'94 KACC (1994. 10. 17 ~ 20)

A CMAC Network Based Controller

Keun-Mo Koo and Jong-Hwan Kim

Department of Electrical Engineering,
Korea Advanced Institute of Science and Technology (KAIST),
373-1 Kusung-dong, Yusung-gu, Taejon-shi 305-701,
Republic of Korea

Abstract – This paper presents a CMAC network based controller on the basis of Lyapunov theory. CMAC network is employed to approximate and to compensate the uncertainties induced by inaccurate modelling of the system. For the improvement of robustness under the bounded disturbances and the approximation error of the CMAC, the adaptation scheme with a deadzone and an additional control input are developed.

I Introduction

Recently, the design of neural network based controller has been extensively studied by numerous researchers. Neural networks make it possible to control an unknown nonlinear plant accurately without a stringent prior information about the plant dynamics. However, the major problem introduced by such a network based controller is that the exact error gradient information can not be obtained since we do not know which plant input generates the desired output in general. Moreover, the learning convergence is not fast enough to be applied to on-line direct control scheme.

The CMAC learning controller proposed by Albus has conceptually simpler structure and faster learning convergence than other artificial neural networks. However, the input region of the CMAC controller should be bounded and quantized *a priori*. This property requires an additional control scheme which stabilizes the system outside the bounded input region.

In this paper, we propose a CMAC network based controller using Lyapunov theory. Since CMAC is essentially a controller in a table look-up fashion which is adaptive, the stability can be effectively analyzed using Lyapunov stability theory. For the improvement of robustness under the approximation error and bounded disturbances, the adaptive control scheme with a deadzone and an additional control input are employed.

II Description of CMAC

CMAC is a table look-up algorithm which is adaptive by modifying the contents of the table using a learning algorithm. It has generalization capability (similar inputs produce similar outputs) due to the distributed storage of information. As shown in Figure 1, CMAC is composed of a mapping from an input space to memory locations and a learning algorithm. Given a state in input space, the operations of CMAC are to find the corresponding memory locations for that state, to sum the content of these memory locations to get the response of the CMAC, to compare it with desired response, and to modify the content of these memory locations based on the learning algorithm. A conventional mapping would be to assign one memory location for each input state. In this case the required memory would be extremely large. For example, a system with N inputs, each of which can take on

R different values, would have R^N memory locations. To reduce such massive memory locations, Albus [1] derived a mapping algorithm inspired by the human cerebellum. In this mapping an element in the input space is mapped into many memory locations. The number of memory locations assigned to each input element is called the generalization size, N_g. Two input elements that are closest and different each other will have their $N_g - 1$ memory locations in common. So the number of total memory locations needed is reduced to $N_g(\frac{R}{N_g})^N$.

The learning of the CMAC is typically based on the desired training data pairs y and $u_d(y)$, where $u_d(y)$ is a desired network output in response to the input y. The typical update rule has the following form :

$$\Delta w_i = \eta(u_d(y) - u_c(y)) \tag{1}$$

where $u_c(y)$ is the output of the CMAC. In control problems, it is difficult to obtain the desired training data pairs y and $u_d(y)$ *a priori*, i.e., we do not know which input generates the desired output in general. The next section describes a mathematical representation of the CMAC and how to design a controller using CMAC.

III Controller Design using CMAC

Let the mapping $H(y)$ represent a mapping of an input element into N_g generalized memory locations, i.e., the region of the mapping $H(y)$ is a set containing N_g generalized memory locations corresponding to an input element y. Then the CMAC output $u_c(y)$ can be expressed as

$$u_c(y) = \sum_{j=1}^{m} w_j \chi(a_j) \tag{2}$$

where m is the total number of actual memory, w_j are weight terms, a_j represents the jth memory location, and the characteristic function $\chi(a_j)$ is defined as

$$\chi(a_j) = \begin{cases} 1 & \text{if } a_j \in H(y) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The above expression of the CMAC output is similar to the linear regressor form which appears frequently in adaptive control theory. So, the design technique developed in the area of adaptive control can also be applied in a CMAC based controller design. We illustrate how this approach can be applied to the CMAC based controller using a simple example.

Let us consider a simple mechanical system

$$I\ddot{x} + h(\dot{x}) = \tau \tag{4}$$

where τ is the input torque, h is the nonlinear friction term, x is the output displacement, and $I > 0$ is the total inertia of the system. It is assumed that only the position and velocity of the system can be measured. So, the uncertainty to be compensated by the neural net should be expressed in terms of the position and velocity. For this purpose, we propose the control law for (4) as

$$\tau = \hat{I}\ddot{x}_r + \hat{h}(\dot{x}) + \tau_n \tag{5}$$

$$\ddot{x}_r \triangleq \ddot{x}_d + K_v\dot{\tilde{x}} + K_p\tilde{x} \tag{6}$$

$$\tilde{x} \triangleq x_d - x \tag{7}$$

where $\hat{I}$ and $\hat{h}$ are the estimates of I and h, respectively, x_d is a desired trajectory, and τ_n is the output of the neural network.

Using (4) and (5), we obtain the following equation :

$$\ddot{\tilde{x}} + K_v\dot{\tilde{x}} + K_p\tilde{x} = I^{-1}[\tilde{I}\ddot{x}_r + \tilde{h}(\dot{x}) - \tau_n] \tag{8}$$

where $\tilde{I} \triangleq I - \hat{I}$ and $\tilde{h} \triangleq h - \hat{h}$.

Let us define e_1 as follows :

$$e_1 = \dot{\tilde{x}} + \lambda\tilde{x} \tag{9}$$

where $\lambda > 0$ is chosen so that transfer function $\frac{s+\lambda}{s^2+K_v s+K_p}$ is strictly positive real. Then there exist the positive definite matrices P and Q such that

$$A^T P + PA = -Q \tag{10}$$

$$PB = C^T \tag{11}$$

where the matrices A, B, and C are the matrices of the following minimal state-space realization of the error equations (8) and (9) :

$$\dot{e} = Ae + B[I^{-1}(g(\ddot{x}_r, \dot{x}) - \tau_n)] \tag{12}$$

$$e_1 = Ce \tag{13}$$

where $e \triangleq [\tilde{x} \ \ \dot{\tilde{x}}]^T$ and $g(\ddot{x}_r, \dot{x}) \triangleq \tilde{I}\ddot{x}_r + \tilde{h}(\dot{x})$. The role of the CMAC is to compensate the uncertainty $g(\ddot{x}_r, \dot{x})$ so that the error dynamics (12) may become asymptotically stable. We assume that the uncertain function $g(\ddot{x}_r, \dot{x})$ can be represented as

$$g(\ddot{x}_r, \dot{x}) = \sum_{j=1}^{m} w_j^* \chi(a_j) \quad (14)$$

where w_j^* are unknown constant weights. The CMAC output may be expressed using the variable weight w_j as

$$u_c = \sum_{j=1}^{m} w_j \chi(a_j). \quad (15)$$

Then the difference between the actual uncertainty and the CMAC output can be represented as

$$g(\ddot{x}_r, \dot{x}) - u_c = \sum_{j=1}^{m} \tilde{w}_j \chi(a_j) \quad (16)$$

where $\tilde{w}_j \triangleq w_j^* - w_j$.

The adaptation law for the w_j can be obtained using the following Lyapunov function :

$$V = e^T P e + \frac{1}{I\eta} \sum_{j=1}^{m} \tilde{w}_k^2 \quad (17)$$

where $\eta > 0$ is the adaptation rate. Differentiating (17) with respect to time and from (12), (13) and (16), we get

$$\dot{V} = -e^T Q e + \frac{2}{I} e_1 \sum_{j=1}^{m} \tilde{w}_j \chi(a_j) - \frac{2}{I\eta} \sum_{j=1}^{m} \tilde{w}_j \dot{w}_j. \quad (18)$$

Let the updating rule for w_j be

$$\dot{w}_j = \eta \chi(a_j) e_1. \quad (19)$$

Then (21) becomes

$$\dot{V} = -e^T Q e \leq 0 \quad (20)$$

which represents the asymptotic stability of the system.

IV Robust Controller

Since the uncertainty can not be exactly represented by the neural net and the output of the CMAC may contain disturbances, we should consider an approximation error of the CMAC and disturbances to improve robustness of the controller.

Let us include these effects in (14) as

$$g(\ddot{x}_r, \dot{x}) = \sum_{j=1}^{m} w_j^* \chi(a_j) + \epsilon(t) \quad (21)$$

where $\epsilon(t)$ represents the effects of the approximation error and the disturbances. It is assumed that the magnitude of $\epsilon(t)$ is bounded by an unknown positive constant α such that

$$|\epsilon(t)| \leq \alpha. \quad (22)$$

Since the constant α is assumed to be unknown, it should be estimated using a suitable estimation law. Let us define the estimate of α as $\hat{\alpha}$. Using the estimate of α an additional control input should be added to suppress the effect of $\epsilon(t)$. For this purpose we propose the following theorem.

Theorem 1 *Let us consider the following estimation and control laws :*

when $|e_1| > \delta$,

$$\begin{cases} \dot{w}_j = \eta \chi(a_j) e_1, \\ \dot{\hat{\alpha}} = \beta |e_1|, \\ \tau = \hat{I}\ddot{x}_r + \hat{h}(\dot{x}) + \hat{\alpha}\frac{e_1}{|e_1|} + \tau_n \end{cases} \quad (23)$$

when $|e_1| \leq \delta$,

$$\begin{cases} \dot{w}_j = 0, \\ \dot{\hat{\alpha}} = 0, \\ \tau = \hat{I}\ddot{x}_r + \hat{h}(\dot{x}) + \hat{\alpha}\frac{e_1}{\delta} + \tau_n \end{cases} \quad (24)$$

where β and δ are arbitrary positive constants. Then the control input is continuous and the closed-loop system is uniformly ultimately bounded.

Proof: Let Ω_1 and Ω_2 be defined as $\Omega_1 \triangleq \{t \mid |e_1(t)| > \delta\}$ and $\Omega_2 \triangleq \{t \mid |e_1(t)| \leq \delta\}$, respectively so that Ω_1 and Ω_2 are partitionings of R^+. Let us define a Lyapunov function candidate for the system (12) as

$$V = e^T P e + \frac{1}{I\eta} \sum_{j=1}^{m} \tilde{w}_j^2 + \frac{1}{I\beta} \tilde{\alpha}^2 \quad (25)$$

where $\tilde{\alpha} \triangleq \alpha - \hat{\alpha}$. When $t \in \Omega_1$,

$$\begin{aligned} \dot{V} \ = \ & -e^T Q e + \frac{2}{I} e_1 (\sum_{j=1}^{m} \tilde{w}_j \chi(a_j) + \epsilon(t) - \hat{\alpha}\frac{e_1}{|e_1|}) \\ & + \frac{2}{I\eta} \sum_{j=1}^{m} \tilde{w}_j \dot{\tilde{w}}_j - \frac{2}{I\beta} \tilde{\alpha}\dot{\tilde{\alpha}}. \end{aligned} \quad (26)$$

Using (23), we get

$$\dot{V} \le -e^T Q e < 0. \tag{27}$$

When $t \in \Omega_2$,

$$\begin{aligned} \dot{V} &= -e^T Q e + \frac{2}{I} e_1 (\sum_{j=1}^{m} \tilde{w}_j \chi(a_j) + \epsilon(t) - \hat{\alpha}\frac{e_1}{\delta}) \\ &\le -e^T Q e + b(t) \end{aligned} \tag{28}$$

where $b(t) \triangleq \frac{2}{I}\delta(\sum_{j=1}^{m} |\tilde{w}_j| \chi(a_j) + \tilde{\alpha})$. Since $\dot{V} < 0$ for $t \in \Omega_1$, $V(\cdot)$ is continuously decreasing and $|\tilde{w}_j|$ and $|\tilde{\alpha}|$ cannot be increasing for $t \in \Omega_1$. Further, for $t \in \Omega_2$, $|\tilde{w}_j|$ and $|\tilde{\alpha}|$ remain constant. Thus $b(t)$ is a bounded function since $\chi(a_j)$ is bounded. So uniformly ultimate boundedness of signals can be obtained. ∎

Remark : In the above discussion we assume that the input signals of the CMAC are always confined in the bounded input space of the CMAC. However, there is no guarantee that the controller acts as we assumed. In this case we may use the hybrid control scheme shown in [4].

V Conclusion

In this paper we have proposed a CMAC based controller. A mathematical model of a CMAC are described and using this model a robust CMAC controller is proposed on the basis of Lyapunov theory. Using the proposed design method we can design the CMAC based controller more analytically. The proposed controller employs the dead-zone technique in robust adaptive control and a sliding control method to improve the robustness.

References

[1] J. S. Albus, "A new approach to manipulator control : the cerebellar model articulation controller (CMAC)," *Trans. ASME, J. Dyn. Sys., Meas. Contr.*, vol. 97, pp.220-227, Sept. 1975.

[2] L. G. Kraft and D. P. Campagna, "A comparison between CMAC neural network control and two traditional adaptive control systems," *IEEE Contr. Syst. Mag.*, pp.36-43, April 1990.

[3] W. T. Miller, R. P. Hewes, F. H. Glanz, and L. G. Kraft, "Real-time dynamic control of an industrial manipulator using a neural-network based learning controller," *IEEE Trans. Robotics Automat.*, vol.6, no.1, pp.1-9, Feb. 1990.

[4] R. M. Sanner and J.-J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Networks*, vol.3, no.6, pp.837-863, Nov. 1992.

[5] B. B. Peterson and K. S. Narendra, "Bounded error adaptive control," *IEEE Trans. Automat. Contr.*, vol.AC-27, pp. 1161-1168, Dec. 1982.

[6] K.-M. Koo and J.-H. Kim, "Robust controller for robot manipulators with parametric uncertainty," *IEEE Trans. Automat. Contr.*, vol.39, no.6, pp.1230-1233, June 1994.

[7] K.-M. Koo and J.-H. Kim, "On the neural network based controller," *Submitted to the ICONIP '94-Seoul.*

[8] S.-W. Lee and J.-H. Kim, "Control of systems with deadzones using neural-network based learning controller," *Proc. of IEEE World Congree on Computational Intelligence*, Orlando, Florida, June 1994.

[9] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, New Jersey : Prentice-Hall, 1991.

[10] J. J. Craig, *Adaptive Control of Mechanical Manipulators*, Addison-Wesley, 1988.

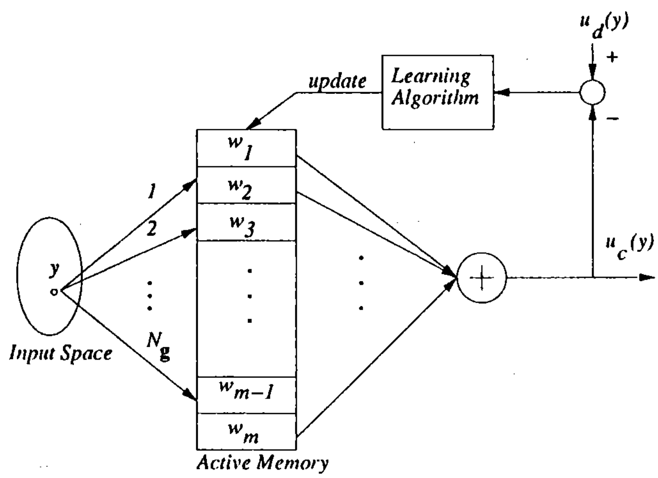

Fig. 1. CMAC structure