

다중 샘플링 타임을 갖는 CMAC 학습 제어기 실현: 역진자 제어

CMAC Learning Controller Implementation With Multiple Sampling Rate: An Inverted Pendulum Example

이 병 수*
(Byoungsoo Lee)

Abstract : The objective of the research is two fold. The first is to design and propose a stable and robust learning control algorithm. The controller is CMAC Learning Controller which consists of a model-based controller, such as LQR or PID, as a reference control and a CMAC. The second objective is to implement a reference control and CMAC at two different sampling rates. Generally, a conventional controller is designed based on a mathematical plant model. However, increasing complexity of the plant and accuracy requirement on mathematical models nearly prohibits the application of the conventional controller design approach. To avoid inherent complexity and unavoidable uncertainty in modeling, biology mimetic methods have been developed. One of such attempts is Cerebellar Model Articulation Computer(CMAC) developed by Albus. CMAC has two main disadvantages. The first disadvantage of CMAC is increasing memory requirement with increasing number of input variables and with increasing accuracy demand. The memory needs can be solved with cheap memories due to recent development of new memory technology. The second disadvantage is a demand for processing powers which could be an obstacle especially when CMAC should be implemented in real-time. To overcome the disadvantages of CMAC, we propose CMAC learning controller with multiple sampling rates. With this approach a conventional controller which is a reference to CMAC at high enough sampling rate but CMAC runs at the processor's unoccupied time. To show efficiency of the proposed method, an inverted pendulum controller is designed and implemented. We also demonstrate its possibility as an industrial control solution and robustness against a modeling uncertainty.

Keywords : Cerebellar Model Articulation Computer(CMAC), CMAC learning controller, multiple sampling rate, inverted pendulum

I. 서론

다양한 응용 분야에서의 기술 발달로 인해 비선형성과 많은 자유도를 가지는 복잡한 시스템을 제어하고자 하는 요구가 증가하고 있다. 이와 같은 시스템은 너무 복잡해서 모델을 이끌어 내는 것이 실용적으로 불가능하거나, 가능하더라도 모델링 비용이 적지 않은 경우가 허다하다. 더구나 닫혀진 형태의 모델을 얻는다 하더라도 계통의 파라미터가 정확하지 않아 제어계통의 모델의 불확실성에 대한 강건성을 확보하는 것이 큰 과제가 된다. 즉, 파라미터의 불확실성, 모델에서 제외된 비선형성, 측정 잡음 그리고 예측하지 못한 외란에 강건한 제어기의 설계가 요청된다.

이와 같은 강건성 문제 때문에 많은 계산량과 변수를 다룰 수 있는 새로운 방법 즉, 생물학적인 신경 구조 모델을 이용하여 시스템을 해석하려는 연구가 진행되어 왔고, 여러 가지의 신경망이 모델화 되었는데 잘 알려져 있는 것으로 Rosenblatt의 Perceptron과 Widrow의 Adaline, Hopfield의 Recurrent Network 등이 있다. 그 중에서도 1970년대 J. S. Albus[1]에 의해 제안된 CMAC(Cerebellar Model Articulation Computer)은 입력과 응답으로 정해지는 계통의 비선형 함수 학습에 적용할 수 있다. 또한 CMAC 학습 알고리즘은 작은 횟수의 반복적인 훈련으로도 수렴하는 장점이 있다.

Thomas Miller(1987)는 Albus가 제안한 CMAC 제어기에 기초하여 실제 산업 로봇 팔의 제어를 위한 새로운 학습 기술인 CMAC learning controller를 제안하였다[2,3]. Miller가 제안한 방법은 전통적인 제어기를 기준 제어기 입력으로 사용하고, 이 입력과 계통의 응답을 CMAC의 훈련 데이터로 사용하는 것이다.

CMAC은 비선형 함수를 학습하고, 수렴 특성이 좋은 장점이 있지만, 모델링의 정확도를 높일 목적으로 분해능을 높이거나, 혹은 변수가 많을 경우 대단히 많은 메모리를 필요로 한다는 단점이 있다. 그 외에 알고리즘을 수행하는데 많은 프로세싱 파워가 필요한데 실시간 제어를 목적으로 하는 경우 고성능의 프로세서를 필요로 하거나 혹은 병렬 프로세싱 기법을 사용해야 하는 등 복잡성은 실시간 제어에 걸림돌이 된다. 최근 반도체 기술의 발달로 대용량 메모리를 값싼 가격으로 얻을 수 있으므로 메모리에 요구에 대한 해결책은 어렵지 않다. 프로세싱 파워에 대한 해결책으로는 한 가지 방법을 제안할 수 있다. 기존의 제어기는 작은 샘플링 시간으로 고속 실행시키고, CMAC 제어기는 나머지 여분의 프로세싱 파워를 이용하는 방법을 제안할 수 있다.

본 연구에서는 불안정하고 비선형적인 계를 제어함에 있어서 Thomas Miller가 제안한 CMAC learning controller를 응용한다. 제안된 제어기는 기준제어기와 CMAC제어기로 구성되는데 기준 제어기로는 상태궤환제어를 채용하며, 이

* 책임저자(Corresponding Author)

논문접수 : 2007. 1. 25., 채택확정 : 2007. 2. 17.

이병수 : 계명대학교 기계자동차공학부(blee@knu.ac.kr)

두 제어기를 서로 다른 샘플링 시간으로 실행시키는 기법을 시험하였다. 대상 계통은 독립진자이며 상태회환제어기만으로는 외란 혹은 모델링 불확실성에 대해 민감하지만 CMAC learning controller를 결합할 경우 외란에 강건해짐을 실험적으로 보인다.

II. The Cerebellar Model Articulation Computer

J. S. Albus[1]는 소뇌의 생물학적 특성을 응용하여 CMAC을 제안하면서, 로봇 팔의 제어에 이용하였다. CMAC은 입력에 대한 정확한 출력을 얻기 위해 일반적인 제어기처럼 시스템의 특성을 분석하여 수학적인 해를 얻는 것이 아니라, 시스템의 함수를 미리 메모리 테이블이라 불리는 곳에 기억시켜 놓고, 어떤 입력이 주어지면 그 입력에 해당하는 메모리 테이블의 위치를 찾아 그 위치에 해당하는 값들의 합을 출력값으로 내어놓는 방법을 이용한다. 즉,

$$p = f(S) \tag{1}$$

여기서 S 는 불연속 다차원 입력 상태벡터이고, p 는 출력 벡터이며 f 는 S 에서 p 로의 매핑이다.

위와 같은 관계를 얻기 위해서는 먼저 제어함수를 기억 테이블에 저장하는 훈련(training)을 해야 하며, 이 훈련과정은 일련의 변환과정(mapping)을 거쳐 이루어진다. 이 변환 과정은 상태에 따른 함수 값을 어디에 저장하고 어디서 불러내는지 결정한다. 이런 변환 과정은 훈련 중에도, 또한 제어기로써 작동할 때에도 거친다. CMAC의 일련의 변환 과정은 다음과 같다[4,5].

$$S \rightarrow M \rightarrow A \rightarrow p \tag{2}$$

여기서

- S = 입력 변수 벡터
- M = 이산화를 위한 중간변수 집합
- A = M 에 의해 구성되는 기억공간 집합
- p = 출력 값.

이며 다음 절에서는 (2)의 화살표로 표현되는 각각의 변환 과정에 대해 설명한다.

1. $S \rightarrow M$ 변환

N 개의 입력변수로 구성된 입력벡터는 N 개의 입력변수 공간의 입력변수 축 $S = (S_1, S_2, \dots, S_N)$ 로 정의 할 수

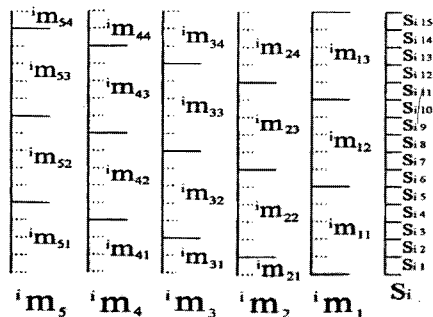


그림 1. An example of $S \rightarrow M$ mapping.
Fig. 1. $S \rightarrow M$ 변환 예.

있다. 이들 각각의 입력변수 축은 일정한 해상도 r_i 로 불연속화하여 입력변수 축의 크기를 해상도로 나눈 수 만큼의 마디로 분할된다. 각 입력변수 축은 K 개의 중간변수 축으로 변환되는데 이들 중간변수 축은 또한 r_i 의 K 배 만큼의 해상도를 가지는 불연속화 된 축이다. 한 입력변수 축에 대한 K 개의 중간변수 축은 각각 입력변수 축의 해상도만큼 전위(offset)되어 있다. 이들 중간변수 축의 집합을 M 이라는 기호를 써서 나타내고 이를 수식으로 표현하면 다음과 같다.

$$M = \begin{pmatrix} {}^1m_1, & {}^1m_2, & \dots, & {}^1m_K \\ {}^2m_1, & {}^2m_2, & \dots, & {}^2m_K \\ \vdots & \vdots & & \vdots \\ {}^Nm_1, & {}^Nm_2, & \dots, & {}^Nm_K \end{pmatrix} \tag{3}$$

$$S \rightarrow M: S_i \rightarrow ({}^im_1, {}^im_2, \dots, {}^im_K)$$

여기서, S_i 는 i 번째 입력 변수축이고, im_K 는 S_i 에 대한 K 번째 중간변수 축이다.

그림 1에서 입력변수 축 S_i 는 해상도 r_i 로 나뉘어 15개의 마디로 분할되어있다. 이 예에서 $K=5$ 이므로 다섯 개의 중간변수 축 ${}^im_1, {}^im_2, {}^im_3, {}^im_4, {}^im_5$ 가 있고 각 중간변수 축에서 분할된 마디의 개수는 ${}^iN_1=3, {}^iN_2=4, {}^iN_3=4, {}^iN_4=4, {}^iN_5=4$ 이다. 만약 입력변수 값이 $s_{i,8}$ 이라면 이는 $({}^im_{12}, {}^im_{23}, {}^im_{33}, {}^im_{43}, {}^im_{52})$ 로 변환된다. 입력변수가 각 중간변수의 어느 마디에 해당하는지 (3)으로 계산할 수 있다[4,5].

$$l_1 = 1 + (n-1)/K \tag{4}$$

$$l_j = (2K + n - j)/K, j \geq 2$$

여기서 n 은 $S_{i,n}$ 의 n 이고 두 번째 식은 나머지가 무시되는 정수 계산이다.

2. $M \rightarrow A$ 변환

그림 2는 $M \rightarrow A$ 변환의 예를 나타낸 그림이다. 각 입력변수 축에 대해 변환층이 같은 중간변수 축은 입력공간과

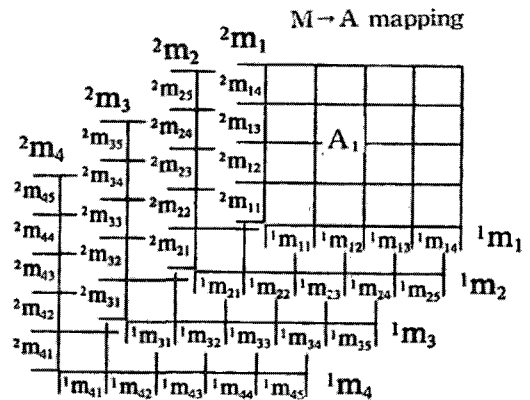


그림 2. $M \rightarrow A$ 변환 ($K=4$).
Fig. 2. $M \rightarrow A$ mapping ($K=4$).

같은 차원의 기억공간을 구축하는데 이 기억공간이 A공간이다. 따라서 A공간은 K개의 기억공간을 가진다. 그림 2를 예로 들어보면 첫 번째 기억공간 A₁은 첫 번째 입력변수 축 S₁의 첫 번째 중간변수 축 ¹m₁과 두 번째 입력변수 축 S₂의 첫 번째 중간변수 축 ²m₁으로 이루어진 기억공간이다. 각각의 중간 변수 축은 4개의 구간으로 분할되어 있고 따라서 기억공간 A₁은 16개의 기억장소로 구분된다. 즉,

$$M \rightarrow A: ({}^1m_j, {}^2m_j, \dots, {}^Nm_j) \rightarrow A_j ({}^1N_j, {}^2N_j, \dots, {}^NN_j) \quad (5)$$

여기서 ^Nm_j는 N번째 입력변수 축에 대한 j번째 중간변수 축이고 ^NN_j는 ^Nm_j를 분할한 마디의 개수이다.

3. A→p 변환

K개의 기억공간 각각에서 입력벡터에 해당하는 기억장소의 기억값이 복사되고, 그 값들이 다시 더해져 제어기의 출력값이 된다.

$$A \rightarrow p: \sum_{j=1}^K w_j = p \quad (6)$$

여기서 w_K는 K번째 기억공간인 A_K의 어느 번지에 저장되었던 기억값이고 가중치라 불린다. 이 가중치는 $\hat{p} = [\hat{p}_1, \hat{p}_2, \dots, \hat{p}_k]$ 를 기대하는 값이라 하고 $p = [p_1, p_2, \dots, p_k]$ 는 실제 측정된 값이라 한다면, p와 \hat{p} 의 모든 요소에 대해 $|\hat{p}_k - p_k| \geq \zeta_k$ 라는 조건하에서 p_k를 구성하는 모든 가중치에 Δ_k를 더하여서 구해진다. 즉,

$$\Delta_k = g(\hat{p}_k - p_k) / K \quad (7)$$

여기서 ζ_k는 허용오차이고, K는 중간변수 축의 수이다. g는 한번의 학습경험으로 생성되는 오차 정정량을 조절하는 이득 인자이다. g=1이면 CMAC는 하나의 값을 저장하는 동작에서 측정오차를 완벽하게 검출하는 학습을 한다. 0 < g < 1이면 각 학습경험은 출력 p_k를 원하는 값 \hat{p}_k 의 방향으로 이동시키며, 목표 성능을 얻기 위해서는 반복 학습이 필요하다.

III. CMAC learning controller 설계

본 연구에서는 도립진자를 제어하기 위해 W. T. Miller III가 산업용 로봇팔을 제어하기 위해 제안한, CMAC learning controller[2]를 응용하여 그림 3과같은 학습 방식을 구성하였다. CMAC learning controller는 모델을 기반으로 하는 제어기(본 연구에서는 LQR제어기)와 비선형성에 대응하고 외란에 대한 강인성을 향상시킬 목적으로 CMAC을 병행한다.

이 두 제어기를 병렬로 실행하는데 있어서 처리속도의 문제가 발생하는데 실제 시스템에서 LQR제어기는 매우 빠른 속도로 계산이 가능하지만, CMAC은 많은 메모리를 사용하고 데이터 처리과정에서 훈련 과정을 거치기 때문에

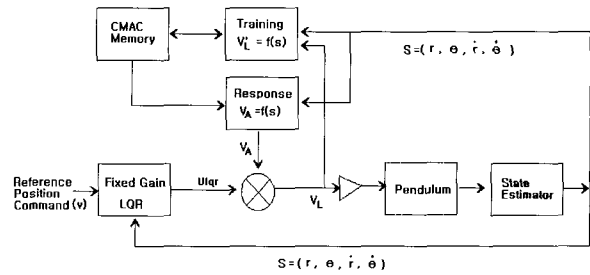


그림 3. CMAC 훈련 제어기의 블록선도.
Fig. 3. CMAC learning controller의 block diagram.

계산에 걸리는 시간이 LQR제어기보다 길다. 이는 제어기의 프로세싱 파워가 부족할 때 문제가 되는데 이런 문제를 해결하기 위해 두 제어기의 샘플링 시간을 달리하는 방법을 고안하였다. LQR제어기는 인터럽터를 이용하여 충분히 빠른 샘플링 시간으로 제어를 하고, CMAC은 제어기의 능력이 허락하는 여유 시간에 처리한다.

우선 본 연구에서 제안하는 제어 방법을 그림 3의 다이어그램을 통해 설명한다.

1. 제어값 V_L과 도립진자의 상태변수 S=(r, θ, \dot{r} , $\dot{\theta}$)가 LQR제어기에 입력되고 LQR제어기는 제어 값인 U_{lqr}를 출력한다.
2. 상태변수 S는 또한 CMAC 제어기에 입력되고, 그 상태에 해당하는 메모리 위치에 있는 가중치가 모두 합쳐져 V_A가 출력된다. 초기 상태에서 메모리 테이블은 빈 공간의 상태로 가중치가 모두 0이기 때문에 초기 상태의 S에 대한 CMAC의 출력 값은 0이 된다.
3. U_{lqr}과 V_A가 합해져서 도립진자 제어값인 V_L을 생성한다.
4. V_L은 CMAC의 훈련 과정으로 입력되고 V_L'과의 비교에 의해 한 번의 훈련을 한다.
5. 순서 3에서 구한 V_L은 또한 도립진자 시스템으로 입력되고, 계를 제어한 후 다음 상태변수 r과 θ를 출력한다. 이 상태변수는 상태산출기로 입력되어 추정된 상태변수 r, θ, \dot{r} , $\dot{\theta}$ 를 출력하고 다시 LQR제어기와 CMAC 제어기로 피드백되어 순서 1에서부터 다시 반복한다. 훈련 과정에서 함수 사이의 상관관계는 다음과 같다.

$$\begin{aligned} V_L'' &= m(r, \dot{r}, \theta, \dot{\theta}) \\ V_L' &= f(s) \\ \Delta w &= g(V_L - V_L') / K \end{aligned} \quad (8)$$

여기서

- m = 각 상태를 이상적인 제어값으로 변환하는 함수
 - f(s) = CMAC 함수
 - g = 가중치 조절 인자 (0 < g < 1)
 - Δw = 가중치 증분 (초기상태에서 w은 모두 0)
 - V_L = 선형 제어기와 CMAC제어기 출력 값의 합
- 이다. 훈련의 처음 시작은 V_L과 V_L'의 차이 비교에 의

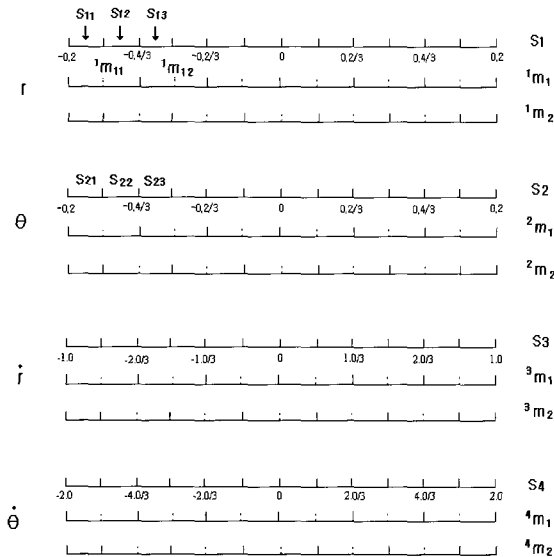


그림 4. 도립진자의 입력변수 및 중간변수의 불연속화.
Fig. 4. Discretization of the state and intermediate variables.

해 시작된다. 두 값의 차가 어떤 허용 가능한 오차의 범위를 벗어났을 때, (8)의 세 번째 식으로 가중치 증분을 계산하고, 이 가중치 증분은 원래 있던 CMAC 메모리 테이블의 위치에 있는 가중치에 합해지게 된다. 다음에 같은 입력변수 값이 주어지면 이미 학습된 가중치가 출력이 되어 V_L' 을 생산하게 되는 것이다. 이렇게 훈련을 반복할수록 V_L' 은 이상 제어값 V_L'' 로 가까워지게 된다.

그림 3에서 도립진자의 제어를 위해 사용되는 입력변수는 카트의 변위 r 과 속도 \dot{r} , 진자의 각변위 θ 와 각속도 $\dot{\theta}$ 로 구성된 4차원 입력공간으로 이루어진 연속적인 변수이다. 이들 연속적인 변수를 CMAC에 적용하기 위해서는 적절한 해상도로 불연속화 시켜야 한다. 본 연구에서는 먼저 기존의 선형 제어기로 모의실험을 한 후 그 결과를 바탕으로 하여 상태변수 값의 최대값과 최소값을 파악하여 그림 4처럼 불연속화 하였다. 그림 4에서 입력 변수 축은 4개 ($r, \theta, \dot{r}, \dot{\theta}$)이고 이 축들은 각각의 일정한 해상도로 12개의 불연속화된 입력 값으로 나누었으며 중간변수 축은 2개로 두었고 각 입력축의 해상도의 2배의 해상도로 불연속화 하였다.

본 연구에서 사용한 도립진자 시스템은 그림 5에서 보는 바와 같이 정해진 길이의 바닥 위에 설치되어 직선운동을 하는 카트와 그 카트에 설치된 축을 중심으로 회전운동을 하는 진자로 되어있다. 그림 5의 도립진자 시스템에 대한 운동방정식은 다음과 같다[6,7].

$$(M+m)\ddot{r} + m\ell\cos\theta\ddot{\theta} = f(t) - F\dot{r} + m\ell\sin\theta\dot{\theta}^2 \quad (9)$$

$$J\ddot{\theta} + m\ell\cos\theta\dot{r} = mg\ell\sin\theta - c\dot{\theta}$$

선형제어기를 설계하기위해 (8)의 방정식을 진자의 불안정평형상태에 대해 선형화하였으며 얻은 시스템 행렬은 다음과 같다.

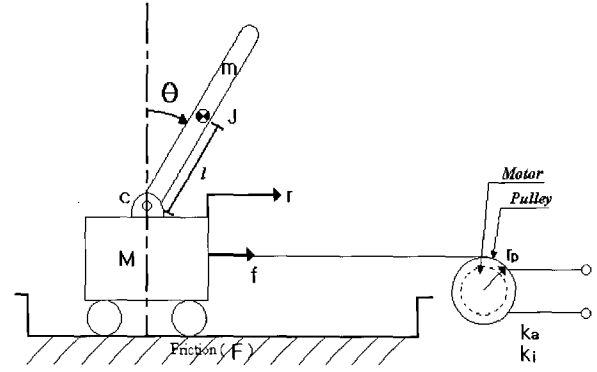


그림 5. 도립진자 시스템의 개략도.
Fig. 5. Schematic of inverted pendulum system.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -1.5358 & -0.81239 & 0.046848 \\ 0 & 32.714 & 2.3424 & -0.99789 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 5.2183 \\ -15.046 \end{bmatrix} \quad (10)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

이 시스템 행렬을 얻기 위해 (11)로 주어지는 모터 토크와 힘 관계식을 (9)에 대입하였고, 결과적으로 제어 입력은 모터 전력증폭기 전단에 가해지는 전압 $u(t)$ 이다.

$$f(t) = \frac{\Gamma_m}{r_p} = \frac{k_a k_i}{r_p R_a} u(t) \quad (11)$$

여기서

- Γ_m = 모터의 토크
- r_p = 풀리의 반지름
- k_i = 모터 토크상수
- R_a = 모터 아마추어 코일 저항
- k_a = 전력증폭기 이득

$u(t)$ = 제어 입력(전력증폭기 입력 전압)

LQR제어기 설계를 위한 성능지수는 다음과 같이 정의하고 상태가중치 행렬 Q 와 제어가중치 행렬 R 의 값을 참고 문헌[7]이 제안하는 값을 채택하였으며 상태피환제어 행렬 K 값을 얻었다.

$$J = \int (x^T Q x + u^T R u) dt$$

$$Q = \begin{bmatrix} 0.079 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.0032 & 0 \\ 0 & 0 & 0 & 0.0001 \end{bmatrix} \quad (12)$$

$$R = 0.00016$$

$$K = [-20.8071 \quad -43.2647 \quad -14.0068 \quad -7.3588]$$

또한 극점이 $[-40 + 5i, -40 - 5i, -50, -60]$ 인 상태변수 산출기 행렬을 구하였고 그 값은 다음과 같다.

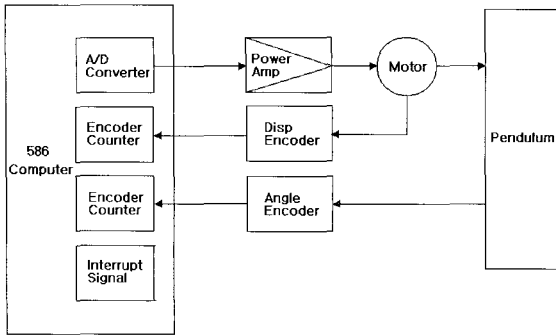


그림 6. 역진자 제어 장치의 구성.
Fig. 6. A schematic of pendulum control system.

$$L = \begin{bmatrix} 0.3866 & 0.0017 \\ 0.0102 & 0.3044 \\ 11.0152 & 0.1036 \\ 0.9710 & 6.9813 \end{bmatrix} \quad (13)$$

그러므로 (12)의 행렬을 이용하여 상태변수를 추정하고 추정된 상태변수를 이용하여 참고 제어값인 상태회환제어기 값을 다음과 같이 구할 수 있다.

$$U_{lqr} = u(t) = -KS \quad (14)$$

IV. 도립진자를 이용한 실험

실험을 위하여 제작된 도립진자 시스템의 개략적인 구성은 그림 6과 같다. 카트의 변위 및 진자의 각변위 측정을 위해 rotary incremental encoder를 사용하였다. 사용된 제어용 컴퓨터는 펜티엄급 프로세서를 탑재하고 있으며 LQR제어기의 샘플링 시간은 1/300초이며 CMAC제어기는 프로세서의 여유시간에 처리되며 초당 40회 내외의 샘플링 시간을 갖는 것을 확인하였다.

본 실험에서는 실제 시스템에서 제안된 제어계의 성능을 비교하여 그 결과를 확인 해보고 또한 매개변수의 변화에 대한 강건성을 알아보려고 한다. 실험순서로는 먼저 LQR제어기 단독의 응답과 CMAC learning controller의 제어응답을 비교하고, 다음으로 대표적 매개변수인 진자의 질량을 바꾼 후의 제어응답을 살펴 제안된 제어기가 모델의 불확실성에 대해 강건함을 보이겠다.

실제계와 모델의 매개변수가 같은 경우의 실험으로 LQR 제어계만의 제어응답과 CMAC learning controller를 사용한 제어응답에서 그 성능의 차이를 비교하였다. CMAC은 일련의 훈련과정을 거쳐 각각의 입력변수에 대한 제어값을 가지는데 한번의 입력과 출력이 실행되는 동안 한번의 훈련을 한다. 본 실험에서는 이런 과정을 두 가지 방법에서 실험하였다. 첫 번째는 도립진자 시스템이 작동되는 동안 계속 훈련을 반복하면서 향상된 출력값을 내어놓는 방법이다. 즉 기억장소(memory)의 값이 처음에는 빈 공간의 상태에서 연속적으로 가중치가 향상되면서 이 향상되었던 값들이 출력되는 것이고, 두 번째는 첫 번째 방법으로 훈련이 이루어진 CMAC의 기억장소를 더 이상의 훈련이 없이 이용하는 것이다. 본 연구에서는 편의상 첫 번째 방법을 이용한 CMAC learning controller를 “훈련 중의 CMAC learning

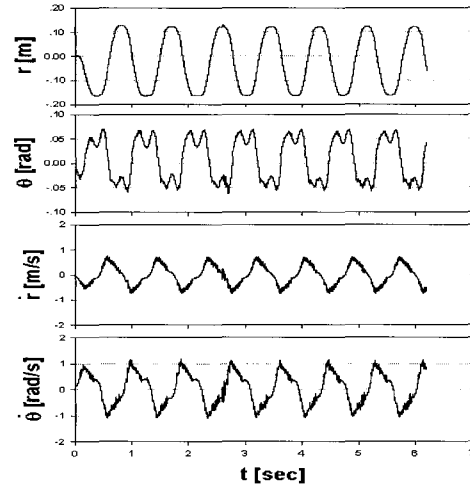


그림 7. LQR 제어기 응답.
Fig. 7. Response of LQR control.

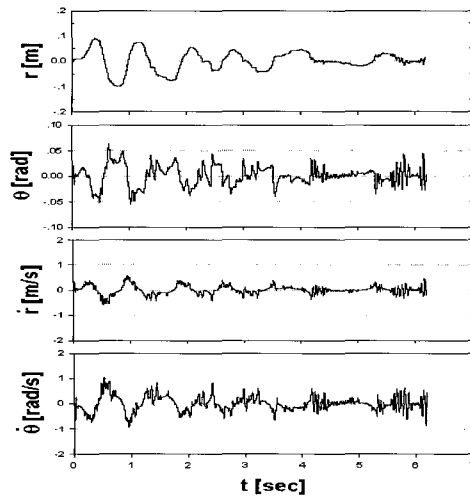


그림 8. 훈련 중 CMAC 제어기 응답.
Fig. 8. Response of CMAC under learning.

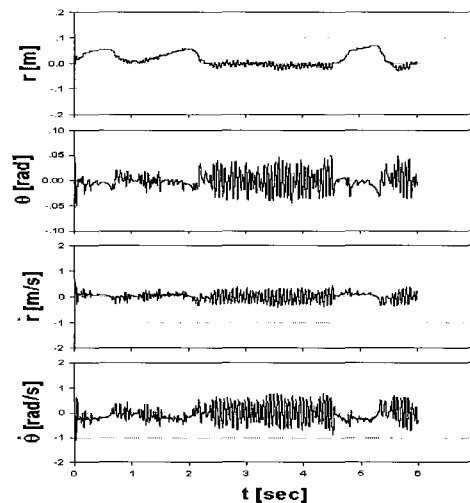


그림 9. 훈련 후 CMAC learning controller 응답.
Fig. 9. Response of CMAC after learning.

controller”라 하고 두 번째 방법을 이용한 CMAC learning controller를 “훈련 후의 CMAC learning controller”라 한다.

그림 7은 LQR제어기만의 제어응답이고 그림 8은 훈련 중의 CMAC learning controller, 그림 9는 훈련 후의 CMAC learning controller의 응답이다. 이 실험에서는 오차 정정 이득인자 g 를 0.05로 두었고 다른 주요 매개변수는 표 1과 같다.

실험 결과 그림 7에서처럼 LQR제어기만으로 제어한 경우는 카트의 변위 r 과 진자의 각변위 θ 가 일정한 주기로 헛헛하는 현상을 보인 반면, 그림 8에서 훈련 중의 CMAC learning controller를 사용한 제어응답은 시간이 흐를수록 그 변위가 작아지면서 안정되는 것을 볼 수있다. 헛헛 현상은 카트와 레일 사이의 비선형 마찰이나 카트의 롤링 등과 같이 모델링에서 제외된 요소에 기인한다. 반면 훈련 중의 CMAC learning controller는 도립진자의 시스템의 비선형성을 훈련을 통하여 제어하기 때문에 LQR제어기보다 더욱 안정된 결과를 보여준다. 그림 9은 훈련 후 CMAC learning controller의 제어응답 그래프이다. 훈련 후의 제어기도 진자의 각변위를 빠른 시간 내에 안정화시키는 것을 볼 수 있다.

제안된 제어기의 모델 불확실성에 대한 강건성을 시험하기 위해 진자의 질량을 바꾼 도립진자의 매개변수는 표 2와 같다. 그림 10은 변경 전 매개변수로 설계된 LQR 제어기만으로 도립진자를 제어했을 때의 응답이다. 진자의 질량이 변화하였을 때 LQR제어기는 얼마 후 갑작스런 동작을 보이며 도립진자를 제어하는데 실패하였다. 그림 11은 훈련 중의 CMAC learning controller를 사용하여 도립진자를 제

표 1. 도립진자 시스템의 매개변수.

Table 1. Parameters of pendulum system.

진자의 질량, m	0.131kg
진자의 길이, l	0.255m

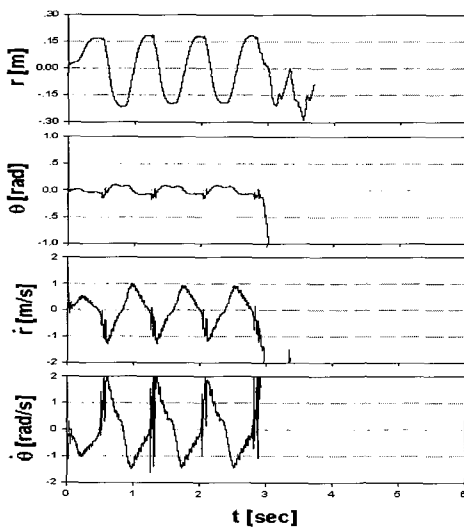


그림 10. 질량이 다른 진자로 바꾼 후 LQR 제어기 응답.
Fig. 10. Response of LQR with change in mass.

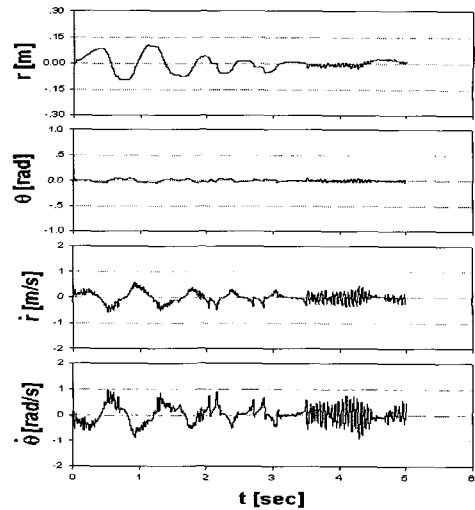


그림 11. 다른 질량의 진자로 바꾼 후 훈련중의 CMAC learning controller 응답.

Fig. 11. Response of CMAC under learning with changed pendulum in mass.

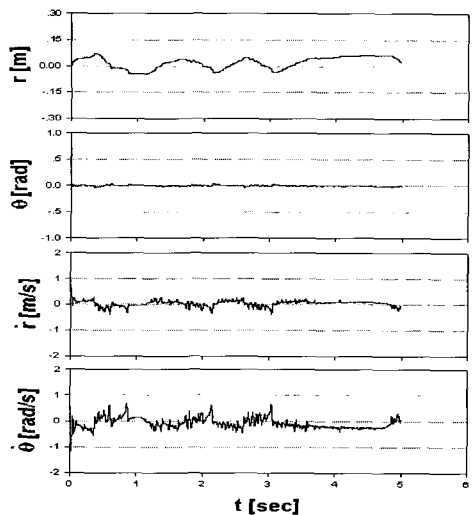


그림 12. 다른 질량의 진자로 바꾼 후 훈련 후의 CMAC learning controller 응답.

Fig. 12. Response of CMAC after learning with change in mass.

표 2. 변경된 시스템의 매개변수.

Table2. Altered system parameters.

진자의 질량, m	0.172kg
진자의 길이, l	0.255m

어한 결과이며 질량의 변화에도 불구하고 시간이 지나면서 안정되는 것을 알 수 있다. 이는 LQR제어기에 비해 훈련 중 CMAC learning controller가 상당히 안정적이고 강인하다는 것을 나타낸다. 그림 12에서 훈련 후의 CMAC learning controller는 훈련 중의 CMAC learning controller에 비해 카트의 변위에서 불안한 현상을 보이는데 이는 CMAC의 기

역장소에 저장되어진 가중치가 진자의 질량을 바꾸기 전의 상황에서 얻어진 값들이기 때문이라 추정한다.

이상의 실험에서 LQR제어기는 도립진자를 제어하는데 있어서 시스템 매개변수의 변화가 없는 상황에서는 안정된 제어를 하였지만, CMAC learning controller에 비해 카트의 막대의 변위가 크고, 범위의 변동이 심하며, 시스템의 매개변수가 변형되었을 때는 도립진자의 제어에 실패하는 불안한 결과를 나타내었다. 반면에 CMAC learning controller는 시스템의 매개변수의 변화에도 크게 영향을 받지 않고 시간이 흐를수록 안정된 제어를 보여주었고, 또한 훈련 후의 제어기보다는 훈련 중의 제어기가 안정성에 있어서 뛰어난 것을 알 수 있었다.

V. 결론

CMAC(Cerebellar Model Articulation Computer)은 입력과 응답으로 정해지는 계통의 상태공간의 학습과, 광범위한 비선형 함수 관계의 학습에 적용할 수 있으며, 학습 알고리즘은 적은 횟수의 반복적인 훈련으로도 수렴하는 장점이 있다. 이런 기법에 기초하여 Thomas Miller는 산업용 로봇 팔의 제어를 위한 새로운 학습 기술(CMAC learning controller)을 제안하였다. 하지만 함수의 정확도를 높일 목적으로 분해능을 높이거나, 혹은 변수가 많을 경우 대단히 많은 메모리를 필요로 한다는 단점이 있다. 그 외에 알고리즘을 수행하는데 많은 프로세싱 파워가 필요한데 실시간 제어를 목적으로하는 경우 고성능의 프로세서를 필요로하거나 혹은 병렬 프로세싱 기법을 사용해야하는 등 복잡성은 실시간 제어에 걸림돌이 된다.

본 연구에서는 프로세싱 파워에 대한 해결책으로 다중 샘플링 시간 방법을 제안하였다. 즉 전통적인 제어기를 기준 제어기 입력으로 사용하고, 이 입력과 계통의 응답을 CMAC의 훈련 데이터로 사용하는 것인데 이때 기준 제어기는 작은 샘플링 시간으로 고속 실행시키고, CMAC제어기는 나머지 여분의 프로세싱 파워를 이용하는 방법이다. 시험 대상으로는 도립진자를 이용하였고 제어를 통해 그 성능과 외란에 대한 강건성을 보였다.

실험 결과 LQR제어기는 카트의 변위와 진자의 각 변위가 불안정 평형점을 주위로 주기적으로 진동하는 헌팅 현상을 보였다. 이런 헌팅 현상은 모델링에서 제외된 마찰 등의 비선형성 때문이며, 계통의 매개변수가 변화되었을 때 외란에 적응하지 못하고 제어에 실패함을 확인하였다. 반면

에 CMAC learning controller를 이용하였을 때에는 시간이 흐를수록 카트의 변위와 진자의 각 변위가 줄어들며 안정되는 결과를 나타내었고 훈련 중의 CMAC learning controller의 제어 응답은 훈련 후의 CMAC learning controller에 비교해 안정된 상태로 도립진자를 제어함을 알 수 있었다. 또한 진자의 질량에 변화를 주었을 때에도 CMAC learning controller는 도립진자를 제어하는데 별다른 영향을 받지 않고 성공적으로 제어할 수 있음을 알 수 있었다. 이로서 CMAC learning controller의 다중 샘플링 알고리즘의 산업용 제어기로서의 가능성과 외란에 대한 강건성을 실험적으로 보였다.

참고문헌

- [1] J. S. Albus, "A new approach to manipulator control: The Cerebellar Model Articulation Computer(CMAC)," *Journal of Dynamic Systems, Measurement, and Control*, vol. 97, no. 3, pp. 220-227, 1975.
- [2] M. W. Thomas III, R. P. Hewes, F. H. Glanz, and L. G. Kraft III, "Real-time dynamic control of an industrial manipulator using a neural-network-based learning controller," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 1, pp. 1-9, February 1990.
- [3] M. W. Thomas III, "Real-time neural network control of a biped walking robot," *IEEE Control Systems*, pp. 41-48, February 1994.
- [4] D. E. Thompson and S. Kwon, "Neighborhood sequential and random training techniques for CMAC," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 196-202, January 1995.
- [5] 권성규, "CMAC의 S-to-M 변환을 위한 알고리즘," 대한기계학회논문집(A), 제 20 권, 제 10 호, pp. 3135-3141, 1996.
- [6] 이병수, "레일 위에 설치된 2자유도 비선형 역진자의 스윙업 회피 제어," 계명대학교 산업기술연구소 논문보고집, 제 23-1 호, pp. 1-9, 2000.
- [7] S. Mori, H. Nishihara, and K. Furuta, "Control of unstable mechanical system: control of pendulum," *International Journal of Control*, vol. 23, no. 5, pp. 673-692, 1976.

이 병 수



1980년 한양대학교 정밀기계공학과 졸업. 1982년부터 2년간 금성사 중앙연구소에서 연구원으로 근무. 1987년과 1991년 Washington State University에서 기계공학 석사 및 박사학위 각각 취득. 1991년~현재 계명대학교 기계자동차공

학부 교수. 관심분야는 기계 및 자동차 계통의 모델링 및 시뮬레이션, 동역학 해석 및 제어기 설계 및 실현.