the sampling time, σ and δ are parameters related to the rise-time and the damping oscillation. And they are determined according

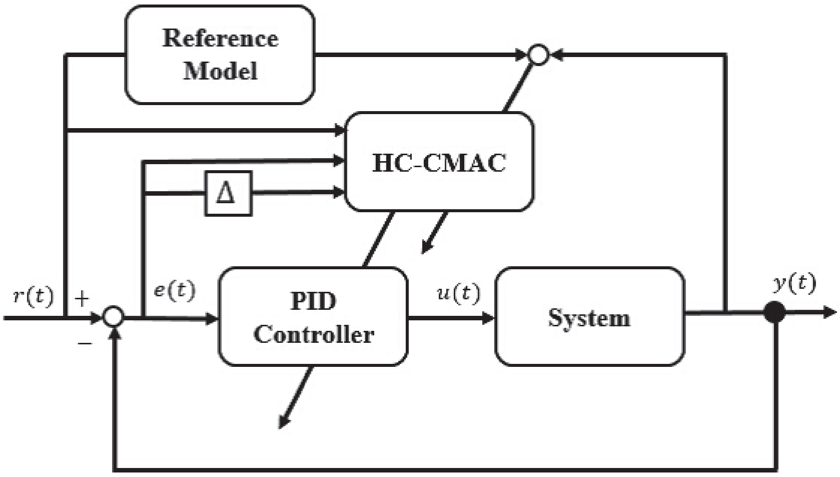

Figure 2.5: Block diagram of HC-CMAC-PID control system

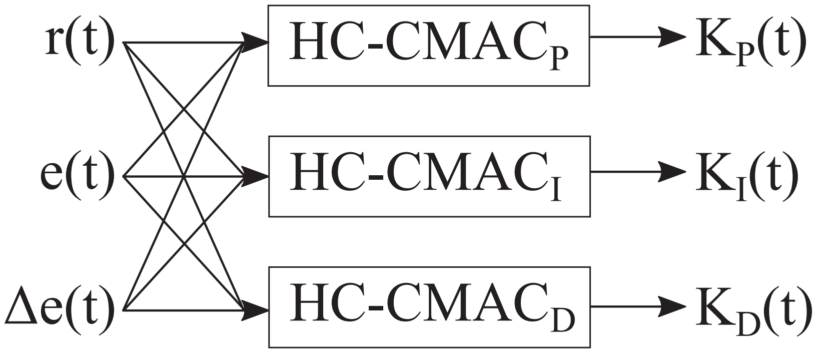

Figure 2.6: Structure of PID parameter tuner

to the requirements of users. Then the reference model $y_m(t)$ is calculated as:

$$\begin{aligned} y_m(t) &= G_m(z^{-1})r(t) \\ &= \frac{z^{-1}(1+p_1+p_2)}{1+p_1z^{-1}+p_2z^{-2}}r(t). \end{aligned} \tag{2.18}$$

Since y_m can be calculated, the error criterion J can be obtained.

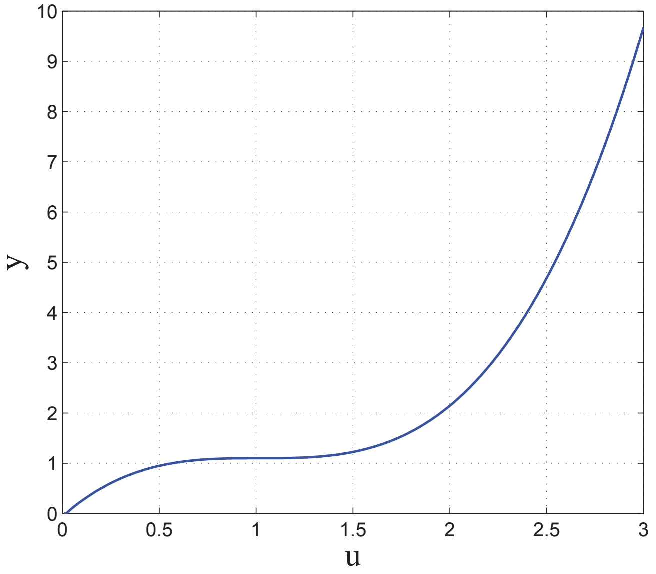

Figure 2.7: Static properties of the control object

2.3 Simulation Examples

The effectiveness of the proposed method are numerically evaluated by using some simulations in this part. PID controller, CMAC-PID controller and HC-CMAC-PID controller are employed to a Hammerstein Model, it is a nonlinear system, the control performances of these controllers are compared. A Hammerstein Model is described as:

$$\left.\begin{array}{l} y(t) = 0.6y(t-1) - 0.1y(t-2) \\ \qquad\quad +1.2x(t-1) - 0.1x(t-2) \\ x(t) = 1.5u(t) - 1.5u^2(t) + 0.5u^3(t) \end{array}\right\}. \tag{2.19}$$

Static property of Hammerstein system is given in Fig.2.7. The PID gains that calculated by using Chien, Hrones & Reswick (CHR) method are firstly applied to the controlled objective [29]. The PID gains are calculated as:

$$K_P = 0.069, K_I = 0.076, K_D = 0.035 \tag{2.20}$$

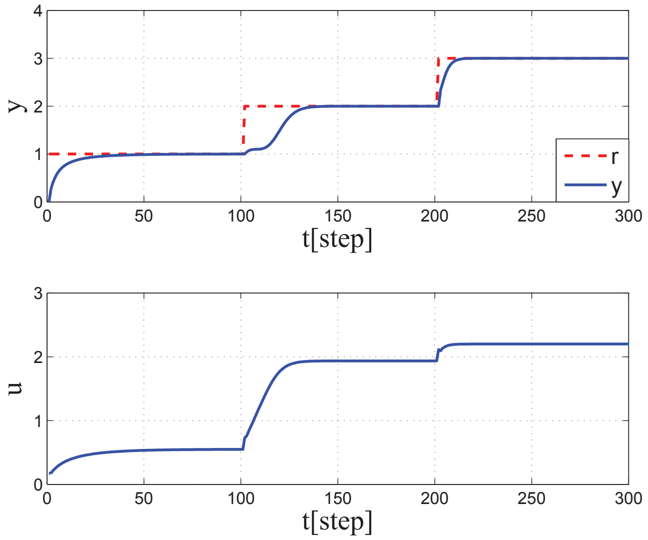

Figure 2.8: Control performance using CHR method

The reference signals $r(t)$ are decided as:

$$r(t) = \begin{cases} 1(1 \leq t \leq 100) \\ 2(101 \leq t \leq 200) \\ 3(201 \leq t \leq 300). \end{cases} \tag{2.21}$$

Due to the nonlinearity of controlled object, in Fig.2.8, the system output can not track reference signal ideally. Then the CMAC-PID controller is employed to the controlled objective, in this example, a CMAC with 3 weight tables and 4 labels is utilized, it is set that when the Integral Squared Error (ISE) is less than 5, the learning process stops. Initial weights in the weight tables are calculated by using the previous PID gains. The learning rates for PID gains are calculated at each step through the following equation:

$$\eta_{P,I,D}(t) = \frac{1}{c + a \cdot \exp(-b|y_m(t) - y(t)|)}, \tag{2.22}$$

$$a = 1000, \quad b = 1000, \quad c = 2000. \tag{2.23}$$

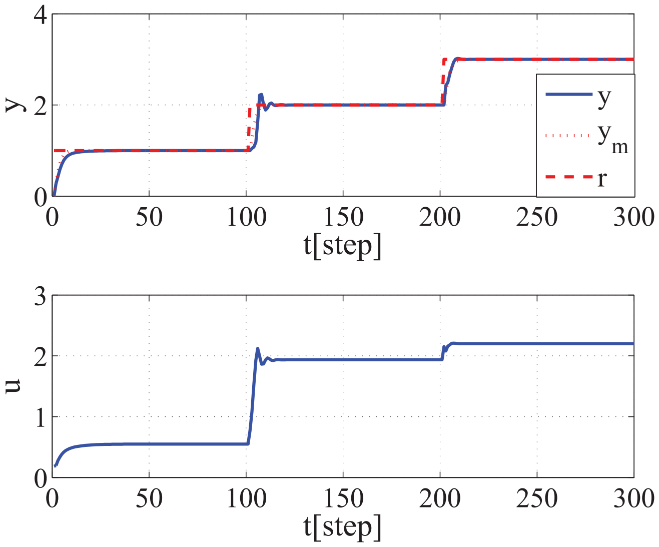

Figure 2.9: Control performance using CMAC-PID method

Simulation result by using CMAC-PID controller is given in Fig.2.9 and the ISE can be obtained from the following formula:

$$ISE = \sum_{t=1}^{k} (y_m(t) - y(t))^2 \tag{2.24}$$

where, k denotes the number of step at each trials. According to the control performance, the system output tracks reference model ideally. Trajectories of PID gains is shown in Fig.2.10, and for each learning trail, an evaluation of ISE is presented, Fig.2.11 shows it. In Fig.2.11, 211 trials are needed to make the ISE small enough.

The simulation result of HC-CMAC-PID controller is performed in the following part.

Firstly, a HC-CMAC with 3 weight tables is constructed by using the closed-loop data which are collected by applying fixed PID controller to controlled objective, in each weight table, the number of labels are set as 2, 3

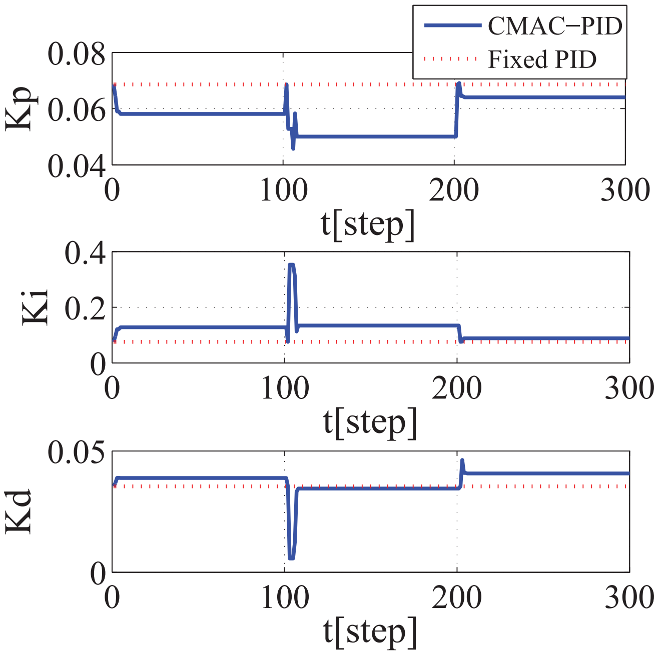

Figure 2.10: Trajectories of PID gians for CMAC-PID

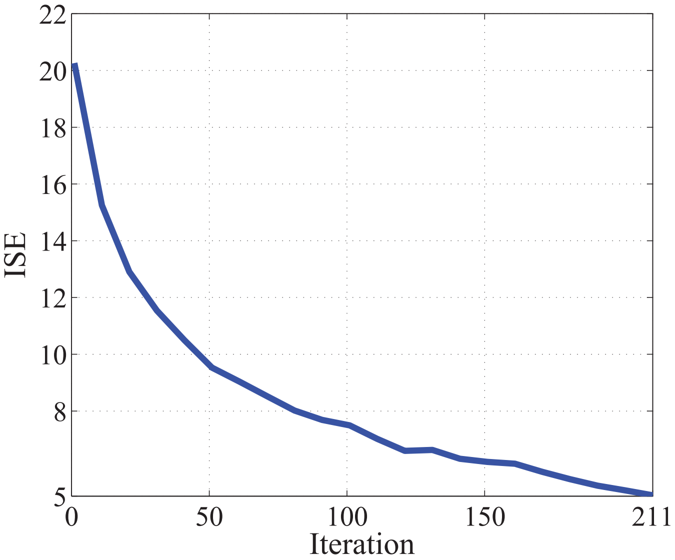

Figure 2.11: Evaluation of ISE for CMAC-PID

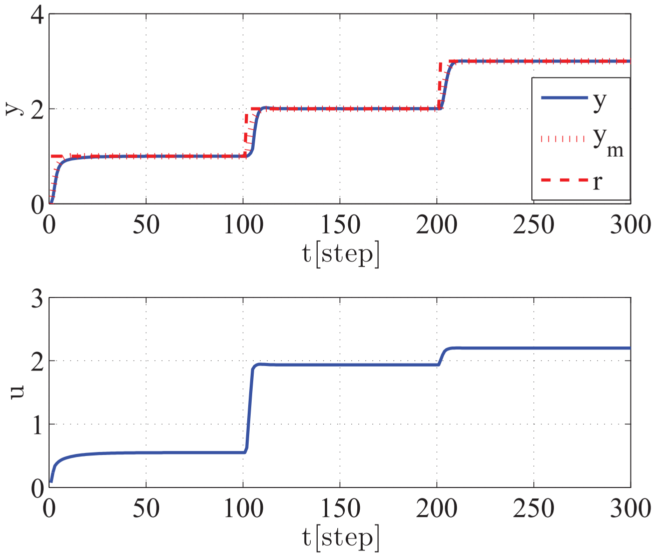

Figure 2.12: Control performance using HC-CMAC-PID method

and 4. Control performance by using HC-CMAC-PID controller is demonstrated in Fig.2.12. It is verified that the tracking performance of system output is ideally. Trajectories of PID gains for the proposed HC-CMAC-PID controller are given in Fig.2.13, the figure evaluates ISE for the proposed method is shown in Fig.2.14. The learning coefficients are also determined through (2.23). According to Fig.2.14, in order to achieve a small enough ISE value, 181 learning times are required. From Fig.2.14 and Fig.2.11, the HC-CMAC-PID controller costs less learning trails to obtain a sufficient small value of ISE, it evaluates HC-CAMC-PID controller a higher generalization ability. In Table 2.1, total memory requirements(TM) of two methods are compared, it shows that the HC-CMAC-PID controller requires a less memory the CMAC-PID controller.

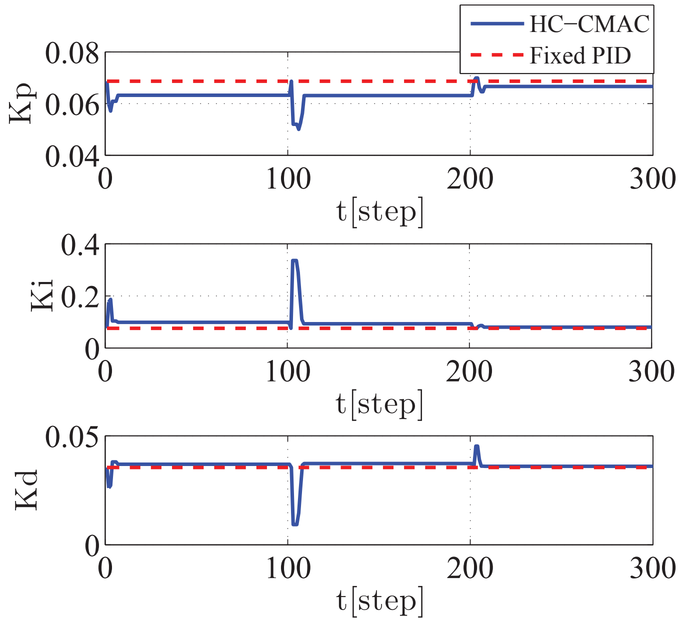

Figure 2.13: Trajectories of PID gains for HC-CMAC-PID method

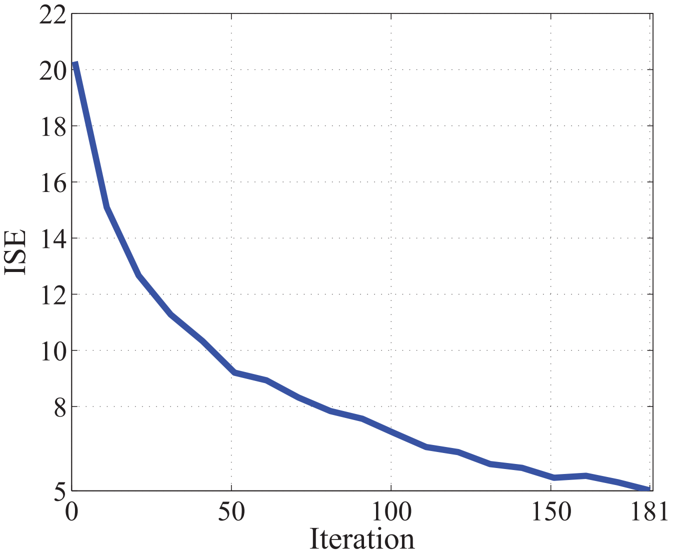

Figure 2.14: Evaluation of ISE for the proposed HC-CMAC-PID method

2.4 Experimental Results

Furthermore some experiments are conducted to evaluate the effectiveness of the HC-CMAC-PID controller. The photograph of the equipment is

Table 2.1: Comparison of CMAC and HC-CMAC

	CMAC	HC-CMAC
TM	4680 bytes	2376 bytes

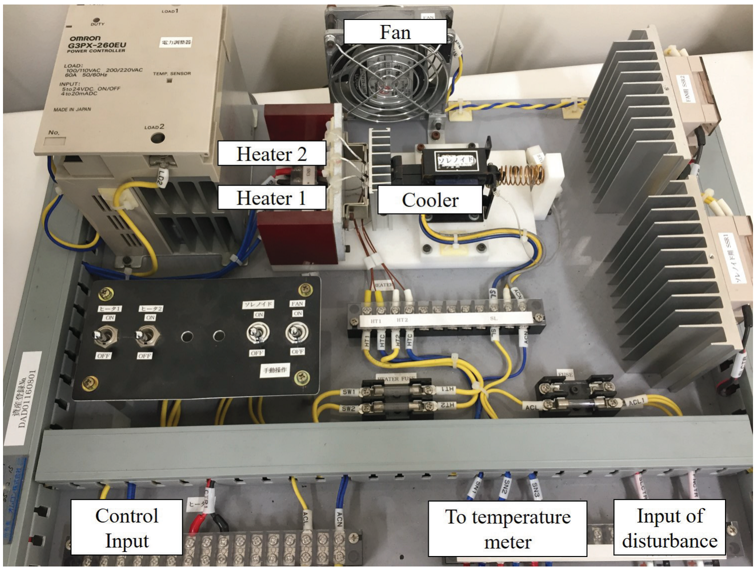

Figure 2.15: Photograph of an experimental temperature control system

shown in Fig.2.15. There are two heaters in this equipment, in this experiment, two heaters are both used. In Fig.2.16, the static property of the controlled objective is demonstrated. The cooler in the figure switches on/off to create the nonlinearity of system. The cooler works for the first reference value and it does not work for the second reference value. The PID gains decided by using CHR method are as follows: $K_P = 0.375 \quad K_I = 0.011$ $K_D = 1.910$. The control performance is shown in Fig.2.17. The sign$(\frac{\partial y(t)}{u(t)})$ of the controlled objective is known as 1, the controller gains' learning coef-

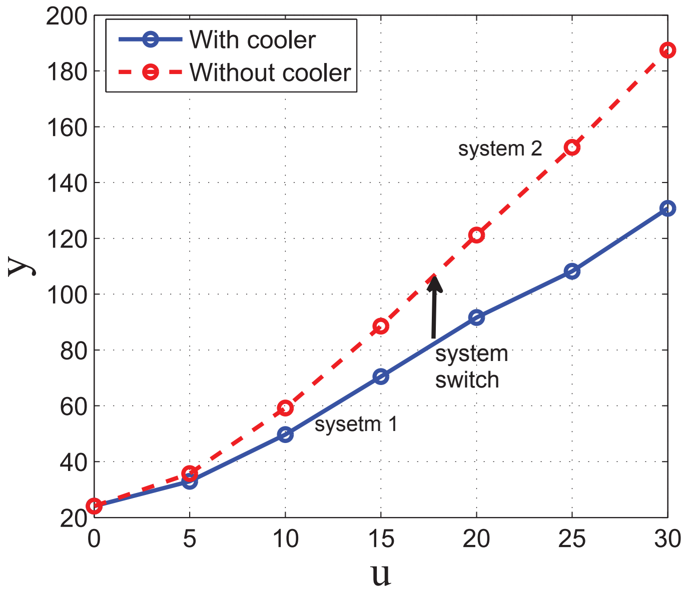

Figure 2.16: Static properties of the control object

ficients and the reference model's parameters are determined as:

$$\eta_P = 1 \times 10^{-4} \quad \eta_I = 2 \times 10^{-7} \quad \eta_D = 1 \times 10^{-3}$$
$$\sigma = 28 \quad \delta = 0 \quad T_s = 1, \tag{2.25}$$

Reference signals are set as:

$$r(t) = \begin{cases} 70(1 \le t \le 300) \quad (\text{with} \quad \text{cooler}) \\ 120(301 \le t \le 600) \quad (\text{without} \quad \text{cooler}) \end{cases}$$

Then, the control abilities of CMAC-PID controller and HC-CMAC-PID controller are examined through some experiments. The trajectories of ISE (from the first trail to the trail with smallest ISE) for learning process of CMAC-PID controllers and HC-CMAC-PID controller are shown in Fig.2.18, and the smallest values of ISE are summarized in Table 2.2 The calculation of ISE bases on the following formula:

$$ISE = \sum_{t=1}^{k} (y_m(t) - y(t))^2. \tag{2.26}$$

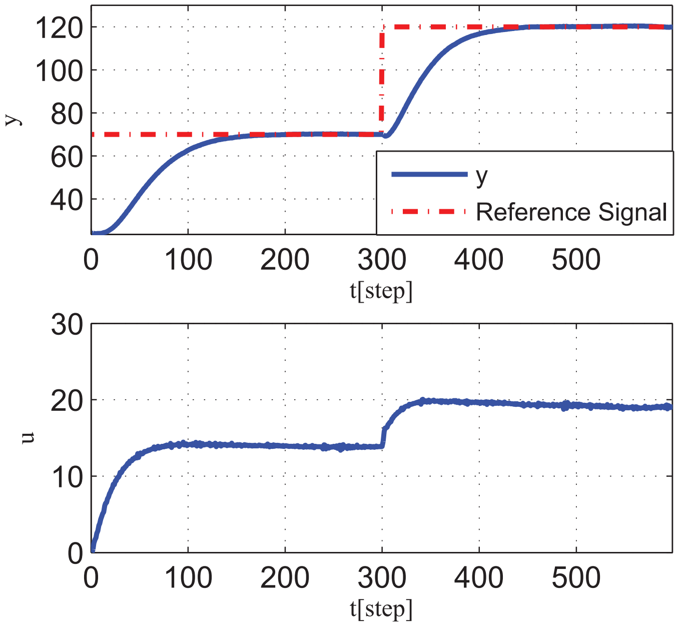

Figure 2.17: Control performance by using CHR method for experiment

In this experimental example, the number of labels included in each weight table in the proposed HC-CMAC-PID controller are six, seven and eight. From Fig. 2.18, the HC-CMAC-PID controller converges to its minimum ISE faster than the CMAC-PID controller with three weight tables and eight labels. At the same time, the HC-CMAC-PID controller has a higher learning accuracy than the CMAC-PID controllers that have three weight tables and six and seven labels for each. It is proved that the proposed HC-CMAC-PID controller balances its control accuracy and the generalization ability.

In Fig.2.19 and Fig.2.21, the experimental result of the CMAC-PID controller with three weight tables and eight labels and the experimental result of the HC-CMAC-PID controller are demonstrated. Moreover, the trajectories of the PID gains for the mentioned two controllers are shown in Fig.2.20 and Fig.2.22, respectively. The number of labels includes in each weight

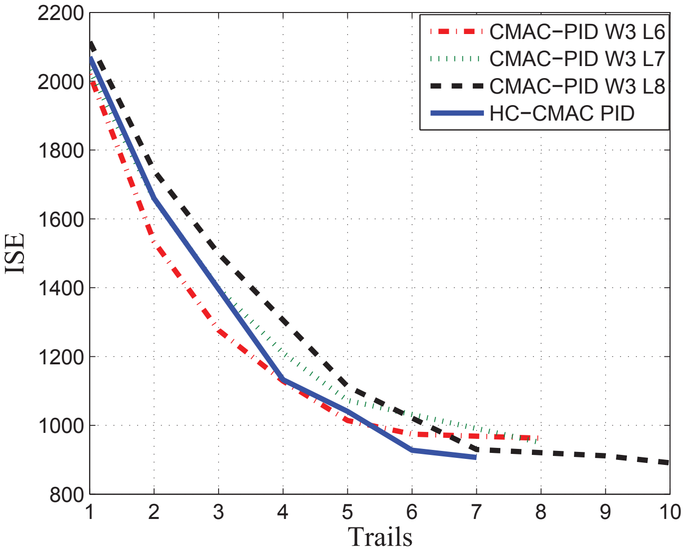

Figure 2.18: Learning performance by using CMACs with three weight tables(W3) and six, seven, eight labels(L6, L7, and L8) and HC-CMAC-PID

Table 2.2: Final value of ISE

	The smallest value of ISE
CMAC PID W3 L6	926.7
CMAC PID W3 L7	949.7
CMAC PID W3 L8	891.1
HC-CMAC-PID	906.9

table(NLW) and the total memory(TM) required in the HC-CMAC-PID controller and CMAC-PID controller are compared in Table2.3. From the table, the proposed method uses less memory. In addition, the HC-CMAC-PID controller and CMAC-PID controller which are trained for the reference signals 70 and 120 are applied to control untrained signals to verify the generaliza-

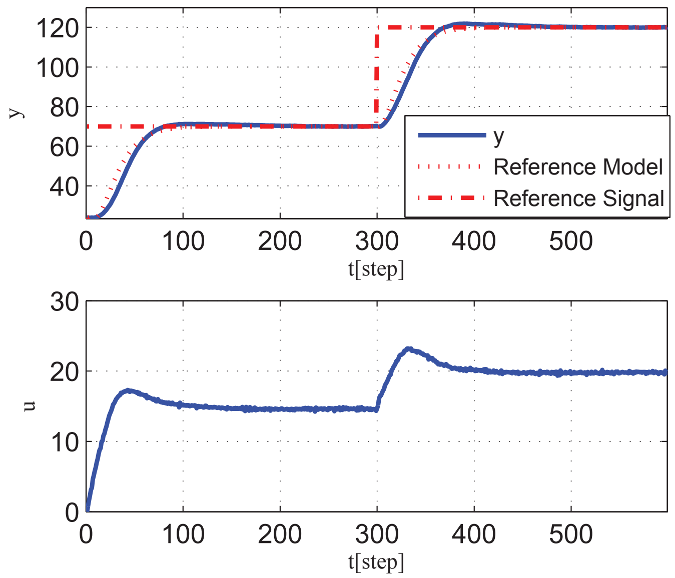

Figure 2.19: Control performance by using CMAC with three weight tables and eight labels

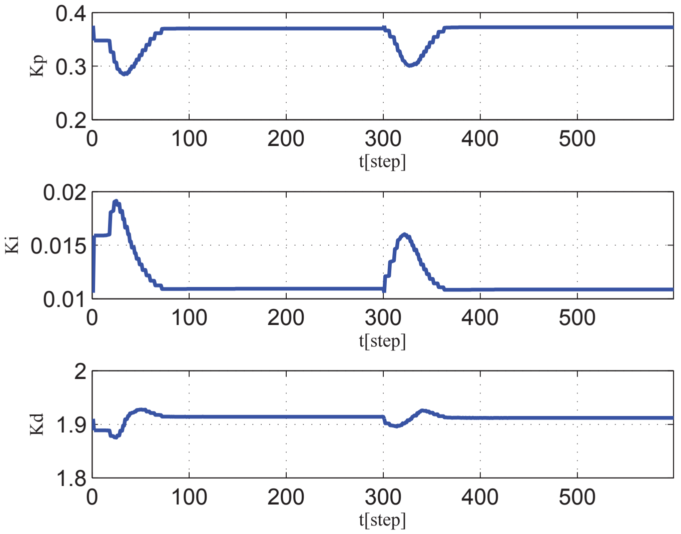

Figure 2.20: Trajectories of PID gains by using CMAC with three weight tables and eight labels

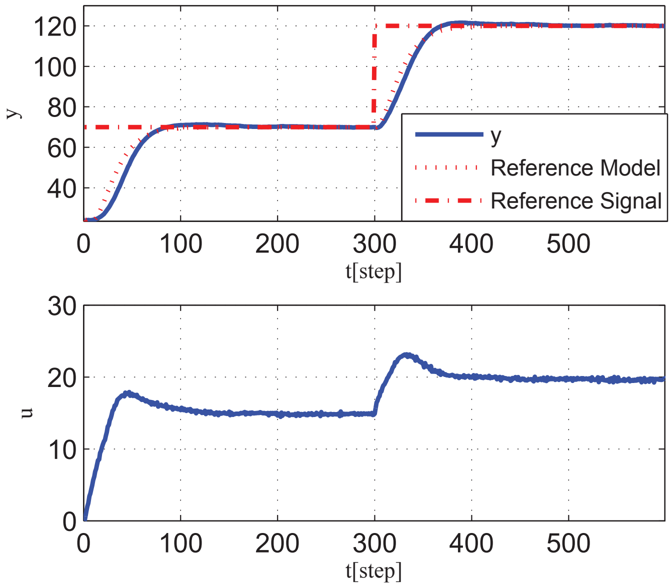

Figure 2.21: Control performance by using HC-CMAC-PID controller

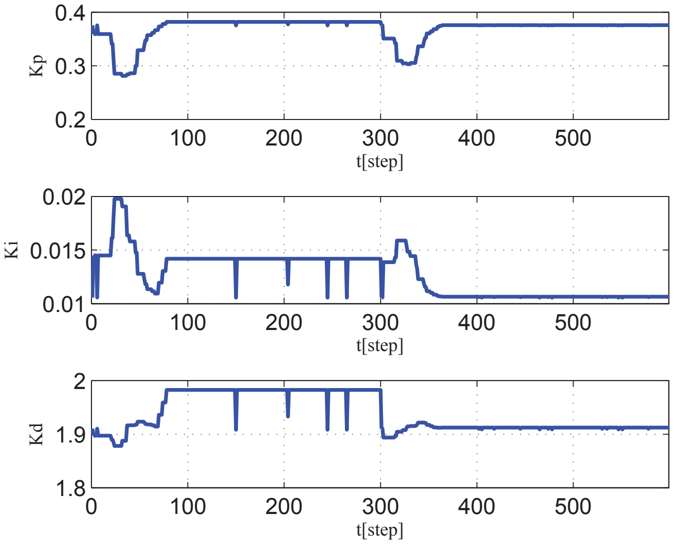

Figure 2.22: Trajectories of PID gains by using HC-CMAC-PID controller

Table 2.3: Comparison of CMAC and HC-CMAC

	CMAC	HC-CMAC
NLW	8, 8, 8	6, 7, 8
TM	36864 bytes	25704 bytes

tion ability of them, the control performances are shown in Fig. 2.23. The untrained reference signals are set as:

$$r(t) = \begin{cases} 75(1 \leq t \leq 300) \quad (\text{with} \quad \text{cooler}) \\ 125(301 \leq t \leq 600) \quad (\text{without} \quad \text{cooler}) \end{cases} .$$

From the experimental results, the HC-CMAC-PID controller shows a better control ability than that of CMAC-PID controller, it certificates that HC-CMAC-PID controller has a better generalization ability than the CMAC-PID controller.

2.5 Conclusions

In this chapter, a novel CMAC which is named as the HC-CMAC is proposed, it is utilized to tune PID gains as a 'gain tuner'. In the proposed HC-CMAC-PID controller, for each weight table, the number of labels is determined individually, so that the control accuracy is compatible with the generalization ability and the requirement of memory. With the application of the proposed method, the requirement of memory reduces and the learning trials for the learning process are reduced, also the control ability for untrained reference signal is improved.

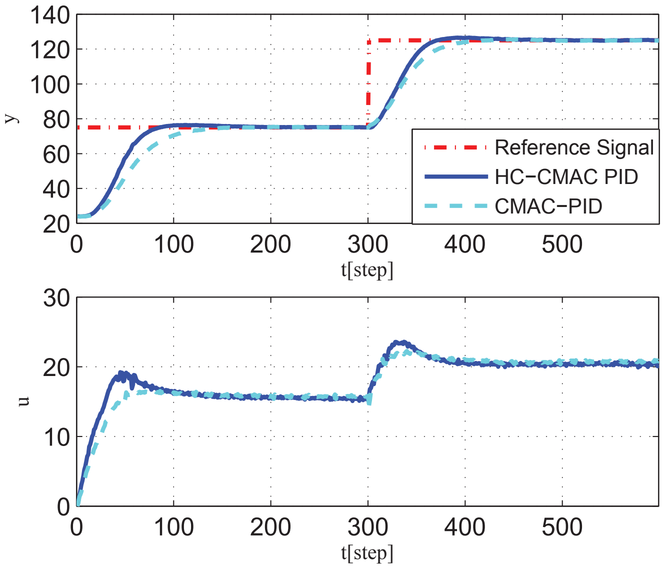

Figure 2.23: Control performances of CMAC with 3 weight tables 8 labels and HC-CMAC-PID controller

2.6 Appendix

To obtain the conditions that are shown in (2.12) and (2.13), the following statement are given [30] [31]:

Consider

$$\Theta = [K_P \quad K_I \quad K_D]^T, \tag{2.27}$$

$$\eta = \begin{bmatrix} \eta_{K_P} & 0 & 0 \\ 0 & \eta_{K_I} & 0 \\ 0 & 0 & \eta_{K_D} \end{bmatrix}. \tag{2.28}$$

Let Lyapunov function be

$$V(t) = \frac{1}{2}e(t)^2, \tag{2.29}$$

then

$$\begin{aligned}\Delta V(t) &:= V(t+1) - V(t) \qquad (2.30)\\ \Delta V(t) &= \frac{1}{2}e(t+1)^2 - \frac{1}{2}e(t)^2\\ &= \frac{1}{2}(e(t+1) - e(t))(e(t+1) + e(t)).\end{aligned}$$

Consider the following relationship

$$e(t+1) = e(t) + \Delta e(t), \qquad (2.31)$$

then

$$\begin{aligned}\Delta V(t) &= \frac{1}{2}\{\Delta e(t)\}\{2e(t) + \Delta e(t)\} \qquad (2.32)\\ &= \frac{1}{2}\Delta e(t)^2 + e(t)\Delta e(t). \qquad (2.33)\end{aligned}$$

Δe can be expressed as:

$$\begin{aligned}\Delta e(t) &= \frac{\partial e(t)}{\partial \Theta^T}\Delta\Theta \qquad (2.34)\\ &= \frac{\partial e(t)}{\partial y(t)}\frac{\partial y(t)}{\partial u(t)}\frac{\partial u(t)}{\partial \Theta^T}\Delta\Theta. \qquad (2.35)\end{aligned}$$

Furthermore, $\Delta\Theta$ is calculated as:

$$\Delta\Theta = -\eta(t)\frac{\partial J(t)}{\partial \varepsilon(t)}\frac{\partial \varepsilon(t)}{\partial y(t)}\frac{\partial y(t)}{\partial u(t)}\frac{\partial u(t)}{\partial \Theta}. \qquad (2.36)$$

sign($\frac{\partial y(t)}{\partial u(t)}$) and $\left|\frac{\partial y(t)}{\partial u(t)}\right|$ are included in $\eta(t)$, (2.36) can be expressed as:

$$\Delta\Theta = -\frac{\partial J(t)}{\partial \varepsilon(t)}\frac{\partial \varepsilon(t)}{\partial y(t)}\eta(t)\frac{\partial u(t)}{\partial \Theta}. \qquad (2.37)$$

From (2.3) and (2.7), based on (2.11) and (2.37)

$$\Delta e(t) = -\frac{\partial J(t)}{\partial \varepsilon(t)}\left|\frac{\partial y(t)}{\partial u(t)}\right| \text{sign}(\frac{\partial y(t)}{\partial u(t)})(\frac{\partial u(t)}{\partial \Theta^T}\eta(t)\frac{\partial u(t)}{\partial \Theta}). \qquad (2.38)$$

Substitute (2.38) into (2.33)

$$\begin{aligned}\Delta V(t) &= \frac{1}{2}\Delta e(t)^2 + \Delta e(t)e(t) \qquad (2.39)\\ &= \frac{1}{2}(\frac{\partial J(t)}{\partial \varepsilon(t)})^2 \left|\frac{\partial y(t)}{\partial u(t)}\right|^2 (\text{sign}(\frac{\partial y(t)}{\partial u(t)}))^2\\ &(\frac{\partial u(t)}{\partial \Theta^T}\eta(t)\frac{\partial u(t)}{\partial \Theta})^2\\ &- \frac{\partial J(t)}{\partial \varepsilon(t)}\left|\frac{\partial y(t)}{\partial u(t)}\right| \text{sign}(\frac{\partial y(t)}{\partial u(t)})(\frac{\partial u(t)}{\partial \Theta^T}\eta(t)\frac{\partial u(t)}{\partial \Theta})e(t).\end{aligned}$$

When $t \to \infty$, $\varepsilon(t) = e(t)$, since $(\text{sign}(\frac{\partial y(t)}{\partial u(t)}))^2 = 1$ and $\frac{\partial J(t)}{\partial \varepsilon(t)} = \varepsilon(t)$, (2.39) can be rewritten as:

$$\begin{aligned}\Delta V(t) = &\frac{1}{2}(e(t))^2 \left|\frac{\partial y(t)}{\partial u(t)}\right|^2 (\frac{\partial u(t)}{\partial \Theta^T}\eta(t)\frac{\partial u(t)}{\partial \Theta})^2 \qquad (2.40)\\ &- e(t)^2 \left|\frac{\partial y(t)}{\partial u(t)}\right| \text{sign}(\frac{\partial y(t)}{\partial u(t)})(\frac{\partial u(t)}{\partial \Theta^T}\eta(t)\frac{\partial u(t)}{\partial \Theta}).\end{aligned}$$

To make sure $\Delta V(t)<0$, sign of $\eta(t)$ and $\text{sign}(\frac{\partial y(t)}{\partial u(t)})$ must be same. When $\text{sign}(\frac{\partial y(t)}{\partial u(t)}) = 1$, the following equation can be obtained by

$$\begin{aligned}\Delta V(t) = &\frac{1}{2}(e(t))^2 \left|\frac{\partial y(t)}{\partial u(t)}\right|^2 (\frac{\partial u(t)}{\partial \Theta^T}\eta(t)\frac{\partial u(t)}{\partial \Theta})^2 \qquad (2.41)\\ &- e(t)^2 \left|\frac{\partial y(t)}{\partial u(t)}\right| (\frac{\partial u(t)}{\partial \Theta^T}\eta(t)\frac{\partial u(t)}{\partial \Theta})<0.\end{aligned}$$

Then the following condition is obtained by

$$\frac{\partial u(t)}{\partial \Theta^T}\eta(t)\frac{\partial u(t)}{\partial \Theta}<\frac{2}{\left|\frac{\partial y(t)}{\partial u(t)}\right|}. \qquad (2.42)$$

There exists a max value η_x such that the following equation holds

$$\frac{\partial u(t)}{\partial \Theta^T}\eta_x I\frac{\partial u(t)}{\partial \Theta} = \frac{2}{\left|\frac{\partial y(t)}{\partial u(t)}\right|}. \qquad (2.43)$$

Based on PID structure in (2.2), (2.43) can be calculated as

$$\eta_x = \frac{2}{\left|\frac{\partial y(t)}{\partial u(t)}\right| (\Delta^2 y(t) + e(t)^2 + \Delta^4 y(t))}. \qquad (2.44)$$

From above, to guarantee (2.42) holds

$$0<\eta_i(t)<\eta_x = \frac{2}{\left|\frac{\partial y(t)}{\partial u(t)}\right|((\Delta y(t))^2 + e(t)^2 + (\Delta^2 y(t))^2)}, \tag{2.45}$$

where, $\eta_i(t)$ is a value belongs to $\eta_{K_P}(t)$, $\eta_{K_I}(t)$ and $\eta_{K_D}(t)$. When $\text{sign}(\frac{\partial y(t)}{\partial u(t)}) = -1$, $\eta(t)$ should be negative, the range of $\eta_i(t)$ should be

$$-\frac{2}{\left|\frac{\partial y(t)}{\partial u(t)}\right|((\Delta y(t))^2 + e(t)^2 + (\Delta^2 y(t))^2)}<\eta_i(t)<0. \tag{2.46}$$

The $\left|\frac{\partial y(t)}{\partial u(t)}\right|$ can be estimated by $\left|\frac{\Delta y(t)}{\Delta u(t)}\right|$.

Chapter 3

Design of HC-CMAC PID Controller Using Closed-Loop Data

3.1 Introduction

In the previous chapter, a HC-CMAC is introduced and an on-line tuning method of PID gains based on it is discussed. The on-line learning controllers always needs a large time cost for its learning process. Therefore, the off-line tuning methods of controllers are needed to be designed.

Some off-line controller tuning methods are proposed [34] [35] [36]. As one of them, a CMAC based PID controller that its weights are updated in an off-line manner is going to be explained. The method considers modeling controlled objectives is nontrivial and modeling errors are difficult to avoid, thus, the fictitious reference iterative tuning (FRIT) algorithm is utilized [37] [38]. FRIT is a method used to calculate control parameters directly by using closed-loop data so that the control parameters can be obtained in an off-line manner. In this previous study, the combining of conventional CMAC and

FRIT was introduced, it is called the CMAC-FRIT-PID controller [39] [40]. This chapter proposes a method called the HC-CMAC-FRIT-PID controller as an improvement of remedy various drawbacks of the conventional method. The effectiveness of the proposed method is verified by some simulations and experiments.

The chapter is organized as follows: First, the FRIT algorithm is explained. Second, the HC-CMAC-FRIT and its learning process are discussed. Finally, some simulations and experiments are provided.

3.2 HC-CMAC based PID Controller using FRIT

3.2.1 Fictitious Reference Iterative Tuning

FRIT algorithm generates controller gains without building a model of controlled objective, based on a set of closed-loop data, the calculation of controller parameters is achievable. Such method is different from the model based controller calculation algorithms, since the model is not established, the negative influences that caused by modeling error can not be considered.

To collect closed-loop data of a controlled objective, some methods should be applied, such as a PID controller.

The schematic block diagram of FRIT is demonstrated in Fig.3.1, some explanations of FRIT is given in the following part. $u_0(t)$ is system input and $y_0(t)$ denotes system output of closed-loop data in the figure. $G_m(z^{-1})$ is user-desired tracking performance, it is a reference model. When the difference of $y_r(t)$ and $y_0(t)$ is minimized, the optimized controller gains can be generated by using FRIT. Under the effect of such optimized controller gains, the portion in the dotted square converges to the reference model $G_m(z^{-1})$.

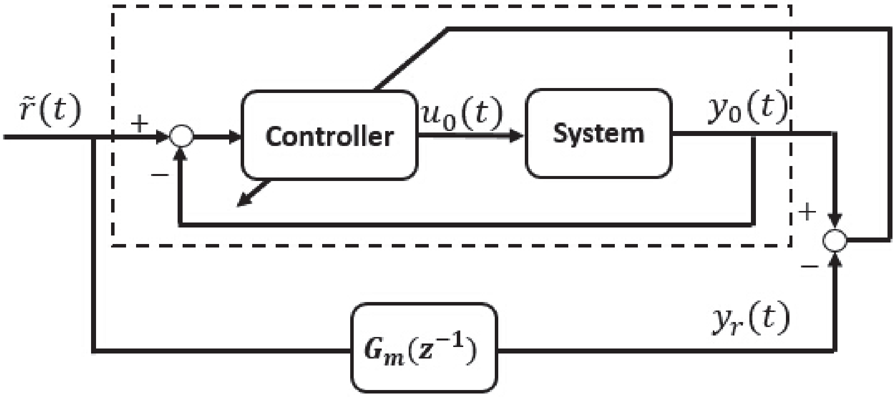

Figure 3.1: Block diagram of FRIT

The design of reference model bases on the following formulas:

$$G_m(z^{-1}) = \frac{z^{-1}P(1)}{P(z^{-1})}, \tag{3.1}$$

$$P(z^{-1}) := 1 + p_1 z^{-1} + p_2 z^{-2}, \tag{3.2}$$

$$\left.\begin{array}{l} p_1 = -2\exp(-\frac{\rho}{\mu})\cos(\frac{\sqrt{4\mu-1}}{2\mu}\rho) \\ p_2 = \exp(-\frac{\rho}{\mu}) \\ \rho := \frac{T_S}{\sigma} \\ \mu := 0.25(1-\delta) + 0.51\delta \end{array}\right\}. \tag{3.3}$$

In (3.3), σ and δ are designed parameters influence rise-time and damping oscillation, respectively. The sample time is represented by T_s. To calculate $y_r(t)$, the fictitious system output, the following equations are taken into consideration:

$$\begin{aligned} y_r(t) &= G_m(z^{-1})\tilde{r}(t) \\ &= \frac{z^{-(k+1)}(1+p_1+p_2)}{1+p_1 z^{-1}+p_2 z^{-2}}\tilde{r}(t). \end{aligned} \tag{3.4}$$

In this chapter, the following PID algorithm is used:

$$\begin{aligned} \Delta u(t) &= K_I(t)e(t) - K_P(t)\Delta y(t) - K_D(t)\Delta^2 y(t) \\ &= C(1)r(t) - C(z^{-1})y(t), \end{aligned} \tag{3.5}$$

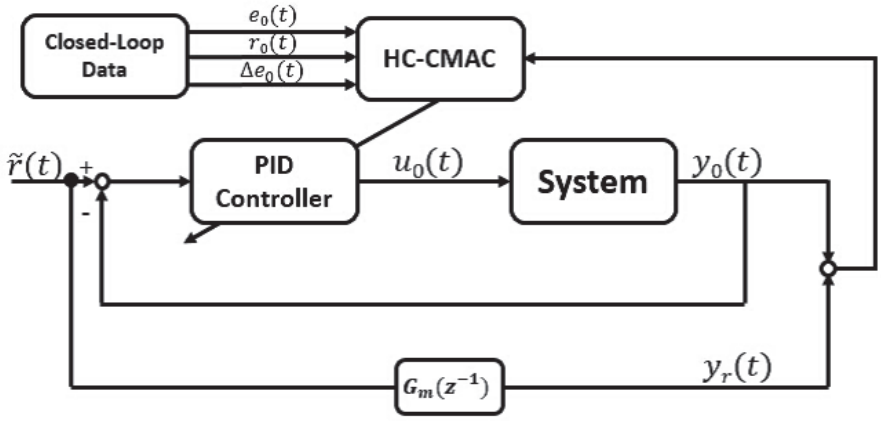

Figure 3.2: Block diagram of HC-CMAC-FRIT

$$e(t) \quad := \quad r(t) - y(t), \tag{3.6}$$

$$C(z^{-1}) = K_I(t) + K_P(t)\Delta + K_D(t)\Delta^2. \tag{3.7}$$

From formulas (3.5) - (3.7), $\tilde{r}(t)$ which represents the fictitious reference signal, can be derived as:

$$\tilde{r}(t) \quad = \quad \frac{\Delta u_0(t) + C(z^{-1})y_0(t)}{C(1)}. \tag{3.8}$$

3.2.2 HC-CMAC-FRIT

The HC-CMAC is combined with FRIT to enable updating of HC-CMAC in off-line. The details are explained in this section. The block diagram of HC-CMAC-FRIT is given in Fig. 3.2. $e_0(t)$, $r_0(t)$, $\Delta e_0(t)$, and $y_0(t)$ denote signals of closed-loop data in the figure. Three signals, the control error signal e_0, reference signal r_0 and difference in control error signal Δe_0 are chosen as coordinates of HC-CMAC. From equation: $K_{P,I,D} = \sum_{h=1}^{K} W_{P,I,D,h}$, PID controller gains are generated, $W_{P,I,D,h}$ are selected weights, controller gains are summation of these weights. The weights' updating rule is given in the following equation:

$$W^{new}_{P,I,D,h} \;=\; W^{old}_{P,I,D,h} - \eta_{P,I,D}\frac{\partial J(t+1)}{\partial K_{P,I,D}(t)}\frac{1}{K}, \tag{3.9}$$

where $J(t)$ denotes an error criterion. It is utilized to judge whether the controller gains are optimized enough. The following formula shows the criterion:

$$J(t) \;=\; \frac{1}{2}(y_0(t) - y_r(t))^2 \qquad (t = 1, ..., N), \tag{3.10}$$

where, number of closed-loop data points is represented by N. In (3.9), the parameter $\eta_{P,I,D}$ denotes the learning coefficients for controller gains K_P, K_I and K_D, partial differentials contained in the formula can be extended as:

$$\frac{\partial J(t+1)}{\partial K_{P,I,D}(t)} \;=\; \frac{\partial J(t+1)}{\partial y_r(t+1)}\frac{\partial y_r(t+1)}{\partial \tilde{r}(t)}\frac{\partial\ \tilde{r}(t)}{\partial K_{P,I,D}(t)}. \tag{3.11}$$

In appendix, an expansion of (3.11) is explained in detail. Based on such learning algorithm, weights of HC-CMAC can be updated in off-line manner.

In the following section, some simulations and experiments are demonstrated to verify the effectiveness.

3.3 Simulation Examples

The controlled objective utilized in the simulation is a Hammerstein Model. The demonstration of control performances using a fixed PID controller, CMAC-FRIT-PID controller, and HC-CMAC-FRIT-PID controller is given. The Hammerstein Model is given as:

$$\left.\begin{array}{l} y(t) = 0.6y(t-1) - 0.1y(t-2) \\ \qquad\quad +1.2x(t-1) - 0.1x(t-2) \\ x(t) = 1.5u(t) - 1.5u^2(t) + 0.5u^3(t) \end{array}\right\}. \tag{3.12}$$

Fig.3.3 presents static properties of this Hammerstein model. Around $y(t)$=1.1,

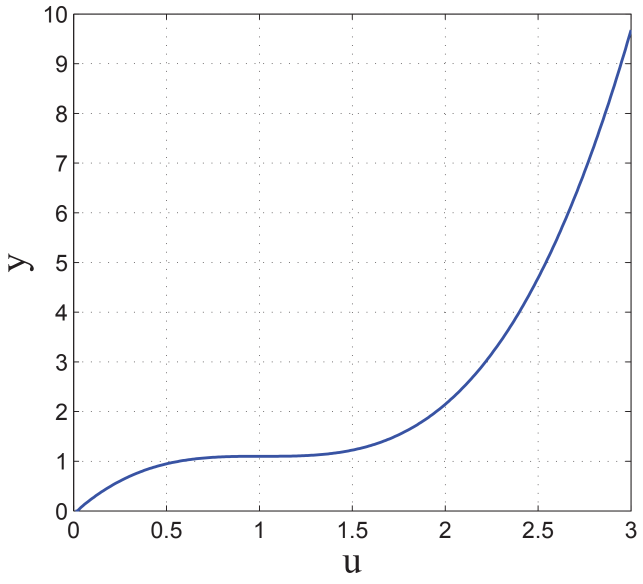

Figure 3.3: Static properties of the control object

the system has a strong nonlinearity. At first, Chien, Hrones & Reswick (CHR) method is utilized to calculated controller parameters. The calculated PID gains are K_P=0.101, K_I=0.111 and K_D=0.052. Reference signal $r(t)$ is decided as:

$$r(t) = \left\{ \begin{array}{l} 1.5(1 \leq t \leq 50) \\ 0.6(51 \leq t \leq 100) \end{array} \right. . \tag{3.13}$$

Fig.3.4 shows the result. It can be observed that the tracking ability is not desirable.

The closed-loop data by using CHR method is collected to train the CMAC and HC-CMAC, thus, controller gains can be optimized. Some design parameters for the CMAC-FRIT controller and HC-CMAC-FRIT controller are summarized in Table.3.1. By employing CMAC-FRIT-PID controller, control performance that presents in Fig.3.5 can be obtained. In this CMAC-FRIT-PID controller, the selected CMAC has three weight tables and four

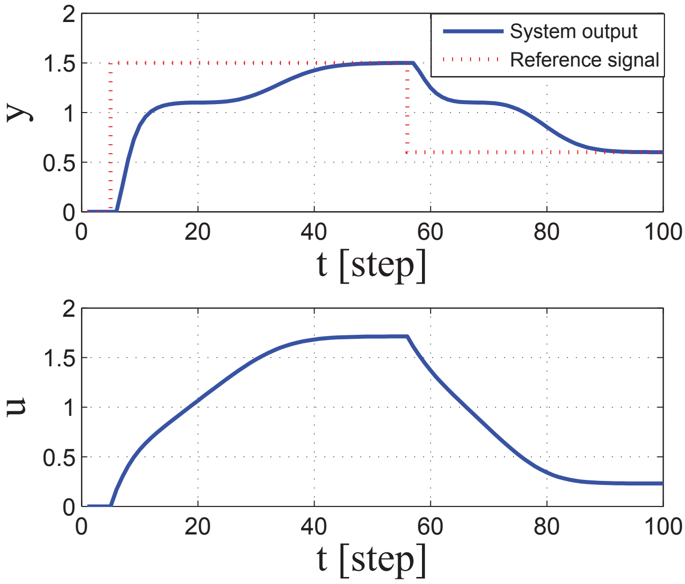

Figure 3.4: Control performance using CHR method

Table 3.1: Coefficients

Learning coefficients	η_P=0.1 η_I=0.05 η_D=0.1
Sampling time	T_S=1
Rise time	σ=3.64
Damping oscillation	δ=0

labels for each. From the figure, desired tracking performance is obtained when the CMAC-FRIT-PID controller is utilized. PID gains' trajectories when using CMAC-FRIT-PID controller are given in Fig.3.6. In Fig.3.7, the control result using HC-CMAC-FRIT-PID controller is shown. For this HC-CMAC-FRIT-PID controller, it has three weight tables include two, three, and four labels for each. The proposed method requires less memory and

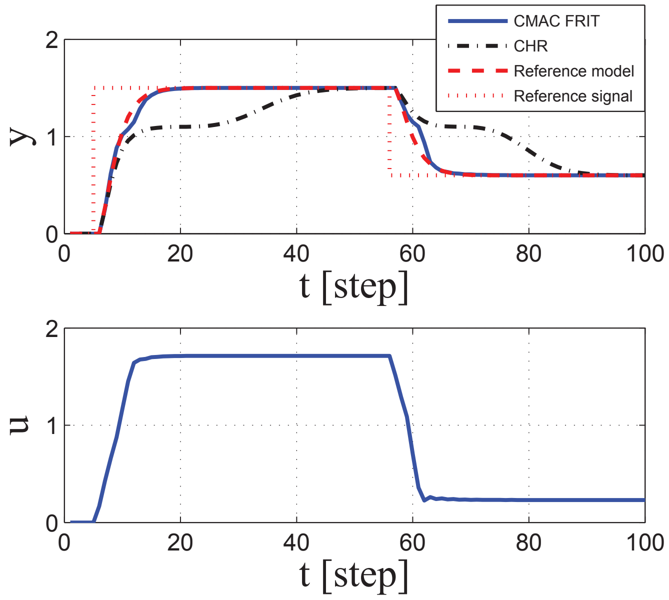

Figure 3.5: Control performance using CMAC-FRIT method

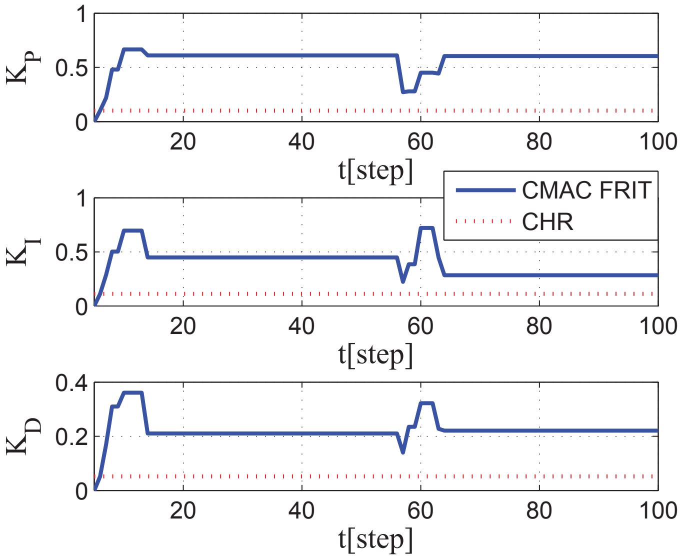

Figure 3.6: Trajectories of PID gains using CMAC-FRIT method

achieves similar control performance as CMAC-FRIT-PID controller. The trajectories of the PID gains when using HC-CMAC-FRIT-PID controller

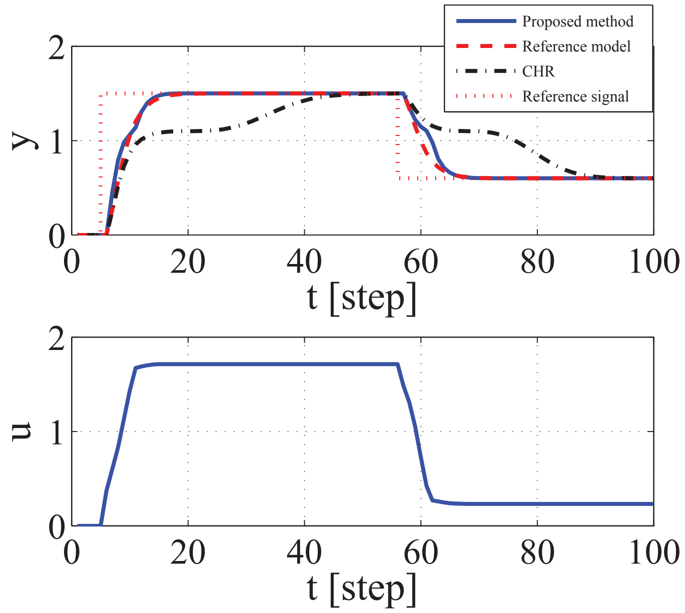

Figure 3.7: Control performance using proposed method

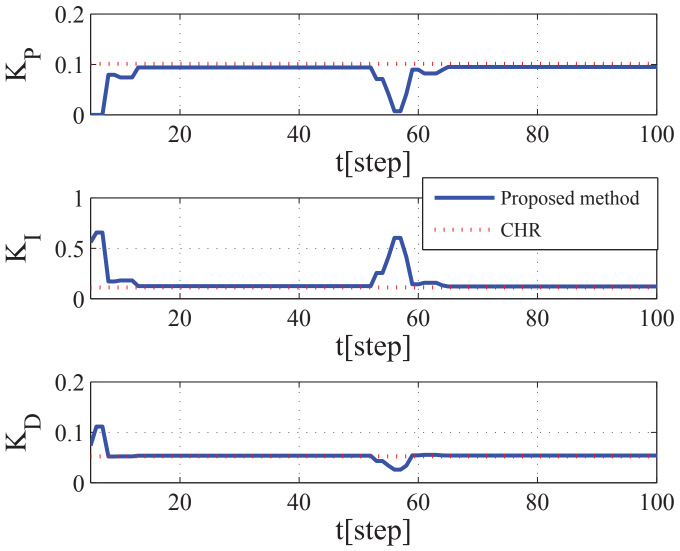

Figure 3.8: Trajectories of PID gains using proposed method

are demonstrated in Fig.3.8. Utilization of memory is explained in detail in Table 3.2.

Table 3.2: Comparison of memory requirement

	CMAC	HC-CMAC
NLW	4, 4, 4	2, 3, 4
TM	4680 bytes	2376 bytes

In Table 3.2, NLW is short for number of labels for each weight table and TM is short for total required memory.

Additionally, in order to verify the generalization ability of the proposed and CMAC-FRIT-PID controller, the trained CMAC and HC-CMAC are applied to control untrained reference signal, they are determined as:

$$r(t) = \begin{cases} 1.7(1 \leq t \leq 50) \\ 0.4(51 \leq t \leq 100) \end{cases} . \tag{3.14}$$

Fig.3.9 shows a comparison of the HC-CMAC-FRIT-PID controller and CMAC-FRIT-PID controller. Fig.3.10 shows the trajectories of the PID gains corresponds to Fig.3.9. The verification of the HC-CMAC-FRIT-PID controller has a better generalization ability is certificated from the above results.

3.4 Experimental Results

Some experiments are demonstrated in this section to verify the effectiveness of the HC-CMAC-FRIT-PID controller. The controlled objective is a temperature control system, its photograph is shown in Fig.3.11. There are two heaters on the machine, in this experiment only heater 1 is used. The static properties of the controlled objective are presented through Fig.3.12. The dashed line shows a property of a system without cooler and the solid line shows a property of a system with cooler. The cooler designs the nonlin-

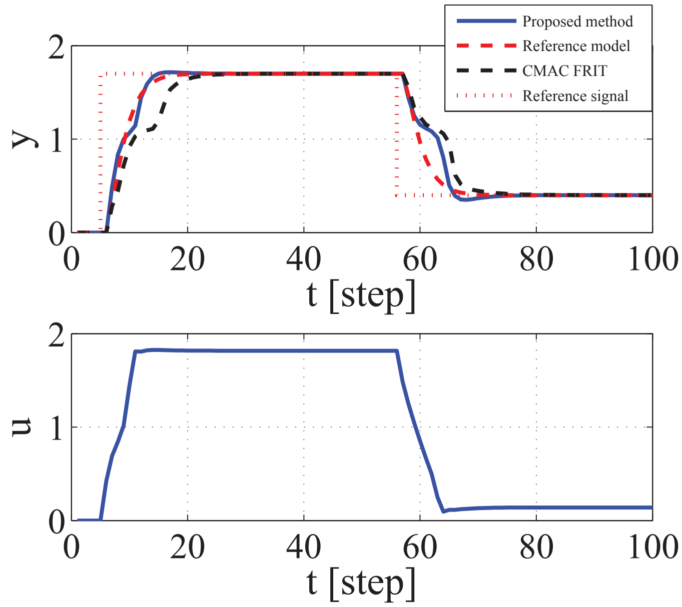

Figure 3.9: Control performance for untrained reference signals using proposed method and CMAC-FRIT

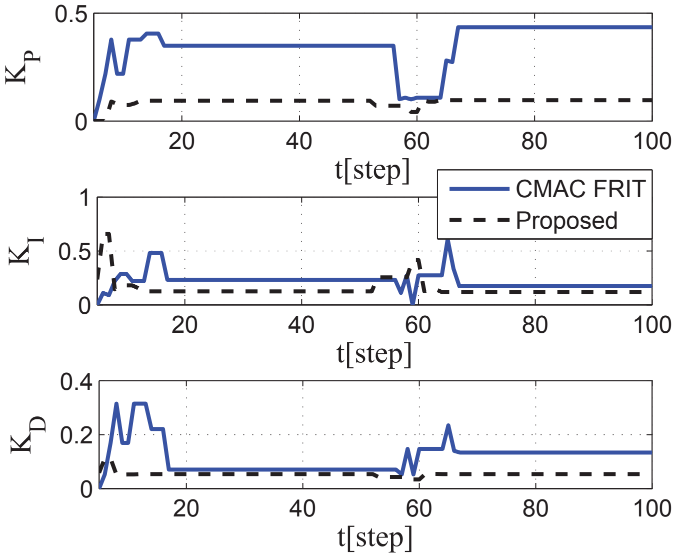

Figure 3.10: Trajectories of PID gains for untrained reference signals using proposed method and CMAC-FRIT

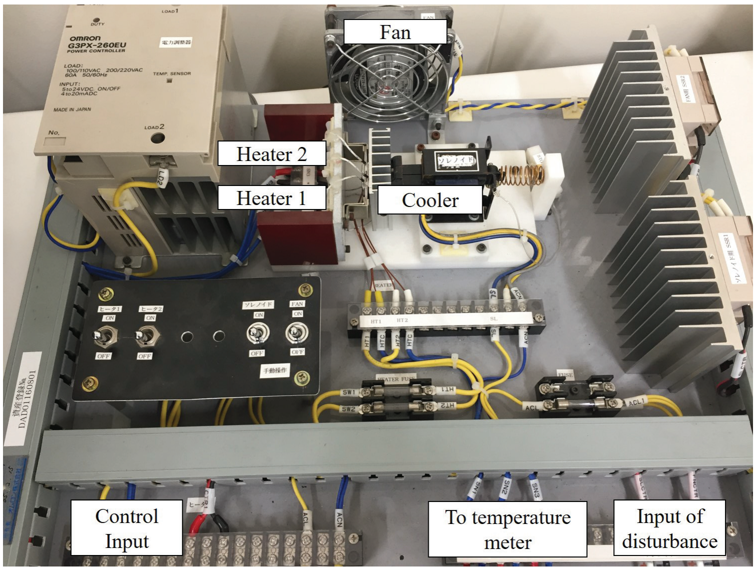

Figure 3.11: Photograph of an experimental temperature control system

earity for the controlled objective during control process. In some controlled objective in industrial control process, there is a cooler for a system, if the temperature changes within some range, the cooler is not working and if the temperature changes out of the set range, the cooler starts working. This experiment aims to simulate a such situation. For this experiment, the cooler does not work for the first value of reference signal and works for the second value of reference signal. The reference signals are decided as:

$$r(t) = \begin{cases} 40(1 \leq t \leq 300) \quad (\text{with} \quad \text{cooler}) \\ 55(301 \leq t \leq 600) \quad (\text{without} \quad \text{cooler}) \end{cases} . \tag{3.15}$$

To collect a set of closed-loop data, at first, CHR method is used to calculate a set of PID gains. The PID gains are $K_P = 5.353$, $K_I = 0.102$, and $K_D = 15.141$. The result shown in Fig.3.13 demonstrates the control ability of such set of PID controller gains. Based on the collected closed-loop date, CMAC and HC-CMAC are trained in an off-line manner. Some designed parameters for two methods are given in Table. 3.3.

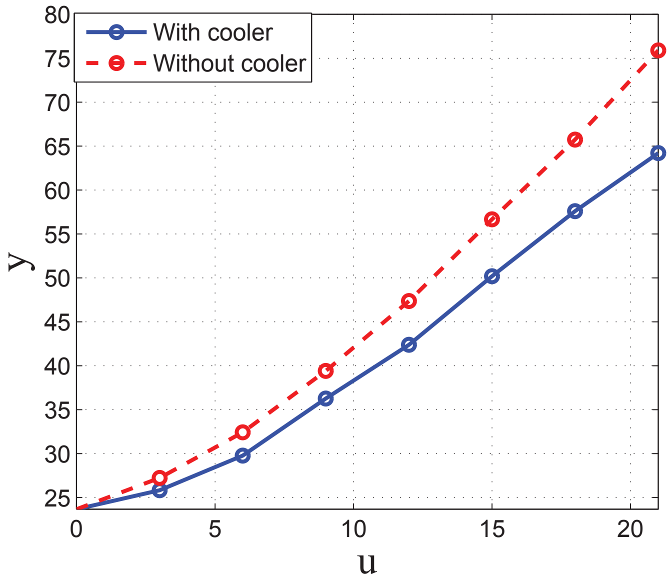

Figure 3.12: Static properties of the temperature control system

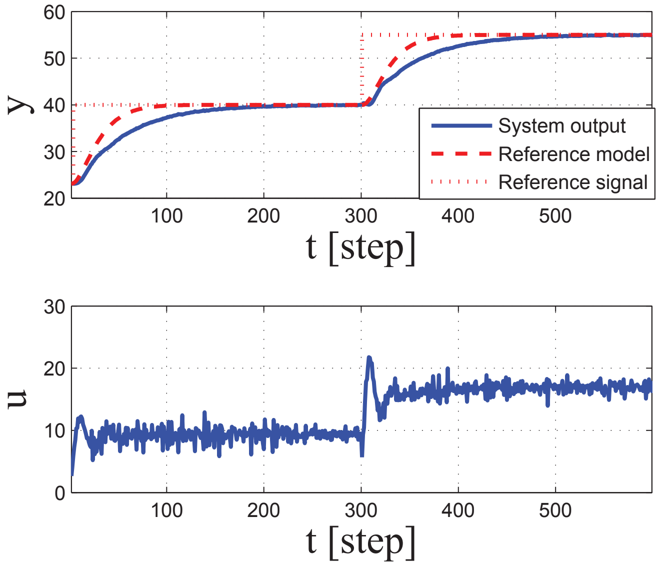

Figure 3.13: Experimental control performance using CHR method

The control performance when using the HC-CMAC-FRIT-PID controller is shown in Fig. 3.14and the control performance by using CMAC-

Table 3.3: Coefficients

Learning coefficients	η_P=0.01 η_I=0.003 η_D=0.01
Sampling time	T_S=1
Rise time	σ=25
Damping oscillation	δ=0

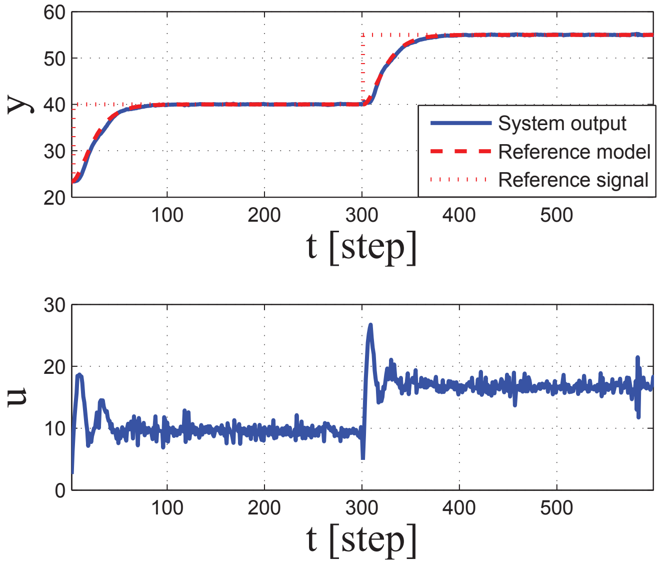

Figure 3.14: Experimental control performance using proposed method

FRIT-PID controller is presented in Fig. 3.15. For the HC-CMAC-FRIT-PID controller, a HC-CMAC with three weight tables include two, three, and four labels for each is selected. For the CMAC-FRIT-PID controller, a CMAC with three weight tables and four labels for each is used. Through the results of HC-CMAC-FRIT-PID controller and the CMAC-FRIT-PID controller, without sacrificing control ability, the proposed method requires

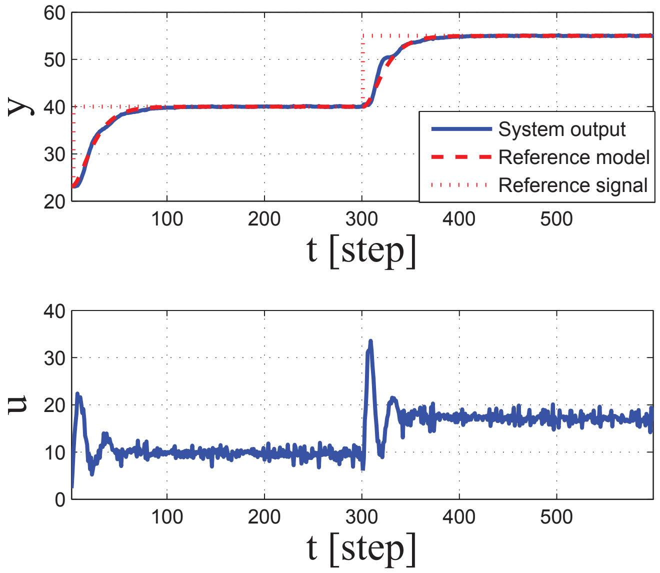

Figure 3.15: Experimental control performance using CMAC-FRIT-PID controller

less memory. The trajectories of the PID gains shown in Fig. 3.16 and Fig. 3.17 corresponds to control performances that are shown in Fig. 3.14 and Fig. 3.15.

In addition, some experimental results that shows generalization ability of the proposed method are discussed. The HC-CMAC-FRIT-PID controller and the CMAC-FRIT-PID controller which are trained for the previous reference signals are applied to control untrained reference signals. The untrained reference signals are designed as:

$$r(t) = \begin{cases} 45(1 \leq t \leq 300) \quad (\text{with} \quad \text{cooler}) \\ 65(301 \leq t \leq 600) \quad (\text{without} \quad \text{cooler}) \end{cases} . \tag{3.16}$$

Fig. 3.18 and Fig. 3.19 show the results of control performances for untrained signals of the HC-CMAC-FRIT-PID controller and the CMAC-FRIT-PID controller. The results present that the proposed method has a better

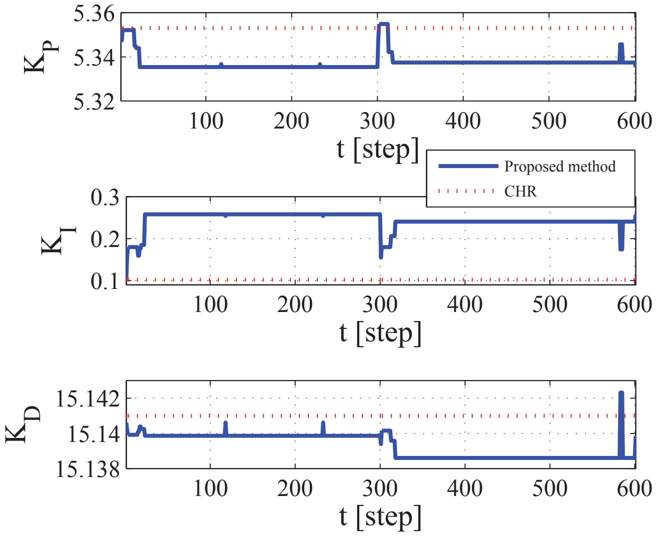

Figure 3.16: Trajectories of controller gains for HC-CMAC-FRIT-PID controller

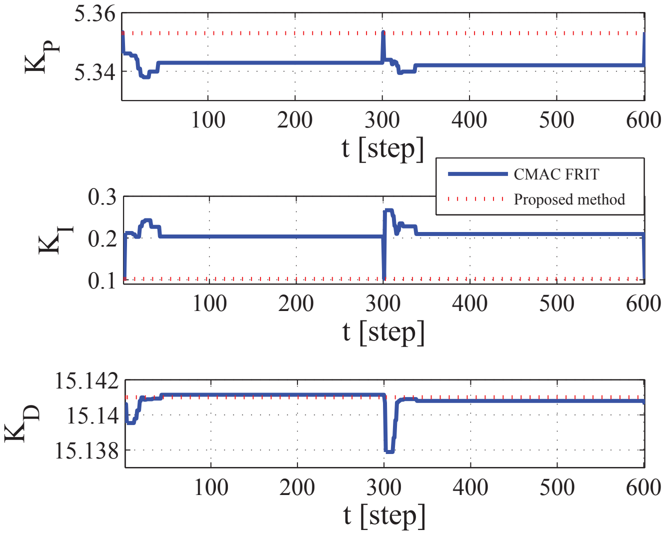

Figure 3.17: Trajectories of controller gains for CMAC-FRIT-PID controller

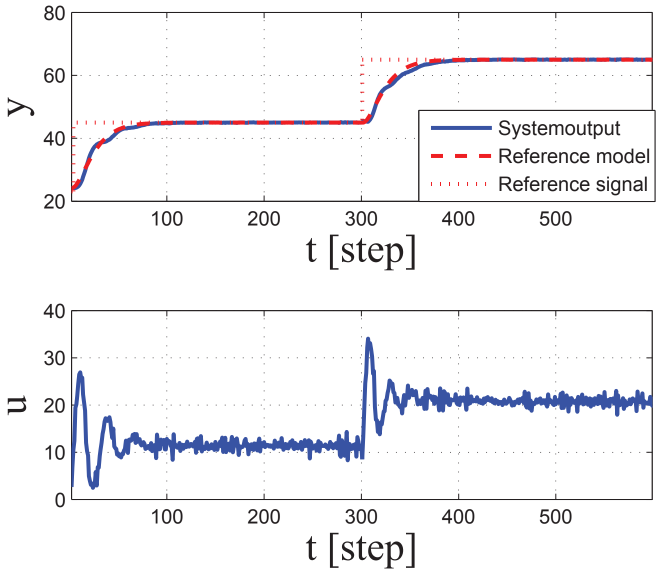

Figure 3.18: Experimental control performance using proposed method for untrained signals

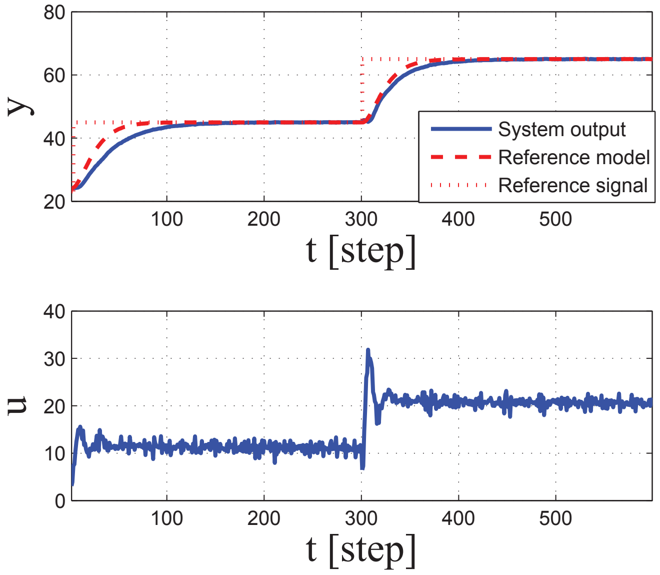

Figure 3.19: Experimental control performance using CMAC-FRIT-PID controller for untrained signals

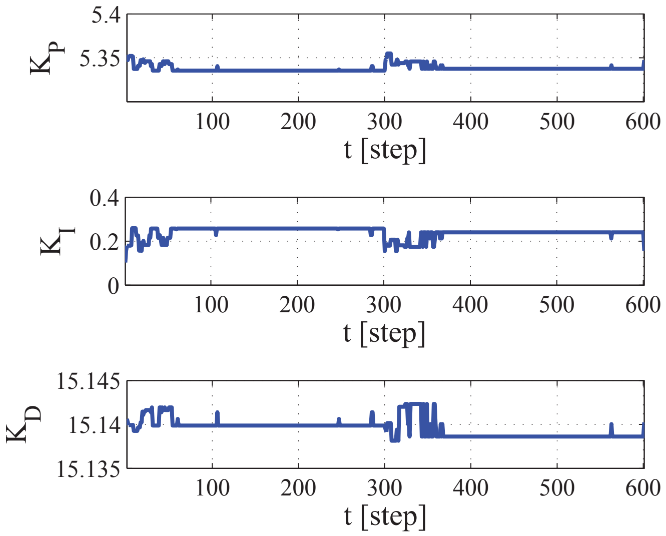

Figure 3.20: Trajectories of controller gains using proposed method for untrained signals

control ability for untrained reference signals, it verifies the proposed method has a superior generalization ability. The trajectories of the PID gains for Fig. 3.18 and Fig 3.19 are presented in Fig. 3.20 and Fig. 3.21, respectively.

Through the explanations and results above, it is verified that the proposed method achieves the desired control ability with lower memory requirements and the propsoed method has a superior generalization ability.

3.5 Conclusions

In this chapter, a HC-CMAC based PID controller and its off-line learning method is introduced. In the proposed structure of HC-CMAC, the numbers of labels for each weight table are determined individually. By setting different numbers of labels included in each weight table, the HC-CMAC

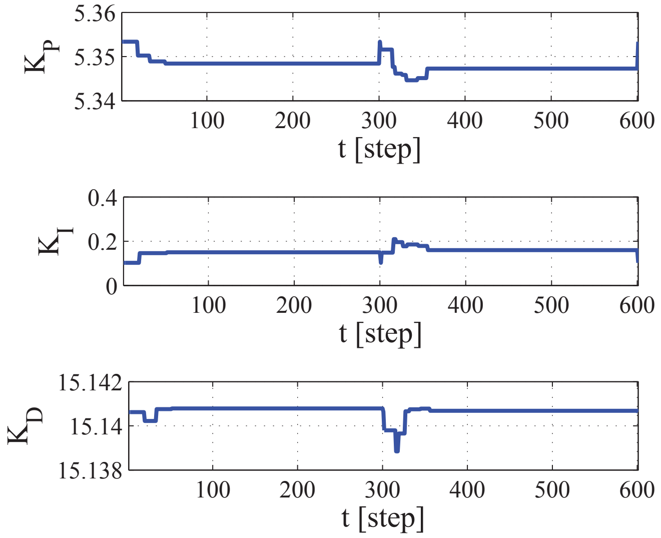

Figure 3.21: Trajectories of controller gains using CMAC-FRIT-PID controller for untrained signals

balances learning accuracy, generalization ability, and memory requirement. Hierarchical clustering method is used to achieve the user-specified number of labels for each weight table, so that the input space is adaptively quantized. The combining of FRIT and HC-CMAC enables the HC-CMAC tuns its weights in an off-line manner. From some simulation and experimental results, the proposed scheme demands lower memory and it provides a superior generalization ability than the CMAC-FRIT-PID controller.

3.6 Appendix

The followings are the partial differential of Eq.3.11 in detail:

$$\frac{\partial J(t+1)}{\partial K_{P,I,D}(t)} = \frac{\partial J(t+1)}{\partial y_r(t+1)} \frac{\partial y_r(t+1)}{\partial \tilde{r}(t)} \frac{\partial\ \tilde{r}(t)}{\partial K_{P,I,D}(t)} \tag{3.17}$$

where,

$$\frac{\partial J(t+1)}{\partial y_r(t+1)} = -(y_0(t+1) - y_r(t+1)) \tag{3.18}$$

$$\frac{\partial y_r(t+1)}{\partial \tilde{r}(t)} = 1 + p_1 + p_2 \tag{3.19}$$

$$\frac{\partial \tilde{r}(t)}{\partial K_P(t)} = \frac{(y_0(t) - y_0(t-1))K_I(t)}{K_I(t)^2} \tag{3.20}$$

$$\frac{\partial \tilde{r}(t)}{\partial K_I(t)} = \frac{-\Delta u_0(t) - K_P(t)\Delta y_0(t) - K_D(t)\Delta^2 y_0(t)}{K_I(t)^2} \tag{3.21}$$

$$\frac{\partial \tilde{r}(t)}{\partial K_D(t)} = \frac{(y_0(t) - 2y_0(t-1) + y_0(t-2))K_I(t)}{K_I(t)^2} \tag{3.22}$$

Chapter 4

Design of HC-CMAC based Performance Driven PID Controller

4.1 Introduction

In the previous chapter, an off-line learning of HC-CMAC, PID gains are calculated on-line using HC-CMAC is introduced. In this chapter, the method is extended to tune controller parameters not only for transient state but also for steady state.

In industrial processes, it is necessary to produce high-quality products and to reduce the cost of energy. According to these requirements, the performance of system output is desired to track and maintains the reference signal ideally. Tracking ability of a controller is always reflected in the transient state, a fast response shows a desired tracking ability of a controller. The control ability in steady state of a controller is always a standard to judge whether a controller has an ideal ability on maintaining system output to reference signal, in some studies, a small variance of system output is desired. Thus, in order to obtain ideal control performance, when designing a controller both of the above two aspects should be taken into consideration.

In the steady state, noise, controller parameters cause variance of system output [41]. Some studies regard this variance as control performance assessment (CPA), hence, the control performance can be monitored in steady state[42] [43] [44]. In addition, some schemes considered designing control systems to minimize the variance of system output, based on CPA, have been conceived, however, these studies are limited to improving control performance in steady state [45] [46].

In previous study, a CMAC based performance-driven (PD) PID controller has been proposed, it tunes its gains for both transient and steady state in an off-line manner, its effectiveness is verified by using some simulation and experiments [47] [48]. This method achieves its learning process based on the partial learning ability of CMAC.

Moreover, in some industrial control processes, if a machine participates more than one production process, a few reference signals may be set to the machine due to different requirements. If a controller tunes its gains once a new reference signal is set, the applicability of controller decreases, hence, generalization ability for similar reference signals is necessary.

To meet above requirements, the HC-CMAC is utilized to create a HC-CMAC based performance driven PID controller. Because the HC-CMAC balances requirement of memory, learning accuracy and generalization ability of a CMAC network. Additionally, the fictitious reference iterative tuning (FRIT) is combined with the HC-CMAC, so that the learning of HC-CMAC-FRIT can be achieved in an off-line manner without system identification.

In learning process, to examine the fitness of HC-CAMC's weights, some criterion should be decided. When using FRIT, the criterion is always set by examining the difference between selected signals of closed-loop data and selected fictitious signals. In this study, for transient state, the criterion is chosen as the difference between system output of closed-loop data and fictitious system output. For steady state, the criterion is determined as the difference between signal of "control performance assessment (CPA)" and signal of "fictitious control performance assessment". If the criterion are minimized, the suitable weights of HC-CMAC can be calculated.

In the end of this chapter, some simulations and experiments of proposed HC-CMAC-PD-PID controller and conventional CMAC-PD-PID controller are compared.

4.2 HC-CMAC Performance Driven PID Controller

In this study, the selected PID control structure is given as:

$$\begin{aligned}\Delta u(t) &= K_I(t)e(t) - K_P(t)\Delta y(t) \\ &\quad -K_D(t)\Delta^2 y(t),\end{aligned} \tag{4.1}$$

$$e(t) := r(t) - y(t). \tag{4.2}$$

When the suitable controller gains are calculated by using proposed method, the user desired control performance can be obtained.

(4.2) is further expanded as:

$$u(t) = \frac{C(1)}{\Delta}r(t) - \frac{C(z^{-1})}{\Delta}y(t), \tag{4.3}$$

$$C(z^{-1}) = K_I + K_P\Delta + K_D\Delta^2. \tag{4.4}$$

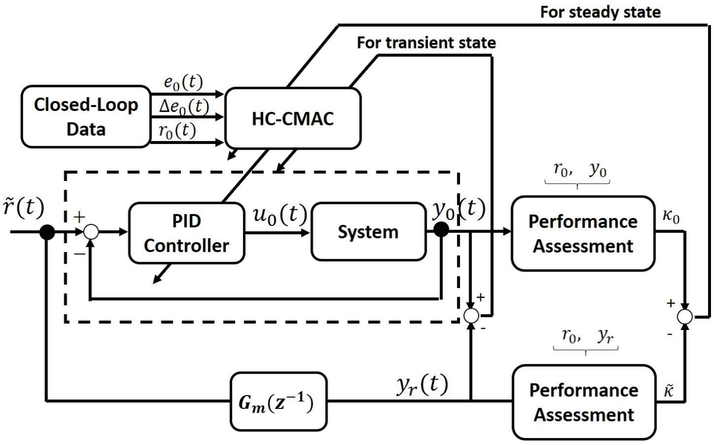

Figure 4.1: Learning process of proposed method

PID controller gains are generated from HC-CMAC according to:

$$K_{P,I,D}(t) = \sum_{h=1}^{K} W_{P,I,D,h}, \tag{4.5}$$

where h=1,2,...,K, K is the total number of the weight tables in the HC-CMAC. $W_{P,I,D,h}$ are the weights of HC-CMACs.

4.2.1 Learning Process

The off-line learning process of HC-CMAC-FRIT-PID controller is introduced in this section. The figure demonstrated in Fig. 4.1 gives a block diagram of the proposed method's learning process. In figure, the signals $u_0(t)$, $y_0(t)$ are closed-loop data, they denote system input data, and system output data, respectively, these data can be collected by any methods. In this chapter, the closed-loop data are collected by a PID controller with fixed controller gains. Equations (4.1) - (4.3) explain the calculation of fictitious reference signal $\tilde{r}(t)$:

$$\tilde{r}(t) = \frac{C(z^{-1})y_0(t) - \Delta u_0(t)}{C(1)}. \tag{4.6}$$

Fictitious reference model $G_m(z^{-1})$ is designed based on the equations in reference [6]:

$$P(z^{-1}) = 1 + p_1 z^{-1} + p_2 z^{-2} \tag{4.7}$$

$$, \left.\begin{array}{l} \rho := \frac{T_S}{\sigma} \\ \mu := 0.25(1-\delta) + 0.51\delta \\ p_1 = -2\exp(-\frac{\rho}{\mu})\cos(\frac{\sqrt{4\mu-1}}{2\mu}\rho) \\ p_2 = \exp(-\frac{\rho}{\mu}) \end{array}\right\}, \tag{4.8}$$

$$G_m(z^{-1}) = \frac{z^{-(k+1)}P(1)}{P(z^{-1})}, \tag{4.9}$$

$$\begin{aligned} y_r(t) &= G_m(z^{-(k+1)})\tilde{r}(t) \\ &= \frac{z^{-(k+1)}(1+p_1+p_2)}{1+p_1 z^{-1} + p_2 z^{-2}}\tilde{r}(t). \end{aligned} \tag{4.10}$$

Parameters σ and δ in (4.8) describe rise time and damping property of reference model. Range of parameter δ set as $0 \leq \delta \leq 2.0$. By setting these parameters, user desired reference model can be determined. The parameter represents sampling time is T_s. Approximated time-delay is k . y_r denotes fictitious system output in (4.10). For transient state, by minimizing the difference between y_0 and y_r the suitable weights of HC-CMAC can be calculated, thus, the controller parameters that enable system output tracking the reference model are obtained. For transient state, the criterion is decided as:

$$J_t(t) = |y_0(t) - y_r(t)| \tag{4.11}$$

In steady state, weights are updated to minimize the difference between "CPA" κ_0 and "fictitious CPA" $\tilde{\kappa}$. The calculation of κ is explained in detail:

A controlled autoregressive and integrated moving average model (CARIMA model) is used to describe a control objective. Because in industrial applications, the noise of a control objective is always non-stationary, and the model is not only limited for describing a linear system, but also the model can be used to locally linearized a nonlinear system at a particular operation point [49]. A CARIMA model is given in the following equation:

$$A(z^{-1})y(t) = z^{-(k+1)}B(z^{-1})u(t) + \frac{\xi(t)}{\Delta} \tag{4.12}$$

$$\left.\begin{array}{l} A(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a}, \\ B(z^{-1}) = b_0 + b_1 z^{-1} + \cdots + b_{n_b} z^{-n_b}, \end{array}\right\} \tag{4.13}$$

where, $\xi(t)$ is a white Gaussian noise, it has zero mean and variance σ_v^2. n_a and n_b are orders of $A(z^{-1})$ and $B(z^{-1})$.

The proposed method uses minimum variance control(MVC) algorithm. Cost function can be set as:

$$J \quad = \quad \mathcal{E}[\phi^2(t+k+1)], \tag{4.14}$$

where $\mathcal{E}[\bullet]$ denotes expectation $\phi(t+k+1)$ is defined according to the following equation, it is the difference of system output and reference model:

$$\phi(t+k+1) \quad := \quad P(z^{-1})y(t+k+1) - P(1)r(t) \tag{4.15}$$

Some Diophantine equation are introduced:

$$P(z^{-1}) = \Delta A(z^{-1})E(z^{-1}) + z^{-(k+1)}F(z^{-1}) \tag{4.16}$$

$$E(z^{-1}) = 1 + e_1 z^{-1} + \cdots + e_k z^{-k} \tag{4.17}$$

$$F(z^{-1}) = f_0 + f_1 z^{-1} + \cdots + f_{n_a} z^{-n_a} \tag{4.18}$$

where $P(z^{-1})$ is explained in (4.7). From appendix, (4.15) is derived as:

$$\begin{aligned} \phi(t+k+1) &= \frac{P(z^{-1})}{T(z^{-1})}\xi(t+k+1) && (4.19) \\ &= E(z^{-1})\xi(t+k+1) + S(z^{-1})\xi(t) && \\ & && (4.20) \end{aligned}$$

$$T(z^{-1}) := \Delta A(z^{-1}) + z^{-(k+1)}B(z^{-1})C(z^{-1}) \tag{4.21}$$

$$S(z^{-1}) := \frac{F(z^{-1}) - B(z^{-1})C(z^{-1})E(z^{-1})}{T(z^{-1})} \tag{4.22}$$

From (4.20), controller does not influence $E(z^{-1})$. Therefore, to obtain minimum variance control performance, $S(z^{-1})$ should equals 0. The optimized controller $C_{opt}(z^{-1})$ can be derived as:

$$C_{opt}(z^{-1}) = \frac{F(z^{-1})}{E(z^{-1})B(z^{-1})} \tag{4.23}$$

When ideal controller is utilized to control the controlled objective, $\phi_{min}(t+k+1)$ can be calculated as:

$$\phi_{min}(t+k+1) = E(z^{-1})\xi(t+k+1) \tag{4.24}$$

In (4.14), from (4.20), the criterion J should be:

$$\begin{aligned} J &= \mathcal{E}[\phi(t+k+1)^2] \\ &= \mathcal{E}[\{E(z^{-1})\xi(t+k+1) + S(z^{-1})\xi(t)\}^2] \end{aligned} \tag{4.25}$$

Since $\xi(t)$ is white noise, (4.25) can be rewritten as:

$$\begin{aligned} J &= \mathcal{E}[\{E(z^{-1})\xi(t+k+1)\}^2] \\ &\quad +\mathcal{E}[\{S(z^{-1})\xi(t)\}^2] \qquad (4.26) \\ &= J_{min} + J_0 \qquad (4.27) \end{aligned}$$

where

$$J_{min} = \mathcal{E}[\{E(z^{-1})\xi(t+k+1)\}^2] \qquad (4.28)$$

$$J_0 = \mathcal{E}[\{S(z^{-1})\xi(t)\}^2] \qquad (4.29)$$

From above, $J_0 = 0$ and J=J_{min} when ideal controller $C_{opt}(z^{-1})$ is utilized, .

The MVC-based control performance assessment is build as:

$$\kappa := \frac{J_{min}}{J_{min} + J_0} = 1 - \frac{J_0}{J_{min} + J_0} \qquad (4.30)$$

The ideal controller can be calculated when J_0=0, thus, when κ closes to 1, the control performance is a “desired control” and when κ closes to 0, the control performance is an “undesired control”.

In order to calculate J_{min}, the approach in reference [50] is utilized to estimate the parameter of polynomial $E(z^{-1})$. By using closed-loop data, it is able to calculate control performance assessment index κ.

$$\phi(t) - \bar{\phi} = \epsilon(t) + \sum_{j=0}^{m} \alpha_j \{\phi(t-k-l) - \bar{\phi}\} \qquad (4.31)$$

$$\epsilon(t) := E(z^{-1})\xi(t), \qquad (4.32)$$

where $\bar{\phi}$ is the average value of $\phi(t)$; α_j denotes autoregressive parameter, its order is$m$. Using M data, α_j is able to be calculated through the least squares method, the formulas to calculate α_j are given as:

$$\tilde{\boldsymbol{P}}(t) = \boldsymbol{Y}(t)\boldsymbol{\alpha}(t) + \boldsymbol{\Xi}(t) \qquad (4.33)$$

$$\tilde{\phi}(t) := \phi(t) - \bar{\phi} \tag{4.34}$$

$$\tilde{\boldsymbol{P}}(t) = [\tilde{\phi}(t), \tilde{\phi}(t-1), \cdots, \tilde{\phi}(t-M+1)]^{\mathrm{T}} \tag{4.35}$$

$$\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_n]^{\mathrm{T}} \tag{4.36}$$

$$\boldsymbol{\Xi}(t) = [\epsilon(t), \epsilon(t-1), \cdots, \epsilon(t-M+1)]^{\mathrm{T}} \tag{4.37}$$

$$\boldsymbol{Y}(t) = \begin{bmatrix} \tilde{\phi}(t-k-1) & \cdots & \tilde{\phi}(t-k-n) \\ \tilde{\phi}(t-k-2) & \cdots & \tilde{\phi}(t-k-n-1) \\ \vdots & \ddots & \vdots \\ \tilde{\phi}(t-k-M) & \cdots & \tilde{\phi}(t-k-n-M+1) \end{bmatrix} \tag{4.38}$$

From following equations, $\boldsymbol{\alpha}(t)$ can be determined:

$$\boldsymbol{\alpha}(t) = (\boldsymbol{Y(t)}^{\mathrm{T}}\boldsymbol{Y(t)})^{-1}\boldsymbol{Y(t)}^{\mathrm{T}}\tilde{\boldsymbol{P}}\boldsymbol{(t)} \tag{4.39}$$

κ is obtained using the following formulas:

$$\kappa = \frac{\{\tilde{\boldsymbol{P}}\boldsymbol{(t)} - \boldsymbol{Y(t)}\boldsymbol{\alpha}(t)\}^{\mathrm{T}}\{\tilde{\boldsymbol{P}}\boldsymbol{(t)} - \boldsymbol{Y(t)}\boldsymbol{\alpha}(t)\}}{\tilde{\boldsymbol{P}}\boldsymbol{(t)}^{\mathrm{T}}\tilde{\boldsymbol{P}}\boldsymbol{(t)}}. \tag{4.40}$$

From above introduction, calculation of κ is possible. When the following criterion is minimized, desired controller gains are acquired.

$$J_s(t) = |\kappa_0(t) - \tilde{\kappa}(t)| \tag{4.41}$$

MATLAB function “fminsearch” is utilized to minimize (4.11) and (4.41). MATLAB function “fminsearch” is a method to find minimum of unconstrained multivariable function, by using this method, optimal PID gains are obtained.

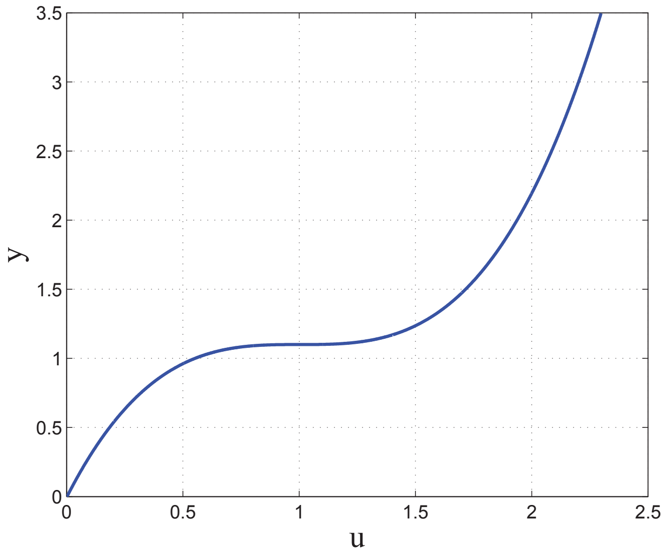

Figure 4.2: Static property of control objective

4.3 Simulation Results

To certificate the effectiveness of proposed method, some simulations are performed. The controlled object is the following Hammerstein model:

$$\begin{aligned} y(t) &= 0.6y(t-1) - 0.1y(t-2) + 1.2x(t-1) \\ &\quad -0.1x(t-2) + \frac{\xi(t)}{\Delta} \\ x(t) &= 1.5u(t) - 1.5u(t)^2 + 0.5u(t)^3 \end{aligned} \tag{4.42}$$

where, $\xi(t)$ is white noise with zero mean and variance σ_v^2=0.001^2. T_s, the sampling interval, is set as 1[s]. Fig. 4.2 shows the static property of controlled system. From the static property of system, around $y = 1.1$, the system has strong nonlinearity. PID gains calculated by using Chien, Hrones & Reswick (CHR) method is firstly applied to the controlled objective. The PID gains are $K_P = 0.233$, $K_I = 0.117$ and $K_D = 0.117$. Control performance using CHR method is demonstrated in Fig. 4.3 and Fig. 4.4.

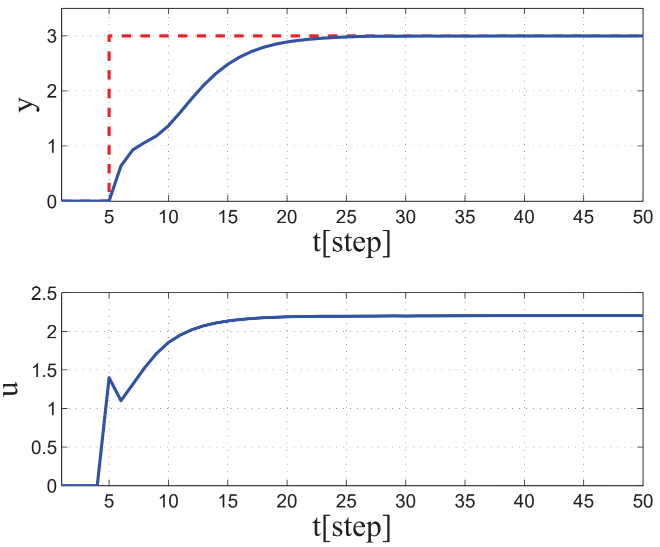

Figure 4.3: Control performance using CHR method in transient state

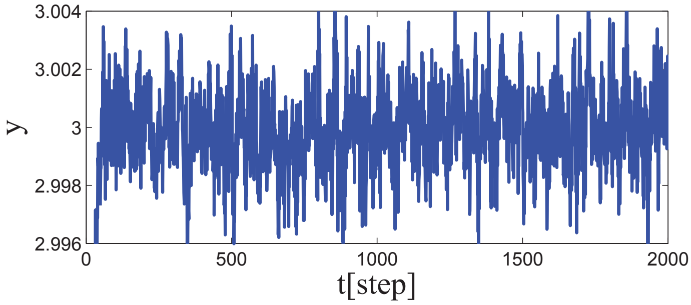

Figure 4.4: Control performance using CHR method in steady state

By using such control result as closed-loop data, the CMAC-PD-FRIT-PID controller and the HC-CMAC-PD-FRIT-PID controller are trained and they are used to control the controlled objective. Designed parameters of two methods are given in Table.4.1. The control performance using CMAC-PD-FRIT-PID in transient state and steady state are demonstrated in Fig. 4.5 and Fig. 4.6, respectively. The control performance shows that by using CMAC-PD-FRIT-PID controller, the control performance tracks reference

Table 4.1: Designed parameters of CMAC-PD-FRIT-PID and HC-CMAC-PD-FRIT-PID

Number of weight tables	3	3
Number of label	4, 4, 4	2, 3, 4
Rise-time	σ=2.03	σ=2.03
Damping property	δ=0	δ=0
Reference Value	r=3	r=3
Number of data	N=500	N=500
Order of autoregressive parameter	n=20	n=20
Memory requirement	4680 bytes	2376 bytes

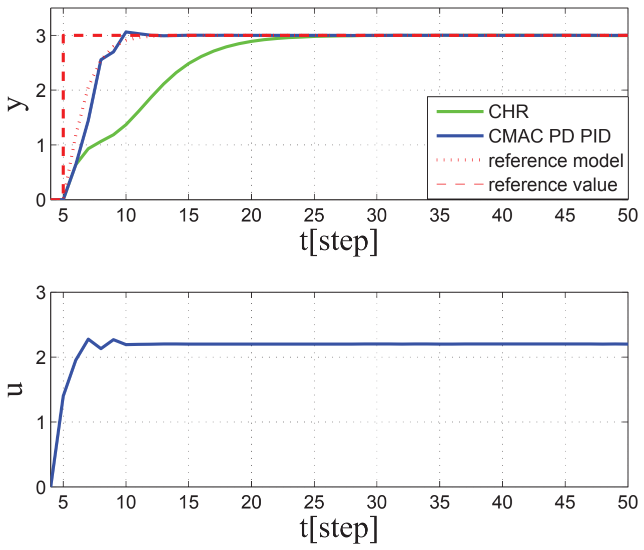

Figure 4.5: Control performance using CMAC-PD-FRIT-PID in transient state

model in transient state and in steady state the variance of system output becomes smaller. PID trajectories correspond to Figs. 4.5 and 4.6 are given in Figs. 4.7 and 4.8.

By applying the proposed method to controlled objective, Figs. 4.9 and 4.10 show the control performance for transient and steady state, respectively.

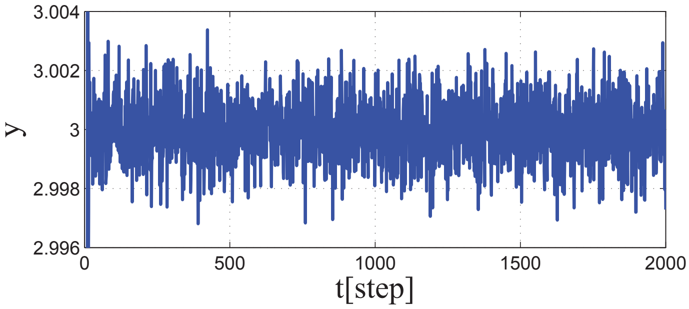

Figure 4.6: Control performance using CMAC-PD-FRIT-PID in steady state

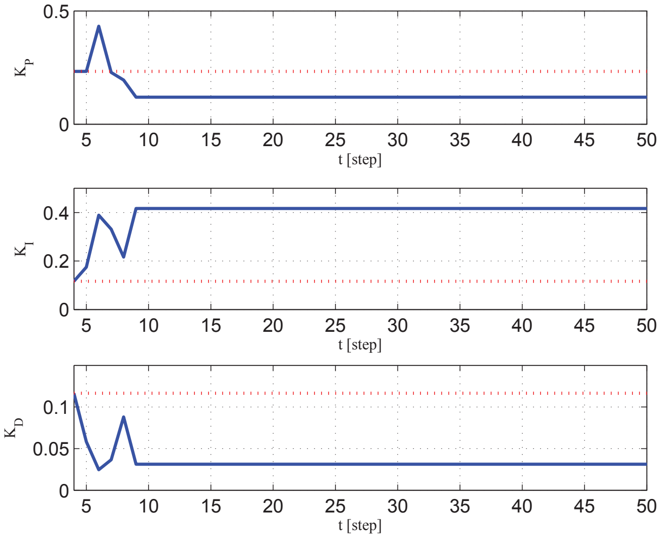

Figure 4.7: Trajectories of PID gains for CMAC-PD-FRIT-PID in transient state

Trajectories of PID gains corresponding to Figs. 4.9 and 4.10 are shown in Figs. 4.11 and 4.12. From results, the HC-CMAC-PD-FRIT-PID controller uses less memory, obtains similar results with CMAC-PD-FRIT-PID controller. Control performance assessment when using CHR, CMAC-PD-FRIT-PID and the proposed method are shown in Fig. 4.13. To verify the generalization ability of the proposed method, the control performances

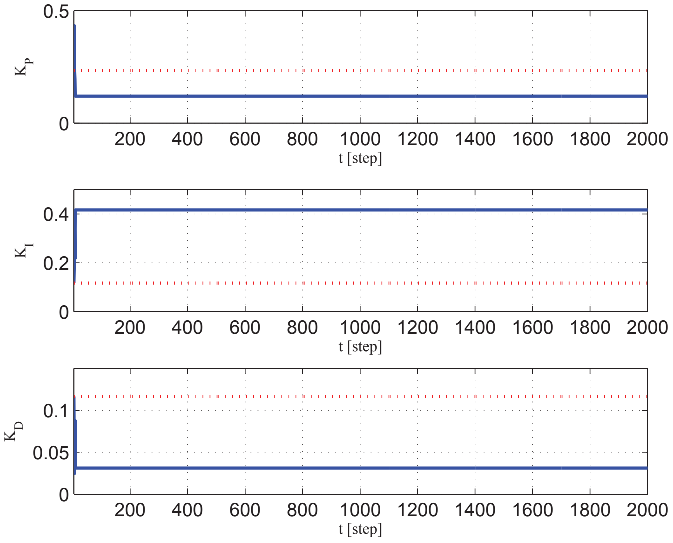

Figure 4.8: Trajectories of PID gains for CMAC-PD-FRIT-PID in steady state

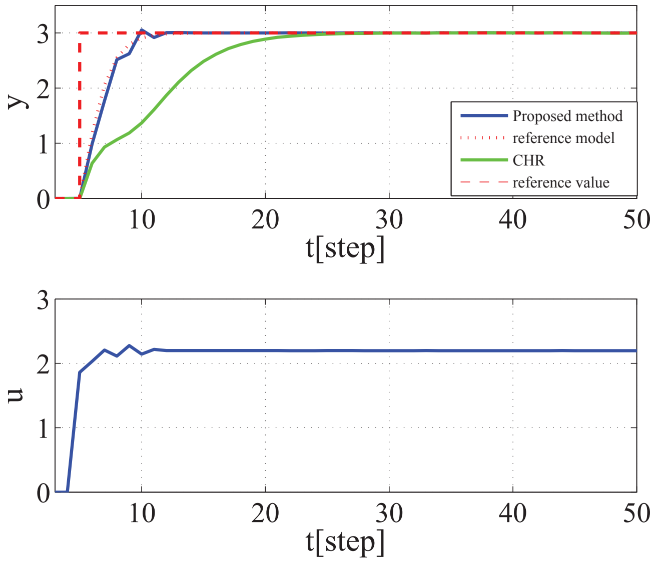

Figure 4.9: Control performance using proposed method in transient state

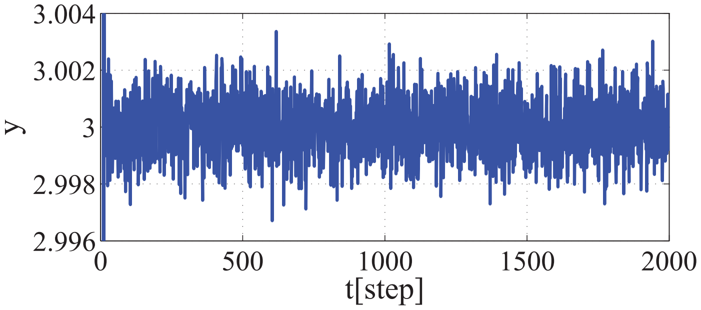

Figure 4.10: Control performance using proposed method in steady state

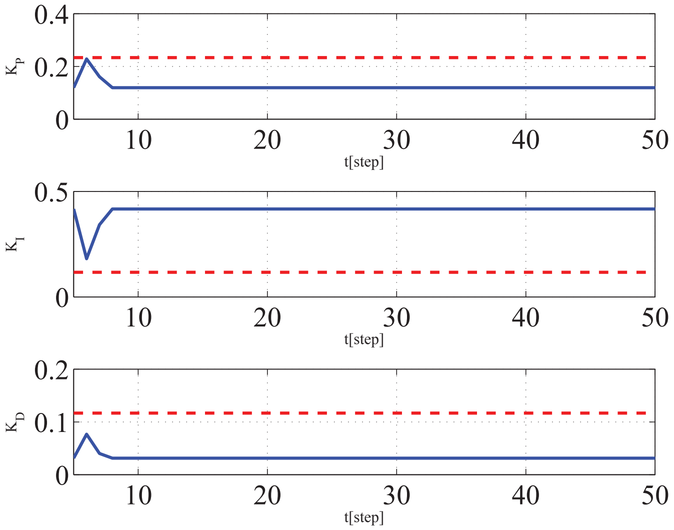

Figure 4.11: Trajectories of PID gains for proposed method in transient state

for untrained reference signal 2.3 are demonstrated in Figs. 4.14 and 4.15. From Fig. 4.14, the proposed method has a better control performance for untrained signal. The variance of Fig. 4.15 is 0.0175 for CMAC-PD-FRIT-PID controller and the variance is 0.0154 for the HC-CMAC-PD-FRIT-PID controller. It shows a better control ability of proposed method in steady state for untrained signal.

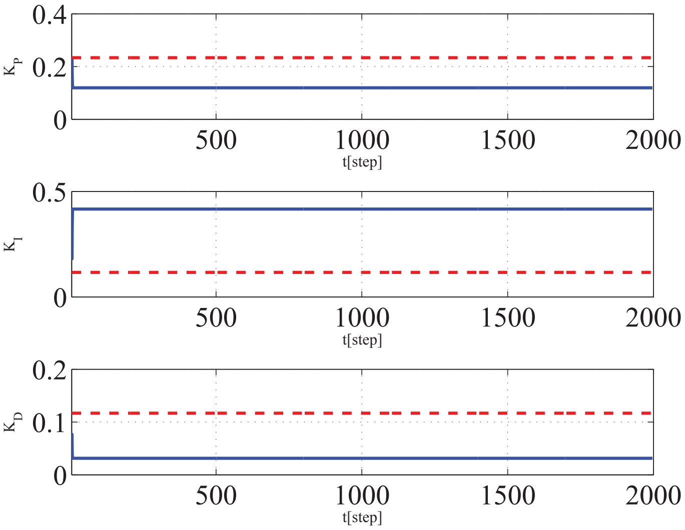

Figure 4.12: Trajectories of PID gains for proposed method in steady state

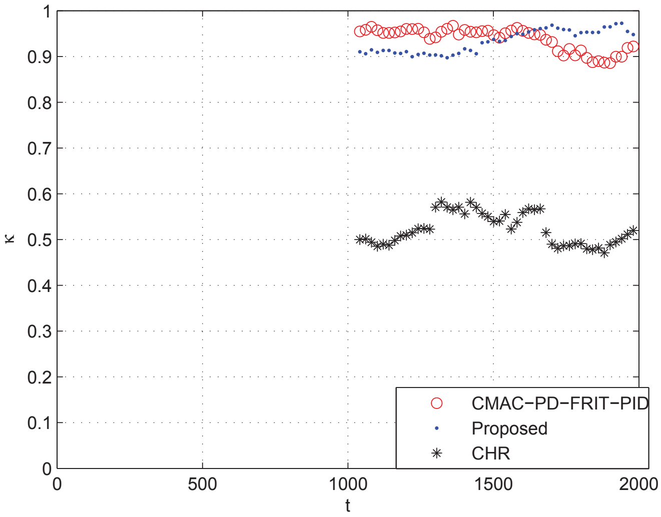

Figure 4.13: Control process assessment

4.4 Experiment Results

In this section, some experiment results are given, to give some comparisons to demonstrate the effectiveness of HC-CMAC-PD-FRIT-PID con-

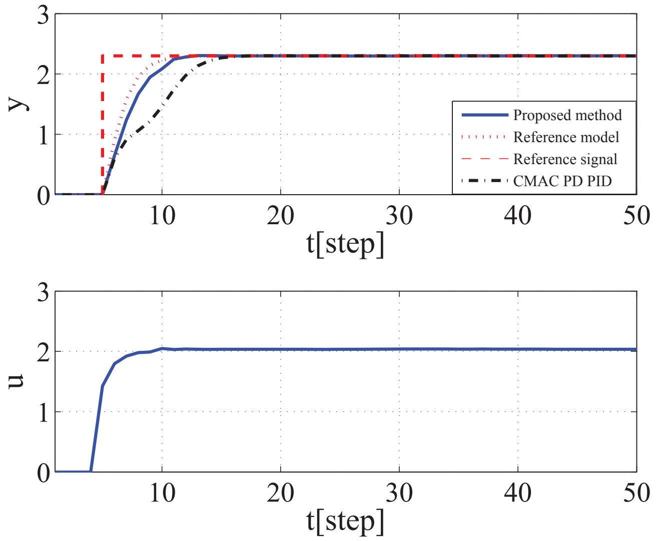

Figure 4.14: Comparison of control performance in transient state for the CMAC-PD-FRIT-PID controller and proposed method for untrained signal

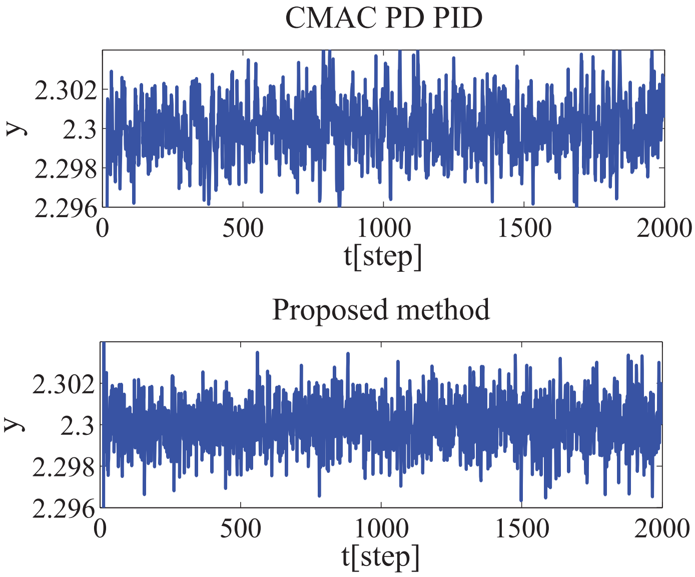

Figure 4.15: Comparison of control performance in steady state for the CMAC-PD-FRIT-PID controller and proposed method for untrained signal

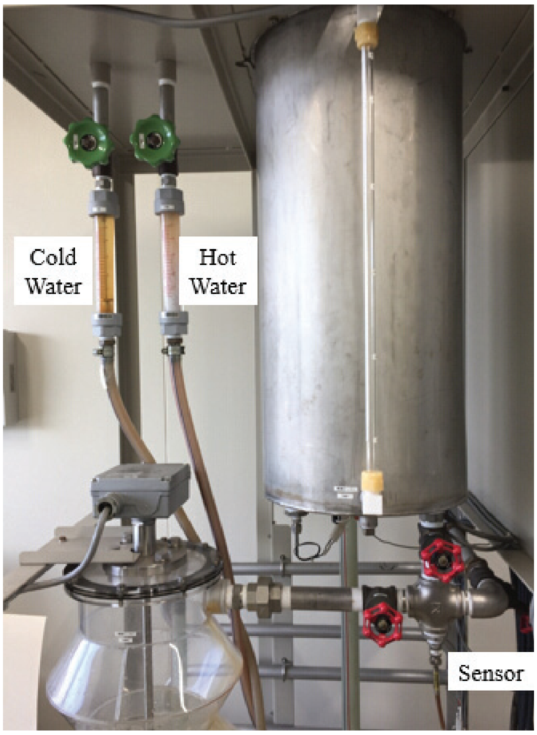

Figure 4.16: Temperature process

troller. Some methods are employed in a temperature control process. Fig.4.16 shows the equipment of the temperature process. The temperature of water is treated as system output $y(t)$ and the temperature signal can be measured by the sensor. Through the A/D converter, the system output signal is sent to the computer. The computer calculates the hot water's valve aperture as the input signal; the limitation of input signal is : $0\% \leq u(t) \leq 100\%$. The input of cold water is a constant value, it is set as 10% in this experiment.

The experimental comparisons are given in two parts, in the first part, a comparison of CMAC-PD-FRIT-PID controller and a PD-FRIT-PID controller is given, to explain the effectiveness of a CMAC-PD-FRIT-PID controller, in the second part, a comparison between CMAC-PD-FRIT-PID controller and a HC-CMAC-PD-FRIT-PID controller is given, to verify the pro-

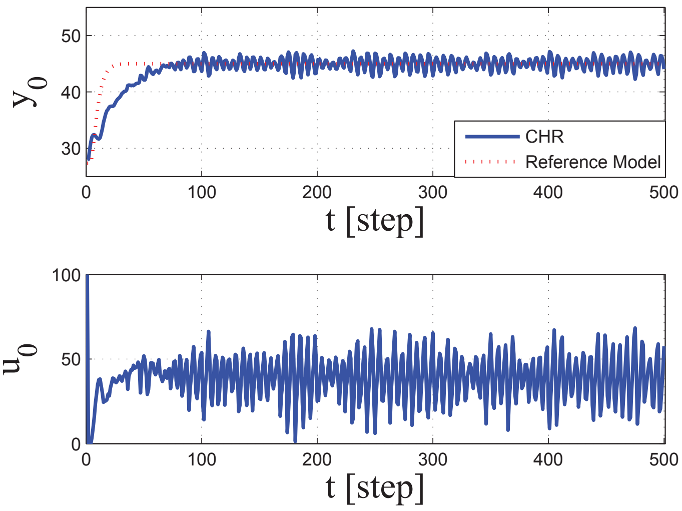

Figure 4.17: Control performance using CHR method

posed method is useful.

4.4.1 Comparison between a PD-FRIT-PID controller and CMAC-PD-FRIT-PID controller

The control performance by using the Chien, Hrones and Reswick method is shown in Fig.4.17. Additionally, the data is utilized as closed-loop data for the CMAC-PD-FRIT-PID controller. From the control performance, the system output can not track the reference model during the transient state and the variance of system output during steady state is undesired. The PID gains are calculated as:

$$K_P{=}5.95 \qquad K_I{=}0.32 \qquad K_D{=}8.93$$

As a comparison, a result by using the method mentioned in reference [41] is shown in Fig. 4.18, it is referred as PD-FRIT-PID method in this chapter, this method focus to improve the control performance in steady state, thus, the control performance can not track the reference model in

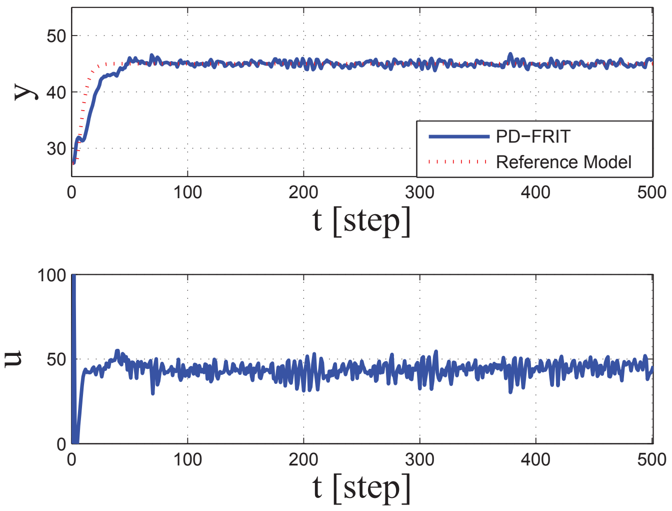

Figure 4.18: Control performance using PD-FRIT-PID method

Table 4.2: Designed parameters of CMAC-PD-FRIT-PID controller for experimental result

Number of weight tables	3
Number of label	3
Rise-time	σ=35
Damping property	δ=0
Reference Value	r=45
Number of data	N=300
Order of autoregressive parameter	n=20
Coefficient related to cost functions	β=0.1

transient state, in steady state the variance of system output is improved. The PID gains are calculated as:

$$K_P=3.76 \qquad K_I=0.38 \qquad K_D=7.74$$

The control performance by using the CMAC-PD-FRIT-PID controller is shown in Fig.4.19 and the designed parameters are summarized in Table 4.2.

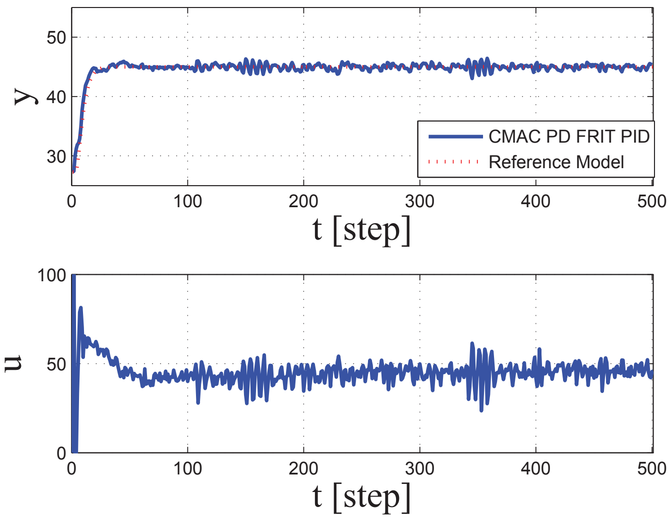

Figure 4.19: Control performance of CMAC-PD-FRIT-PID controller

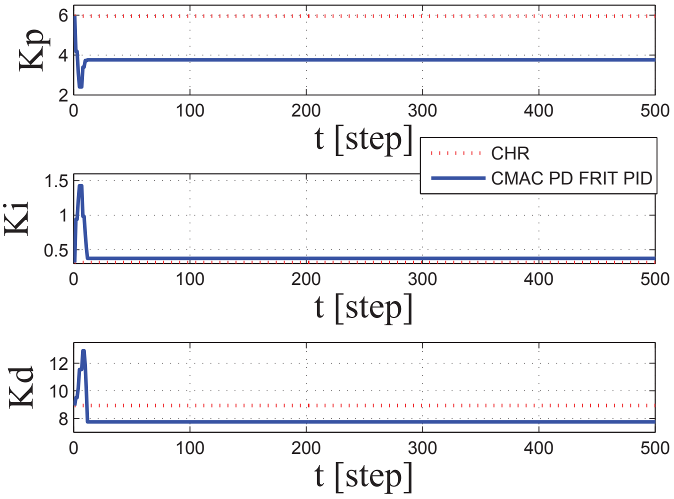

Figure 4.20: Trajectories of PID gains for the CMAC-PD-FRIT-PID controller

Trajectories of PID gains, for the CMAC-PD-FRIT-PID controller, are shown in Fig.4.20. By comparing Fig.4.17, Fig.4.19 and Fig.4.18, when the CMAC-PD-FRIT-PID controller is utilized, the control performances tracks

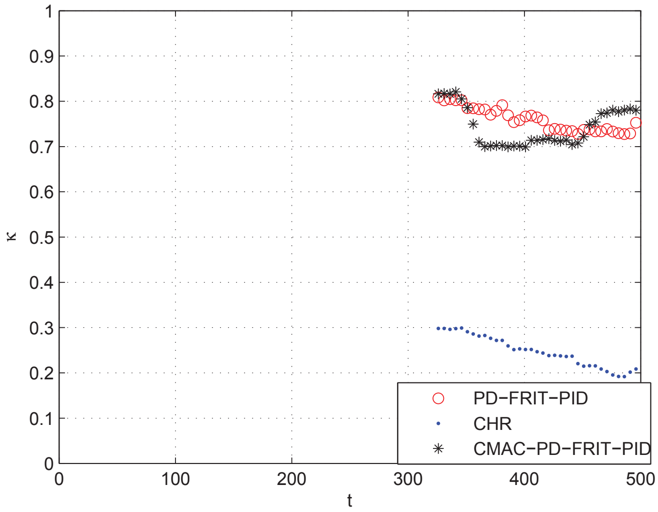

Figure 4.21: Comparison of κ and κ_0

reference model in transient state and in steady state the variance of system output is reduced. Fig.4.21 shows the control performance assessment using the CHR, PD-FRIT-PID and CMAC-PD-FRIT-PID controller. From the figure, the value of control performance assessment of the CMAC-PD-FRIT-PID controller and PD-FRIT-PID controller are between 0.7 and 0.8, the value of control performance assessment of the CHR method between 0.2 and 0.3. This demonstrates the control ability for the CMAC-PD-FRIT-PID controller is almost same with PD-FRIT-PID method and better than CHR method in steady state. From the above, the effectiveness of the CMAC-PD-FRIT-PID controller has been verified numerically by simulation ands experiments.

Table 4.3: Designed parameters of CMAC-PD-FRIT-PID and HC-CMAC-PD-FRIT-PID controllers

Number of weight tables	3	3
Number of label for each	3, 3, 3	1, 2, 3
Rise-time	σ=35	σ=35
Damping property	δ=0	δ=0
Reference Value	r=40	r=40
Number of data	N=300	N=300
Order of autoregressive parameter	n=20	n=20
Coefficient related to cost functions	β=0.1	β=0.1
Memory requirement	M=2484	M=1404

4.4.2 Comparison between CMAC-PD-FRIT-PID controller and HC-CMAC-PD-FRIT-PID controller

The comparison between HC-CMAC-PD-FRIT-PID controller and CMAC-PD-FRIT-PID controller is demonstrated in the following statement:

The designed parameters and memory requirement of two methods are given in Table. 4.3.

A set of PID gains calculated form CHR method is applied to control objective, to gather a set of closed-loop data which is shown in Fig. 4.22:

$$K_P=3.76 \qquad K_I=0.38 \qquad K_D=7.74.$$

Based on the closed-loop data, the HC-CMAC-PD-FRIT-PID and CMAC-PD-FRIT-PID controller are trained, the control performances of each method are given in Fig. 4.23 and Fig. 4.24 Form the control performances, it is known that both of the two methods tacks the reference signal, and the variance of system output in steady state are improved. Form Table. 4.3, it is known that proposed method achieves similar control performance with less memory requirement. The trajectories of PID gains for CMAC-PD-FRIT-PID controller is shown in Fig. 4.25 and the trajectories of PID gains for

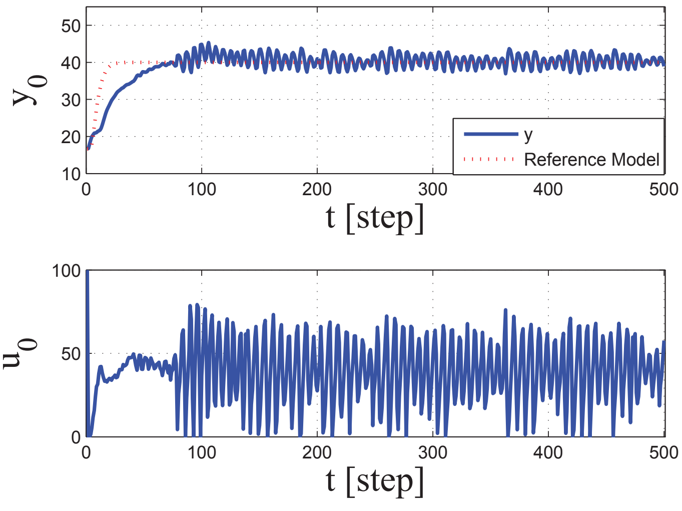

Figure 4.22: Closed-loop data

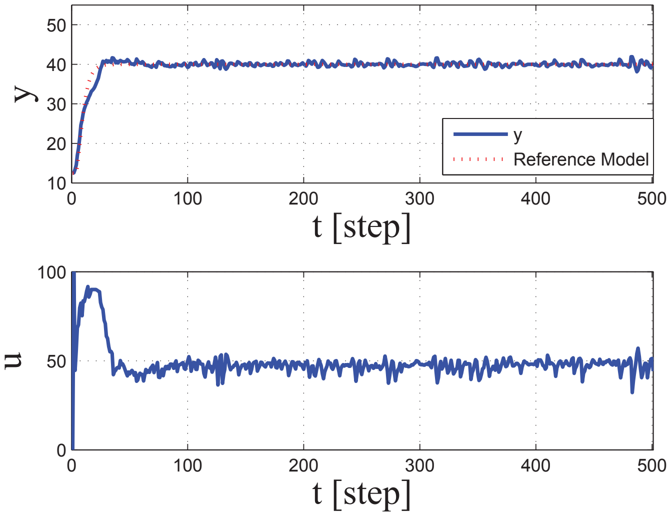

Figure 4.23: Control performance of CMAC-PD-FRIT-PID controller

HC-CMAC-PD-FRIT-PID controller is shown in Fig. 4.26

The generalization ability of two controllers are compared as follows:

The CMAC and HC-CMAC trained for reference signal 40 are then employed to control reference signal 30, the control performances are shown

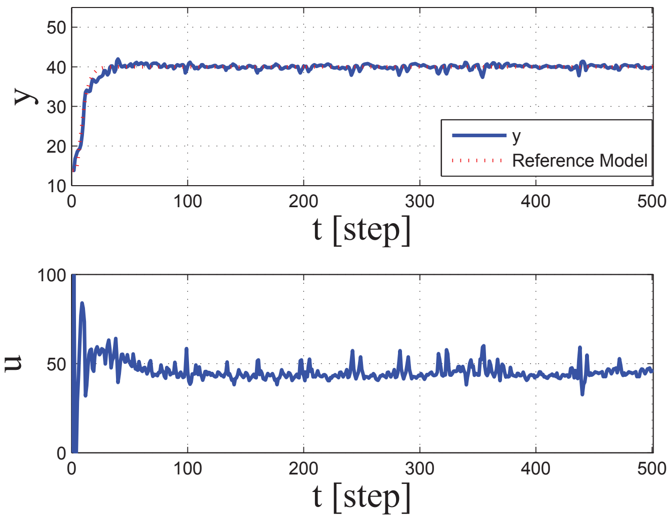

Figure 4.24: Control performance of HC-CMAC-PD-FRIT-PID controller

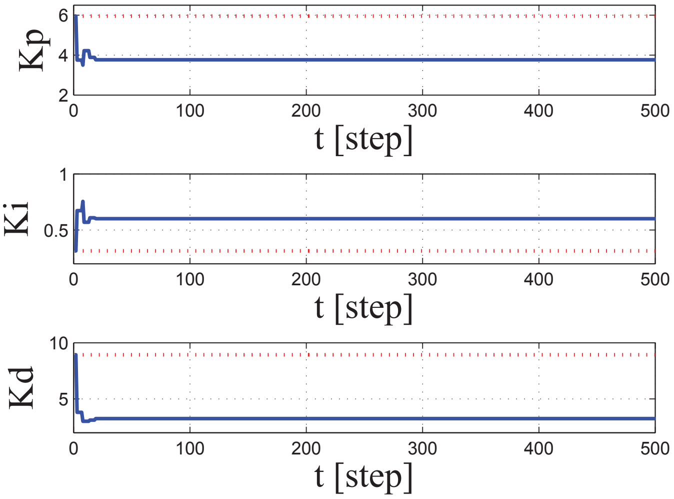

Figure 4.25: Trajectories of PID gains using CMAC-PD-FRIT-PID controller

in Fig. 4.27 and Fig. 4.28. In the transient state, the integral absolute error(IAE) of CMAC-PD-FRIT-PID is 85.32, the IAE for HC-CMAC-PD-PID controller is 45. It demonstrated that the HC-CMAC-PD-FRIT-PID

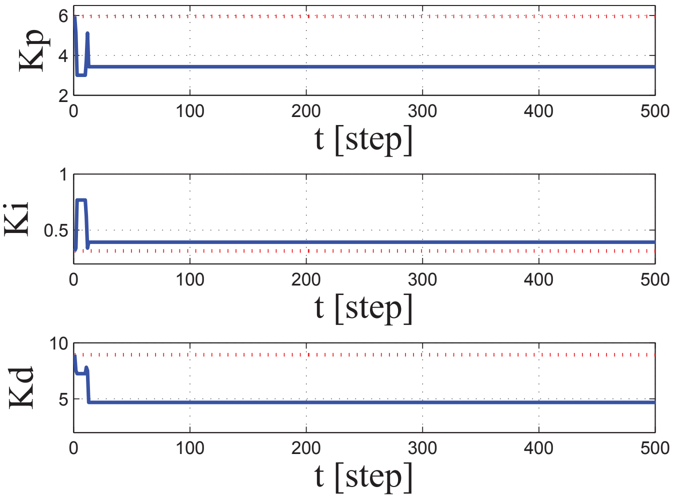

Figure 4.26: Trajectories of PID gains using HC-CMAC-PD-FRIT-PID controller

controller has a better tracking ability than the CMAC-PD-PID controller for untrained reference signal.

A figure shows the comparison of CPA for untrained reference signal is given in Fig. 4.29, it shows that the variance of system output in steady state by using the HC-CMAC-PD-FRIT-PID controller is smaller than the one for CMAC-PD-FRIT-PID controller.

4.5 Conclusions

This chapter has proposed a HC-CMAC-PD-FRIT-PID controller, the HC-CMAC enables a different number of labels for each weight table, thus, the balance of learning accuracy, memory requirement and generalization ability of CMAC can be achieved. Through some simulations and experiments, the CMAC-PD-FRIT-PID controller improves the control performance for both transient state and steady state, the HC-CMAC-PD-PID

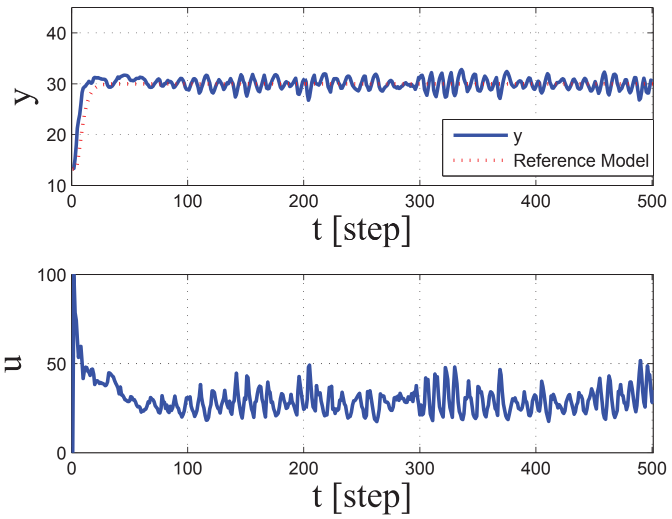

Figure 4.27: Control performance for untrained signal using CMAC-PD-FRIT-PID controller

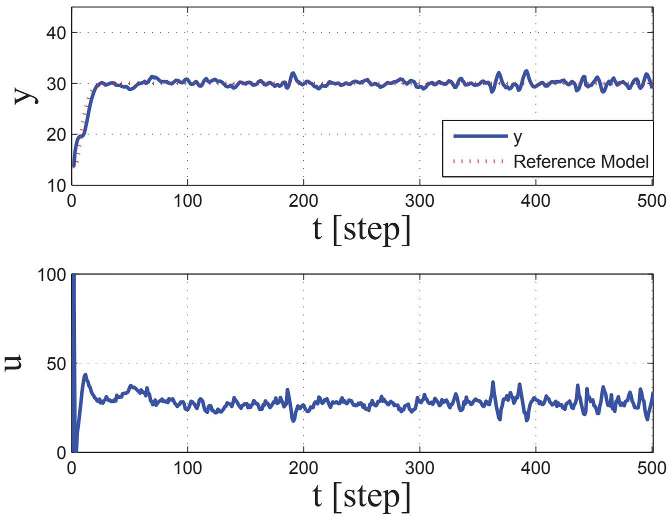

Figure 4.28: Control performance for untrained signal using HC-CMAC-PD-FRIT-PID controller

controller remains the advantage of CMAC-PD-FRIT-PID controller with less memory requirement and better generalization ability.

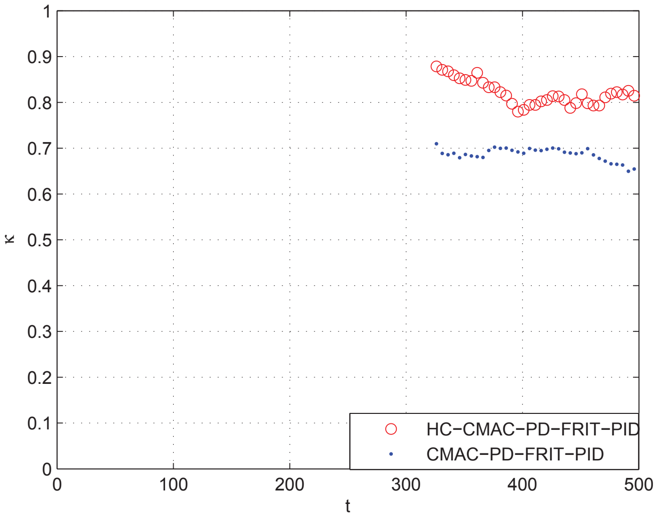

Figure 4.29: Comparison of CPA for untrained reference signal

4.6 Appendix

In the following equations, (z^{-1}) is omitted. From (4.3) and (4.12), the following equation is obtained by

$$\Delta Ay(t) \quad = \quad z^{-(k+1)}BC(1)r(t) - z^{-(k+1)}BCy(t) + \xi(t). \tag{4.43}$$

Based on (4.21), the following formula is derived:

$$y(t) = \frac{z^{-(k+1)}BC(1)r(t)}{T} + \frac{\xi(t)}{T}. \tag{4.44}$$

Substituting (4.44) into (4.15) yields

$$\begin{aligned} \phi(t+k+1) &= \frac{PBC(1)r(t)}{T} + \frac{P\xi(t+k+1)}{T} \\ &\quad -P(1)r(t) \\ &= \frac{PBC(1) - P(1)(\Delta A - z^{-(k+1)}BC)}{T}r(t) \\ &\quad +\frac{P\xi(t+k+1)}{T}. \end{aligned} \tag{4.45}$$

Considering the reference value is a constant value, it is known that $r(t)$=$z^{-1}r(t)$ =...=$z^{-(k+1)}r(t)$. On the right side of (4.45), the first term equals zero. Hence, (4.19) can be derived by

$$\phi(t+k+1) \;=\; \frac{P(z^{-1})}{T(z^{-1})}\xi(t+k+1)$$

In addition, from (4.16) and (4.21) the following equation can be calculated as

$$\begin{aligned} ET &= \Delta AE + z^{-(k+1)}BCE \\ &= (P - z^{-(k+1)}F) + z^{-(k+1)}BCE. \end{aligned} \tag{4.46}$$

Multiplying ET for both sides of (4.19) yields

$$ET\phi(t+k+1) = EP(z^{-1})\xi(t+k+1). \tag{4.47}$$

At last, substituting (4.46) into (4.47) and take (4.19) and (4.22) into consideration, the following formulas can be obtained:

$$\begin{aligned} \phi(t+k+1) &= E\xi(t+k+1) + \frac{F - BCE}{T}\xi(t) \\ &= E\xi(t+k+1) + S\xi(t). \end{aligned} \tag{4.48}$$

Chapter 5

Conclusions

In this dissertation, three kinds of Hierarchical Clustering CMAC based controllers have been proposed.

The proposed HC-CMAC is an optimized structure of a conventional CMAC, in the conventional CMAC, the memory requirement increases exponentially and the generalization ability of the network scarifies as the demand of learning accuracy increases, the proposed structure overcomes such drawbacks, it can balance the memory requirement, generalization ability and learning accuracy of a network.

In Chapter 2, an on-line tuning method of PID gains using HC-CMAC is introduced. Compare with the conventional CMAC PID controller, under the condition that they have similar control performances for a trained reference signal, the proposed scheme uses less learning cost and less memory, it has a better control ability for untrained reference signal. From the results, the HC-CMAC PID controller has a desired control performance for nonlinear system, and it is a more applicable controller.

In Chapter 3, a HC-CMAC FRIT PID controller is proposed. In this chapter, a detail of constructing of an off-line tuning scheme of PID gains is explained. The FRIT algorithm is utilized so that the modeling of a control objective is unnecessary and by using a set of closed-loop data, the learning of HC-CMAC is available. In such case, the time cost of on-line learning can be avoid. This method is compared with a CMAC FRIT PID controller, and some advantages are shown in detail.

In Chapter 4, the HC-CMAC is extended to be used to design a performance-driven controller. The HC-CMAC maintains the partial learning ability of CMAC network, it is possible to tune controller gains for different state of a control process. This HC-CMAC PD FRIT PID controller leads a fast response in transient state and decreases variance of system output in steady state. The learning of PID gains is also achieved in off-line manner. Some simulations and experiments are demonstrated to show the effectiveness of the controller. Some comparisons are given to explain the controller is more applicable.

From all the contents introduced in the dissertation, the effectiveness of HC-CMAC based PID controllers are verified. In this thesis, the controllers are designed without modeling of control objectives, this is an important feature, since in the application to real systems, some control objectives would be complex and sometimes modeling of them becomes very difficult.

In the future, the proposed HC-CMAC is considered to be constructed automatically, which means some user-specified parameters such as number of weight tables, number of labels for each weight table are going to be decided based on some conditions, for example, the requirement of memory, the requirement of learning accuracy and so on.

The HC-CMAC is constructed from data, its structure reflects a distribution of data, thus, applying the HC-CMAC to classification field is taken into consideration. Such thinking comes up with the highly developments of computer science, measurement of hardware and data processing techniques. By using these techniques, the gathering of large amount of data becomes available, almost everything can be reflected through data. Thus, when analyzing some problems, multi-dimensional database can be created, a multi-HC-CMAC that has classification ability can be constructed based on such database.

The HC-CMAC is also being considered to be utilized in the field of modeling. Especially, when the gathered data include nonlinear informations; the HC-CMAC is constructed adaptively by separating data into similar or dissimilar groups, this feature may helps the modeling for nonlinear system more accurate and efficient.

Bibliography

[1] T. Samad, Ed., “Perspectives in Control: New Concepts and Applications,” IEEE Press, NJ, 2001.

[2] K. Gurney, “An introduction to neural networks,” UCL Press, London, 1997.

[3] D. Kriesel, “A Brief Introduction to Neural Networks,” dkriesel.com, 2005.

[4] M. T. Hagan, H. B. Demuth, M. H. Beale,O. D. Jesús “Neural Network Design”, Martin Hagan, 2014.

[5] S. Haykin, “Neural Networks - A Comprehensive Foundation,” Pearson Education, 2005.

[6] M. T. Hagan, H. B. Demuth, “Neural networks for control,” Proceedings of the American Control Conference, 1999.

[7] K. Cheng “CMAC-based neuro-fuzzy approach for complex system modeling,” Neurocomputing, Vol. 72, pp. 1763-1774, 2009.

[8] W. Yu, F. Rodriguze, M. Moreno-Armendariz “Hierarchical Fuzzy CMAC for Nonlinear Systems Modeling,” IEEE Transactions on Fuzzy Systems, Vol. 16, No.5, pp. 1302-1314, 2008.

[9] M. Yeh, M. Leu “ART-type CMAC network classifier,” Neuroncomputing, Vol. 74, pp. 783-791, 2011.

[10] B. Li, D. Elliman “Fuzzy classification by a CMAC network,” Proceedings of 9th IEEE International Conference on Tools with Artificial Intelligence, November, 1997.

[11] M. Yeh “Single-input CMAC control system,” Neurocomputing, Vol. 70, pp. 2638-2644, 2007.

[12] H. Shiraishi, S. L. Ipri and D. -I. D. Cho “CMAC neural network controller for fuel-injection systems,” IEEE Transactions on Control Systems Technology, Vol. 3, No. 1, pp. 32-38, 1995.

[13] J. S. Albus, “A Theory of Cerebellar Function,” Mathematical Biosciences, No.10, pp. 25-61, 1971.

[14] J. S. Albus, “A New Approach to Manipulator Control: the Cerebellar Model Articulation Controller (CMAC),” Trans. ASME, Series G. Journal of Dynamic Systems, Measurement and Control, No.97, Vol 3, pp. 220-233, 1975.

[15] J. S. Albus, “Mechanisms of Planning and Problem Solving in the Brain,” Mathematical Biosciences, No.45, pp. 247-293, 1979.

[16] Xing, Frank Z. “A Historical Review of Forty Years of Research on CMAC,” eprint arXiv:1702.02277.

[17] D. Purves; et al. “Neuroscience,” Sunderland, Massachusetts U.S.A. 2004.

[18] Y. Liao, K. Koiwai, T. Yamamoto. ”Design of a hierarchical-clustering CMAC-PID controller.” 2017 International Joint Conference on Neural Networks (IJCNN 2017), Alaska, American, 2017, May.

[19] Y. Liao, K. Takuya, K. Koiwai, T. Yamamoto. ”Design of a performance-driven PID controller Using a Hierarchical-Clustering CMAC.” 2017 The 49th ISCIE International Symposium on Stochastic Systems Theory and Its Applications (SSS' 17), Hiroshima, Japan, 2017, November.

[20] C. Jia, X. Shan, Y. Cui, T. Bai, F. Cui, ”Modeling and Simulation of Hydraulic Roll Bending System Based on CMAC Neural Network and PID Coupling Control Strategy.” Journal of Iron and Steel Research, International, No. 10, Vol. 20, pp. 17-22, 2013.

[21] H. Zhao, M. Sugisaka, ”Simulation study of CMAC control for the robot joint actuated by McKibben muscles,” Applied Mathematics and Computation, No. 1, Vol. 203, pp. 457-462, 2008.

[22] L. Wang, X. Fang, S. Duan and X. Liao, ”PID Controller Based on Memristive CMAC Network,”Abstract and Applied Analysis,vol. 2013, Article ID 510238, 6 pages, 2013.

[23] R. Kurozumi, T. Yamamoto, S. Fujisawa and O. Sueda “Development of Training Equipment with Adaptive and Learning Using a Balloon Actuator-Sensor System” Journal of Robotics and Mechatronics, No. 1, Vol. 21, pp. 156-163, 2009.

[24] R. Kurozumi, T. Yamamoto, “A Design of CMAC Based Intelligent PID controllers”, IEEJ trans. EIS, No.4, Vol. 125, pp. 607-615, 2005. (In Japanese)

[25] T. Yamamoto, R. Kurozumi, S. Fujisawa “A Design of CMAC Based Intelligent PID controllers”, ICANN/ICONIP 2003, Berlin Heidelberg 2003.

[26] Y. Liao, T. Yamamoto , “ Design and Experimental Evaluation of a Human Skill-Based PID Controller ” Journal of Robotics, Networking and Artificial, Vol.2 No.3 pp.140-143, 2015.

[27] K. Koiwai, Y. Liao, T. Yamamoto, T. Nanjo, Y. Yamazaki, Y. Fujimoto, “Feature Extraction for Excavator Operation skill using CMAC”, Journal of Robotics and Mechatronics, Vol.28 No.5 pp. 715-721, 2016.

[28] P. Tan, M. Steinbach and V. Kumar ”Introduction to Data Mining,” Pearson Addison-Wesley, 2006.

[29] S. Cong, Y. Liang, “PID-Like Neural Network Nonlinear Adaptive Control for Uncertain Multivariable Motion Control Systems,”IEEE Transactions on Industrial Electronics, No.10, Vol. 56, pp.3872-3879, 2009.

[30] T.Yamamoto and S.L.Shah, ”Design and experimental evaluation of a multivariable self-tunning PID controller,” IEE Proc. of Control Theory and Applications Vol.151, No.5, 645-652, 2004.

[31] K.L. Chien, J.A. Hrones, and J.B. Reswick, “On the automatic control of generalized passive systems”, Trans. ASME, Vol.74, pp.175-185, 1972.

[32] M. Farahani, S. Ganjefar and Mo. Alizadeh “A Self-tunning PID Design Based on Wavelet Neural Network Using Lyapunov Method,” 2nd International Conference on Control, Instrumentation and Automation (ICCIA) 2011.

[33] J. Kang, W. Meng, A. Abraham and H. Liu “An adaptive PID neural network for complex nonlinear system control,” Neurocomputing, Vol. 135, pp. 79-85, 2014.

[34] Y. Suehiro, Y. Obika and T. Yamamoto, “FRIT-based tuning of PID parameters using a genetic algorithm,” Proceedings of International Conference on Networking, Sensing and Control, 2009, Japan, Okayama.

[35] M. Demirtas, “Off-line tuning of a PI speed controller for a permanent magnet brushless DC motor using DSP,” Energy Conversion and Management, Vol. 52, pp. 264-273, 2011.

[36] M. Villarreal-Cervantes, J. Alvarez-Gallegos, “Off-line PID control tuning for a planar parallel robot using DE variants,” Expert Systems With Applications, Vol. 64, pp. 444-454, 2016.

[37] O.Kaneko, S.Souma, and T.Fujii, ”Fictitious reference iterative tuning in the two-degree of freedom control scheme and its application to a facile closed loop system identification,” Trans. of SICE, Vol. 42, No. 1,pp. 17-25, 2006.

[38] T.Shigemasa, Y.Negishi, and Y.Baba, ”From frit of a pd feedback loop to process modeling and control system design,” Proc. of 11th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing, 2013.

[39] S.Wakitani, Y.Ohnishi, and T.Yamamoto, ”Design of a cmac-based pid controller using operating data,” Distributed Computing and Artificial Intelligence Advances in Intelligent and Soft Computing, Vol. 151, pp. 545-552, 2012.

[40] S.Wakitani, T.Nawachi, G.R.Martins, and T.Yamamoto, "Design and implementation of a data-oriented nonlinear pid controller," Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 17, No. 5, pp. 690-698, 2013.

[41] T. Yamamoto, T. Kinoshita, Y. Ohnishi, and S. L. Shah "Design and Experimental Evaluation of a Performance-Driven PID Controller" Journal of Robotics and Mechatronics Vol.28 No.5 pp. 616-624, 2016.

[42] T. J. Harris: Assessment of closed loop performance; Canadian Journal of Chemical Engineering, Vol.67, pp.856-861, 1989.

[43] M. Jelali, "Control Performance Management in Industrial Automation," Springer-Verlag,London, 2013.

[44] B. Huang and S. L. Shah "Performance assessment of control loops: theory and applications," Springer, London, 1999.

[45] T. Yamamoto, Y. Ohnishi, and S. L. Shah, "Design of a Performance-Adaptive Proportional-Integral-Derivative Controller for Stochastic Systems," Proceedings of the Institute of Mechanical Engineering, Part-I: Journal of Systems and Control Engineering, Vol.222, pp. 691-699, 2008.

[46] Y. Ohnishi, K. Takao, T. Yamamoto, and S. L. Shah,"Design of a PID Controller with a Performance-Driven Adaptive Mechanism," Proceedings of American Control Conf., New York, pp.1359-1364, 2007.

[47] Y. Liao,T. Kinoshita, K. Koiwai and T. Yamamoto, "Design of a Performance-Driven CMAC PID Controller," IEICE Transactions on Funda-

mentals of Electrics Communication and Computer Sciences Vol.E100-A, No.12. pp. 2963-2971, 2017.

[48] Y. Liao, K. Takuya, K. Koiwai and T. Yamamoto. “Design of a performance-driven PID controller for a nonlinear system.” 2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP 2017), Taipei, Taiwan, 2017, May.

[49] R. Gao, A. O ’ Dwyer, E. Coyle ”Model Predictive Control of CSTR Based on Local Model Networks” Proceedings of the Irish Signals and Systems Conference, University College Cork, pp. 397-402, 2002.

[50] L. Desborough, T. J. Harris: Performance assessment measures for univariate feedback control; The Canadian Journal of Chemical Engineering, Vol.70, pp.1186-1197, 1992.

[51] Y. Liao, K. Koiwai and T. Yamamoto. “Design and Implementation of a Hierarchical-Clustering CMAC PID Controller,” Asian Journal of Control. (in press)

Publication Lists

Journal Publications Related to This Thesis

[1] Y. Liao, T. Yamamoto, “Design and Experimental Evaluation of a Human Skill-Based PID Controller,” Journal of Robotics, Networking and Artificial, Vol.2 No.3 pp.140-143, 2015.

[2] Y. Liao, T. Kinoshita, K. Koiwai, T. Yamamoto, “Design of a Performance-Driven CMAC PID Controller,” IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences, Vol.E100-A,No.12. pp. 2963-2971, 2017.

[3] Y. Liao, K. Koiwai and T. Yamamoto, “Design and Implementation of a Hierarchical-Clustering CMAC PID Controller,” Asian Journal of Control. (in press)

International Conference Publications Related to This Thesis (Peer-Reviewed)

[1] Y. Liao, K. Takuya, T. Yamamoto, ”Design and experimental evaluation of a human skill-based controller,” 2014 International Conference on Advanced Mechatronic Systems (ICAMechS 2014),Kumamoto, Japan, August, 2014.

[2] Y. Liao, K. Koiwai, T. Yamamoto. ”Design of a hierarchical-clustering CMAC-PID controller.” 2017 International Joint Conference on Neural Networks (IJCNN 2017), Alaska, American, May, 2017.

[3] Y. Liao, K. Takuya, K. Koiwai, T. Yamamoto. ”Design of a performance-driven PID controller for a nonlinear system.” 2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP 2017), Taipei, Taiwan, May, 2017.

[4] Y. Liao, K. Takuya, K. Koiwai, T. Yamamoto. ”Design of a performance-driven PID controller Using a Hierarchical-Clustering CMAC.” 2017 The 49th ISCIE International Symposium on Stochastic Systems Theory and Its Applications (SSS’ 17), Hiroshima, Japan, November, 2017.

Journal Publications not Related to This Thesis

[1] K. Koiwai, Y. Liao, T. Yamamoto, T. Nanjo, Y. Yamazaki, Y. Fujimoto, “Feature Extraction for Excavator Operation skill using CMAC,” Journal of Robotics and Mechatronics, Vol.28 No.5 pp. 715-721, 2016.

International Conference Publications not Related to This Thesis (Peer-Reviewed)

[1] K. Koiwai, Y. Liao, T. Yamamoto, T. Nanjo, Y. Yamazaki, Y. Fujimoto， “ A Consideration on Feature Extraction for Operation Skill Based on Control Engineering Approach ”， The 2016 International Conference on Artificial Life and Robotics (ICAROB 2016), Okinawa, Japan, January, 2016.

[2] K. Koiwai, Y. Liao, T. Yamamoto, T. Nanjo, Y. Yamazaki, Y. Fujimoto, "Human Skill Evaluation Based on Control Engineering Approach," The 13th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems (IFAC-HMS 2016), Kyoto, Japan, September, 2016.