

신경 회로망을 이용한 가변 구조 로봇 제어

이종수 최경삼 김성민

홍익대학교 공과대학 전기제어공학과

Variable Structure Control of Robot Manipulator Using Neural Network

Jong-Soo Lee Kyung-Sam Choi Seong-Min Kim

Dept. of Electrical & Control Eng.
Hongik University

Abstract

In this paper, we propose a new manipulator control scheme based on the CMAC neural network. The proposed control consists of two components. The feedforward component is an output of trained CMAC neural network and the feedback component is a modified sliding mode control. The CMAC accepts the position, velocity and acceleration of manipulator as input and outputs two values for the controller: One is the nominal torque used for feedforward compensation (M1 network) and the other is the inertia matrix related information used for the feedback component (M2 network). Since the used control algorithm guarantees the robust trajectory tracking in spite of modeling errors, the CMAC mapping errors due to the memory limitation are little worth consideration.

1. 서 론

현재의 산업용 로봇은 관절별 고정 이득의 선형 궤환 제어기 및 감속 기어를 사용한 동력 전달 장치를 사용하고 있으므로 고속의 정밀 작업이 어렵다.[1, 2, 3]. 전동기 축이 직접 로봇의 링크와 연결되는 직접 구동 방식을 이용하면 동력 전달 장치로 인한 오차를 줄이게 되어 정확한 경로 제어가 가능하다. 그러나 관절간의 상호 작용 토크 등이 감쇄없이 직접 전동기 축에 작용하게 되므로 로봇의 동력학을 고려한 제어(동적 제어) 방식이 필요하며, 토크 계산법 (Computed Torque Method) 이후 동적 제어에 대한 많은 연구가 발표되었다[4, 5]. 그러나, 초기에 제안된 많은 동적 제어방식이 의미를 갖기 위해서는 정확한 모델이 요구되었으며, 대표적인 동적 제어 방식인 토크 계산법의 경우 로봇 모델의 오차, 특히 관성 행렬의 오차에 의해 시스템의 경로 추적 오차가 커지며 안정성을 상실할 수 있다는 단점이 있다[6, 7]. 따라서 모델링 오차에도 불구하고 안정된 경로 추적을 보장하기 위한 동적 제어 알고리즘이 연구되었으며, 대표적으로는 적응제어[8, 9, 10] 및 가변구조 제어 [11, 12] 등이 있다. 적응제어 시스템은 정확한 동력학 정보가 필요하지 않지만 큰 계산시간을 요구되므로 실현에 불리하다. 가변구조제어 방식은 시스템의 정확한 모델을 필요로 하지 않으며 모델링 오차의 영향이 작다는 장점이 있어 로봇 제어에 적합하다. 이 방식은 슬라이딩 표면 부근에서 무한대 주파수의 입력 절환을 가정하고 있으나, 실제의 시스템에서는 계산 시간등으로 인하여 제어입력은 유한한 고 주파수 절환을 하게 되어 시스템의 무시된 고주파 성분을 여자시키는 떨림 현상이 야기된다[11]. 이를 방지하기 위해 슬라이딩 표면 주위에서 불연속 입력을 연속 함수로 보간하는 연구가 진행되었으나[13, 14, 15], 매 표본 주기마다 모델 계산을 포함한 많은 연산이 필요하고 입력은

시간지연을 갖게 되어 경로오차가 커지며 떨림 현상이 나타나게 된다. 즉, 궤환 제어 입력의 계산에 로봇 모델의 계산이 포함되는 것은 입력의 시간지연을 발생하므로 경로 오차가 커지며 시스템의 안정성을 상실하게 된다. 이를 피하기 위해 규정 경로에 대해 모델을 이용하여 계산한 토크 오크로서 관절간의 비선형 결합 토크를 근사적으로 전향 보상하고, 그 보상 오차를 간단한 PD제어기 등을 이용하여 궤환 제어하는 방식이 현실적이다[16]. 그러나 이 방식은 모델 오차 및 계산 시간 지연으로 인한 보상 오차가 커지게 되면, 정확한 경로 제어를 수행할 수 없다.

본 논문에서 제시하는 제어기의 구조는 전향보상 성분과 가변구조 형태의 궤환 성분으로 구성되어 있으며 시스템의 안정성에 대한 해석이 이루어져 있다[17, 18]. 궤환 성분은 보간된 가변구조의 형태로서 강인한 경로추적 특성을 갖고 있으므로 모델오차 및 시간지연으로 인한 전향보상의 보상 오차를 어느 정도 허용하는 특징이 있다. 궤환 성분은 구조가 단순하고 계산량이 매우 작아 시간 지연을 무시할 수 있으며 실현에 유리하나 모델의 관성행렬에 대한 정보를 필요로 하며, G-D[19] 알고리즘등을 고려할 수 있다.

이 제어를 실현하기 위하여는 첫째, 근본적으로 입/출력 벡터간의 비선형 Mapping으로서 많은 계산을 요구하는 전향보상 입력 및 관성 행렬의 계산을 가능한 짧은 시간에 수행하기 위한 알고리즘 및 계산 구조가 필요하며, 둘째, 모델 오차를 줄이기 위해 로봇의 물리적 계수(길이, 질량, 관성 및 마찰등)에 대한 정확한 계측 혹은 추정 알고리즘이 필요하며, 셋째, 제어기를 구성하는 두개의 성분이 서로 다른 표본시간을 가지므로 정보 처리를 위한 시간 계획이 필요하다 등의 사항을 고려하여야 한다. 따라서, 본 논문에서는 이러한 어려움을 피하기 위하여 동력학 Mapping의 계산을 CMAC(Cerebellar Model Articulation Controller) 구조의 신경 회로망에 의해 대체하는 방식을 제안한다.

1975년 J. Albus가 제시한 CMAC 모델은[20] 복잡한 비선형 함수의 해를 수치적 연산에 의하여 구하지 않고, 입/출력 Mapping을 테이블로서 작성해 두고, 임의의 입력에 대한 출력값을 찾는 방법이다. CMAC 회로망은 임의의 부드러운 연속 함수를 단순한 학습을 통해 실현할 수 있으며, 기존의 디지털 소자를 이용하여 실시간 동작이 가능하도록 설계할 수 있으므로 동력학 계산을 포함한 로봇의 제어에 매우 적합하다[21, 22]. CMAC를 이용하여 제어기를 구성할 경우, 계산 시간이 필요하지 않아 전향보상 성분과 궤환 성분을 짧은 시간에 구할 수 있으며, 동력학 Mapping을 학습만에 의해 얻을 수 있으므로 로봇의 물리적 계수에 대한 정보가 필요하지 않다는 장점이 있다. CMAC 회로망이 기억 용량의 제한으로 인하여 많은 Mapping 오차를 갖고 있음에도 불구하고, 제안한 제어 방식은 모델링 오차에도 불구하고 강인한 경로 추적 안정성을 보장하는 특징이 있으므로 이를 충분히 보상할 수 있다.

2. 제어 알고리즘

로봇 시스템의 동력학을 나타내는 일반적인 식은

$$\tau = D(\theta) \ddot{\theta} + h(\theta, \dot{\theta}) \quad (1)$$

으로 표현할 수 있으며, 여기서 θ , $\dot{\theta}$ 및 $\ddot{\theta}$ 는 관절들의 회전각, 속도 및 가속도를 나타내는 n 차 벡터, τ 는 관절의 구동 토크를 나타내는 n 차 벡터이며, $D(\theta)$ 는 $n \times n$ 관성 행렬을 $h(\theta, \dot{\theta})$ 는 중력, 원심력 및 Coriolis력 등을 포함하는 n 차 비선형 결합 토크 벡터를 나타낸다. 대부분의 동적 제어 알고리즘에서는 동력학을 고려한 토크 정보를 필요로 하며, 이를 위해 로봇의 물리적 규격(길이, 질량, 관성 및 마찰 계수 등)에 대한 정보를 알고 있다는 가정하에 순환형 Newton-Euler 알고리즘 등을 이용하여 식 (1)로 표현되는 동력학 계산을 수행 한다. 그러나 이러한 계산 알고리즘을 이용하여 계산하는 동력학 모델은 물리적 규격의 오차, 마찰과 백-래쉬 등에 기인하는 오차 및 작업 부하의 변화에 의한 오차등으로 인하여 실제의 시스템 (1)과는 무시할 수 없는 오차를 갖게 된다. 따라서, 우리가 이용 가능한 로봇의 모델은

$$\tau = \hat{D}(\theta) \ddot{\theta} + \hat{h}(\theta, \dot{\theta}) \quad (2)$$

으로 표현이 가능하다. 따라서 로봇 제어 문제는 오차가 포함된 모델 (2)를 이용하여 시스템 (1)이 원하는 운동 경로 O_d (\dot{O}_d 및 \ddot{O}_d 포함)를 운동하도록 하는 것이다. 본 논문에서 이용하는 제어 토크는 전향보상 성분과 수정된 가변 구조 제어 방식의 궤환 성분이 결합된 구조를 가지며

$$\tau(t) = \tau_{ff}(t) + \tau_c(t) \quad (3)$$

으로서 표현할 수 있고, 여기서 전향 성분 $\tau_{ff}(t)$ 는

$$\tau_{ff}(t) = \hat{D}(O_d(t)) \ddot{O}_d(t) + \hat{h}(O_d(t), \dot{O}_d(t)) \quad (4)$$

으로, 궤환 성분 $\tau_c(t)$ 는

$$\tau_c(t) = -\hat{D}(O_d(t)) u(t) \quad (5)$$

$$u(t) = k\dot{e}(t) + k_s s(t) + k_o \sigma(t) \quad (6)$$

으로 주어지며, $\sigma(t) = [\sigma_1(t), \dots, \sigma_n(t)]^T$ 는

$$\sigma_i(t) = \frac{s_i(t)}{|s_i(t)| + \Delta} \quad (7)$$

이다. 여기서 $s(t)$ 는

$$s(t) = \dot{e}(t) + ke(t) \quad (8)$$

으로 정의되는 슬라이딩 표면으로서 $e(t)$ 는 $\theta_d(t) - \theta(t)$ 로서 정의된 경로 오차이다. 위의 제어기는 식 (7)에 의해 항상 연속적인 입력이 되도록 보간함으로서 불연속 입력 절환을 방지하게 되어 가변 구조의 단점인 떨림 현상이 제거된다. 식(4) 와 식(5)에서 사용한 $\ddot{O}_d(t)$, $\dot{O}_d(t)$, $\ddot{O}_d(t)$ 는 원하는 경로 $O_d(t)$, $\dot{O}_d(t)$, $\ddot{O}_d(t)$ 로 부터 큰 시간간격(T_s)으로 표본을 취한 계단 함수임을 나타내고 있다. 즉, 모든 $t \in [kT_s, (k+1)T_s)$ 에 대하여

$$O_d(t) = O_d(kT_s), \dot{O}_d(t) = \dot{O}_d(kT_s), \ddot{O}_d(t) = \ddot{O}_d(kT_s)$$

이 된다. 위의 (3)-(8) 로서 주어 지는 제어 입력을 실제 시스템 (1)에 인가한 경우의 경로 추적 안정성에 대한 해석은 이루어져 있으므로 생략하기로 한다[17, 18]. 다음 절에서 기술하는 CMAC 신경 회로망은 위의 제어 입력을 실현하기 위한 두가지의 정보를 제공하도록 학습 받는다. 하나는 식 (2)로 표현된 로봇 시스템의 근사적인 모델을 대체하는 Mapping으로서의 CMAC로서

$$M1 : \ddot{O}_d(t), \dot{O}_d(t), O_d(t) \Rightarrow \tau_{ff} \quad (9)$$

이며, 다른 하나는 식(5)의 Mapping을 대체하는 CMAC인

$$M2 : O_d(t), u(t) \Rightarrow \tau_c \quad (10)$$

이다. 이를 이용하여 제시된 제어 알고리즘 (3)-(8)을 실현하기 위한 블록 선도는 그림 1과 같다.

3. CMAC의 구조 및 학습

로봇의 동력학을 고려한 제어를 실현하기 위하여 다층 Perceptron 구조에 EBP(Error Back Propagation)을 이용한 방식등도 연구되고 있으나, 로봇 제어에 필요한 복잡한 비선형 함수 Mapping을 처리하는 데에는 CMAC를 이용한 제어 방식이 가장 적합하다고 판단된다. 그 이유로서 CMAC를 이용한 로봇 제어 시스템은

첫째, 동력학 방정식에 대한 정보가 필요 없으며, 시스템에서 측정된 관측 데이터만을 이용하여 동력학 계산이 가능하다. 따라서 특히 복잡한 구조를 가진 여유 자유도의 로봇 시스템에 대한 적용에서 유리하다.

둘째, CMAC의 학습에 필요한 계산량은 자유도에 선형 비례하여 증가할 뿐이므로, 현재 이용 가능한 계산 소자로서 실현이 가능하다.

셋째, 학습이 완료된 후에는 정확한 동력학 정보를 복잡하고 방대한 량의 계산이 필요없이 테이블 조사 방법에 의해 구할 수 있으므로 실시간 실현에 매우 유리하다. 는 등의 장점이 있음을 들 수 있다.

3.1. CMAC의 구조[20]

CMAC는 "Table look-up 방법"을 기본으로 하여 임의의 비선형 Mapping을 구현하는 알고리즘이다. CMAC를 이용하여 로봇의 동력학 모델 (2)를 계산하는 과정은 $3 \cdot n$ 차 입력 벡터 $q = [\theta, \dot{\theta}, \ddot{\theta}]^T$ 에 대응하는 n 차 출력 토오크 벡터의 값 $\tau = f(q)$ 를 구하는 Mapping을 구현하는 과정이며, 이 구조는 그림 2와 같다. CMAC에서는 일차적으로 입력 공간 Q 에 있는 각각의 성분을 테이블 A 의 있는 C 개의 위치로 Mapping하며, 이어서 테이블 A 를 이보다 작은 용량의 기억 테이블 A' 로 균일하고 임의로 Mapping(Hash 코딩)한다. 이러한 과정을 통하여 단순한 테이블 조사법을 이용하는 경우에 요구되는 방대한 기억 용량의 크기를 축소시킬 수 있다. CMAC 회로망에서의 출력은 그림에서와 같이 각각의 입력이 Q 로부터 A , A 로부터 A' 으로의 Mapping 과정을 통하여 A' 테이블의 C 개의 위치로 Mapping된 후, 이 위치에 기억되어 있는 값(가중치)을 합하여 출력 $f(q)$ 를 계산하게 된다.

Q 로부터 A 로의 Mapping으로 인하여 입력 공간에서 가까운 거리의 입력 벡터들은 A' 에서 최소의 기억 위치를 가짐을 보장하게 되므로 유사한 입력들의 일반화[20]라는 바람직한 성질을 가지게 된다. 그러나 큰 용량의 테이블 A 로부터 작은 용량의 테이블 A' 으로의 임의의 Mapping으로 인하여, 입력 공간에서 멀리 있는 입력들이 A' 테이블에서 일부 위치를 공유하는 Mapping충돌 현상이 발생할 확률이 있게 된다. 일반적으로 $3n$ 차 입력 벡터의 각각의 요소들이 가질 수 있는 값의 범위를 R 개 구간으로 분할할 경우, 이로서 조합이 가능한 입력 벡터의 경우의 수는 모두 R^{3n} 이 된다. 그러나 일반적으로 CMAC에서는 A' 의 용량을 총 경우의 수보다 작게 선정하므로 CMAC의 입력 벡터를 관심 있는 입력 조합(원하는 작업 경로 부근)에 한정해야만 한다.

3.2 CMAC의 학습 과정[21]

위에서 설명한 CMAC를 이용하여 로봇 제어 시스템의 동력학 Mapping을 학습 시키기 위한 블럭 선도는 그림 3과 같으며, 2절에서 제시한 제어기의 실현을 위해 식(9), (10)으로 표현되는 두가지의 CMAC회로망($M1$ 및 $M2$)을 학습시켜야 한다. 이를 위해 다음의 네 단계로 학습을 진행한다.

단계 1 : 회로망 CMAC1의 학습

학습을 위한 경로로서 $\theta_d, \dot{\theta}_d$ 및 $\ddot{\theta}_d$ 를 이용하여 다음과 같은 절차로 수렴이 완료될 때까지 반복하여 학습을 진행한다.

- ① 각 제어 주기마다 원하는 경로 $q_d = [0_d, \dot{0}_d, \ddot{0}_d]^T$ 를 계산 한다.
- ② 측정된 시스템의 상태 $q = [\theta, \dot{\theta}, \ddot{\theta}]^T$ 를 고려하여, 제어 경로 $q_c = [0, \dot{0}, \ddot{0}_c]^T$ (11)를 구성한다. 여기서 $\ddot{0}_c$ 는 이 가속도를 유지할 경우 현재의 위치(0)를 앞으로 $5T$ (T : 표본 구간)후에 원하는 위치(θ_d)가 되도록 결정한다.
- ③ q_c 를 CMAC1 회로망에 입력하여 출력 $f(q_c)$ 을 구한다.
- ④ $\tau = f(q_c) + k [q_d - q]$ 를 로봇트에 인가 한다.
- ⑤ τ 와 $f(q_c)$ 를 이용하여 q 에 대응되는 C 개의 기억 위치(가중치)를

$$\frac{\beta [\tau - f(q_c)]}{C} \quad (12)$$

만큼 변화시킨다.

여기서, 학습 인자 $0 \leq \beta \leq 1$ 이며, $\beta = 1$ 일때는 학습 후에 $f(q_c) = \tau$ 가 되며, $\beta < 1$ 일 때에는 $f(q_c)$ 는 τ 의 방향으로 변화가 된다. 단계 1의 학습이 완료된 후에 CMAC 회로망은 식 (2)와 동등한 의미로서 $\tau = D(\theta) \ddot{\theta} + h(\theta, \dot{\theta})$ 의 Mapping을 나타내게 된다.

단계 2 : 회로망 CMAC2의 학습

- ① 각 제어 주기마다 경로 계획부는 $\theta_d, \dot{\theta}_d$ 및 $\ddot{\theta}_d \equiv 0$ 을 이용하여 단계 1과 같은 절차로 학습을 진행한다.
- ② ①에서 가속도 경로를 0으로 선택하였으므로 학습이 완료된 후의 CMAC는 $\tau = h(\theta, \dot{\theta})$ 을 나타내는 Mapping이 된다.

- ① 원하는 경로 $q_1 = [0_d, \dot{\theta}_d, \ddot{\theta}_d]^T$ 를 발생하여 단계 1 에서 학습된 CMAC1 에 입력하여 출력

$$\tau_1 = D(\theta_d) \ddot{\theta}_d + h(\theta_d, \dot{\theta}_d) \quad (13)$$

- 를, $q_2 = [0_d, \dot{\theta}_d, 0]^T$ 를 발생하여 단계 2에서 학습된 CMAC2 회로망에 입력하여 출력

$$\tau_2 = h(\theta_d, \dot{\theta}_d) \quad (14)$$

를 얻는다.

- ② 이제 따라서 우리가 원하는 Mapping인

$$\tau_c = \tau_1 - \tau_2 = D(\theta_d) \ddot{\theta}_d \quad (15)$$

- 를 얻기 위하여 $q_3 = [0_d, 0, \ddot{\theta}_d]^T$ 를 발생하여 M2 회로망에 입력하여 출력 τ_3 를 얻는다.

- ③ M2 회로망에서 q_3 에 대응되는 C개의 기억 위치를

$$\frac{B \{ \tau_c - \tau_3 \}}{C} \quad (16)$$

만큼 변화시킨다.

회로망 M2의 학습 과정을 블럭 선도로 도시하면 그림 4와 같다.

단계 4 : 회로망 M1 의 구성

M1 회로망으로서 CMAC1을 이용할 수도 있으나 이보다 작은 기억 용량을 가지는 테이블에 의하여 같은 출력을 얻도록 구성하는 것이 유리하므로 그림 5와 같이 M2 및 CMAC2 회로망을 이용할 수 있으며 이의 절차 다음과 같다.

- ① $q_2 = [0_d, \dot{\theta}_d, 0]^T$ 를 구성하여 단계 2에서 학습된 CMAC2 회로망에 입력하여 출력 $h(\theta_d, \dot{\theta}_d)$ 를 얻는다.

- ② $q_3 = [0_d, 0, \ddot{\theta}_d]^T$ 를 구성하여 단계 3에서 학습된 M2 회로망에 입력하여 출력 $D(\theta_d) \ddot{\theta}_d$ 를 얻는다.

- ③ CMAC2 회로망과 M2 회로망의 출력을 합하여 $D(\theta_d) \ddot{\theta}_d + h(\theta_d, \dot{\theta}_d)$ 의 출력을 발생한다.

3.3 제안한 CMAC 방식의 검토

본 논문에서 제시한 CMAC회로망은 학습된 M1과 CMAC2 의 두개의 회로망을 이용하여 2절에서 제시된 제어 입력을 실현 하기 위해 필요한 M1 과 M2 의 Mapping을 구성하게 되는 구조를 갖고 있다. 특히 M2 회로망을 별도의 테이블에 항상 기억하지 않고 M1과 CMAC2에 의해 구하도록 함으로써, 필요한 기억 용량의 크기를 감소시킬 수 있게 된다. 즉 M2회로망에서는 입력 벡터의 경우수가 모두 R^{3n} 인 반면에, M1이나 CMAC2 회로망에서는 입력 벡터의 차수가 $2n$ 이므로 각각 R^{2n} 개의 입력 경우 수만이 있게 되므로 기억 용량이 감소하게 된다.

CMAC를 이용해 세밀 구간의 모든 입력에 대한 Mapping 정보를 지원하는 것은 매우 큰 기억 용량을 필요로 하므로 실현이 거의 불가능하나, 초집적 기억 소자 제조 기술의 개발에 의해 여건이 개선되고 있다. 테이블 기억 용량의 한계로 인해 입력의 범위를 큰 구간으로 나누어 Mapping정보를 학습시키는 경우, CMAC회로망이 제공하는 동력학 정보에는 Mapping 오차가 있게 되나 본 논문에서 제시한 제어 방식은 모델링 오차에도 불구하고 강인한 경로 제어 안정성을 보장하므로 이를 보상할 수 있다.

4. 결론

본 논문에서 제시하는 제어기는 학습된 신경 회로망의 출력인 feedforward 성분과 가변구조 제어 방식의 궤환 제어 성분으로서 구성되어 있다. 신경 회로망은 CMAC와 유사한 구조를 가지며 원하는 작업 경로의 위치, 속도 및 가속도 정보를 입력으로 받아 두가지 출력을 제공하도록 학습 받는다. 하나의 출력은 로봇의 동력학을 고려한 feedforward 보상 토오크이며 또 하나의 출력은 궤환 제어의 계산에 필요한 관성 행렬의 정보이다. CMAC 신경 회로망의 특성상 제공하는 로봇의 동력학 정보는 오차를 가지게 되나 가변 구조 방식의 궤환 제어 성분이 충분히 보상할 수 있을것으로 예측할 수 있다.

감사 : 본 연구는 한국 과학 재단의 기초(신진)연구 지원을 받아 수행 되었습니다.

참고문헌

- [1] J.J. Craig, *Introduction to Robotics : Mechanics and Control*. Reading, Mass.: Addison-Wesley, 1986.
- [2] J.M.Brady et al.(eds), *Robot Motion:Planning and Control*. Cambridge, Mass.: MIT Press, 1982.
- [3] C.S.G. Lee, R.C. Gonzalez and K.S. Fu, *Tutorials on Robotics*(2nd ed.). Silver Spring, Md. : IEEE Computer Society Press, 1986.
- [4] K.S. Fu, R.C. Gonzalez and C.S.G. Lee, *Robotics : Control, Sensing, Vision and Intelligence*. New York: McGraw-Hill, 1987.
- [5] M.W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. New York : John Wiley & Sons, 1989.
- [6] O. Egeland, "On The Robustness of Computed Torque Technique in Manipulator Control", *Proc. IEEE Int. Conf. Robotics and Automation*, pp.1203-1208, 1986.
- [7] V.D. Tourassis and C.P. Neuman, "Robust Nonlinear Feedback Control for Robotic Manipulators", *IEE Proc.D*, pp.134-143, July 1985.
- [8] S. Dubowsky and D.T. DesForges, "The Application of Model Reference Adaptive Control to Robotic Manipulators", *Trans. ASME, J. Dynamic Systems Measurement and Control*, vol.101, pp.193-200, 1979.
- [9] A.J. Koivo and T.H. Guo, "Adaptive Linear Controller for Robotic Manipulators", *IEEE Trans. Automatic Control*, vol.AC-28, no.1, pp.162-171, 1983.
- [10] C.S.G. Lee and M.J. Chung, "Adaptive Perturbation Control with Feedforward Compensation for Robot Manipulators", *Simulation*, vol.44,no.3, pp.127-136, 1985.
- [11] V.I. Utkin, "Variable Structure Systems with sliding mode : A Survey", *IEEE Trans. Automatic Control*, vol.AC-22, pp.212-222, 1977.
- [12] K.K.D. Young, "Controller Design for a Manipulator Using Theory of Variable Structure Systems", *IEEE Trans. Sys. Man Cybern.*, vol.SMC-8, no. 2, pp.101-109, 1978.
- [13] J.J.E. Slotine, "The Robust Control of Robot Manipulators", *Int. J. of Robotics Research*, vol.4, pp.49-64, Summer 1985.
- [14] F. Harashima, H. Hashimoto, and K. Maruyama, "Practical Robust Control of Robot Arm Using Variable Structure System", *Proc. IEEE Int. Conf. Robotics and Automation*, pp.532-539, 1986.
- [15] E.G. Gilbert and I.J. Ha, "An Approach to Nonlinear Feedback Control with Application to Robotics", *IEEE Trans. Sys. Man Cybern.*, vol.SMC-14,no.2, pp.101-109, 1984.
- [16] C.H. An, C.G. Atkeson and J.M. Hollerbach, "Experimental Determination of the Effect of Feedforward Control on Trajectory Tracking Errors", *Int. Conf. Robotics and Automation*, pp.55-60, 1986.
- [17] 이종수, 로봇의 하이브리드 제어기 설계에 관한 연구, 박사학위논문, 서울 대학교, 1988.
- [18] J.S. Lee and W.H. Kwon, "A Multirate Robust Robot Control Algorithm", *Proc. IEEE TENCON*, Seoul, pp.909- 914, 1987.
- [19] C.S.G. Lee, B.H. Lee and R. Nigam, "Development of the Generalized d'Alembert Equations of Motion for Mechanical Manipulators", *Proc. IEEE 22nd CDC*, San Antonio, Tex., pp.1205-1210, 1983.
- [20] J.S. Albus, "A New Approach to Manipulator Control : The Cerebellar Model Articulation Controller", *Trans. ASME, J. Dynamic Systems Measurement and Control*, pp.220-227, 1975.
- [21] W.T. Miller III, F.H. Glanz and L.G.Kraft III, "Application of a General Learning Algorithm to the Control of Robotic Manipulators", *Int. J. Robotics Research*, vol.6, No.2, pp.84-98, Summer 1987.
- [22] V.A. Jennings, "Fork Lift Robot", *DARPA Neural Network Study* : AFCEA Int. Press, pp.445-450, 1987.

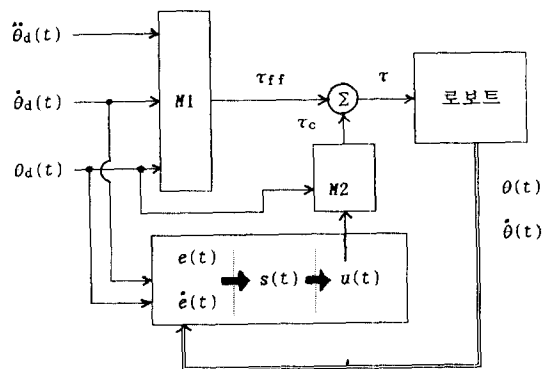


그림 1. 제안하는 제어기의 실현을 위한 블록 선도

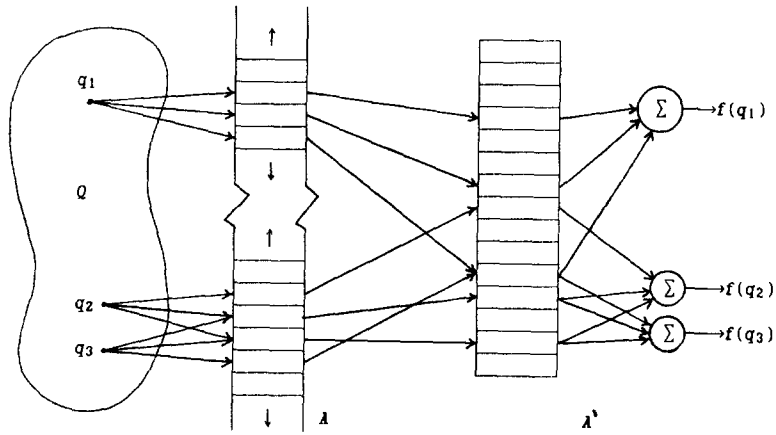


그림 2. 로봇의 동역학 계산을 위한 CMAC 구조[21]
(입력이 3개의 기억 위치로 Mapping되는 경우)

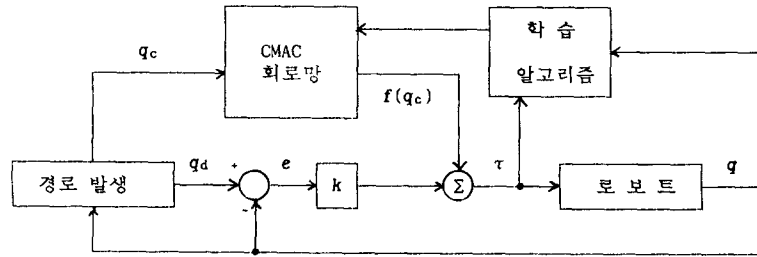


그림 3. CMAC1/CMAC2 회로망의 학습을 위한 블록 선도[21]

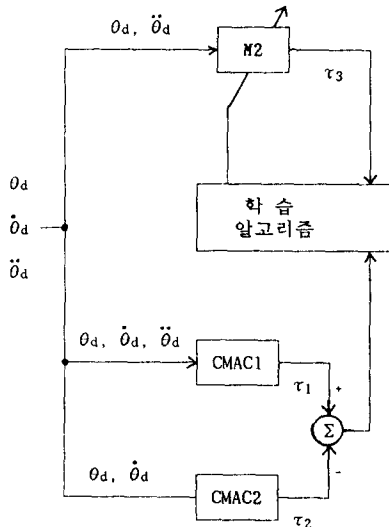


그림 4. M2 회로망의 학습을 위한 블록 선도

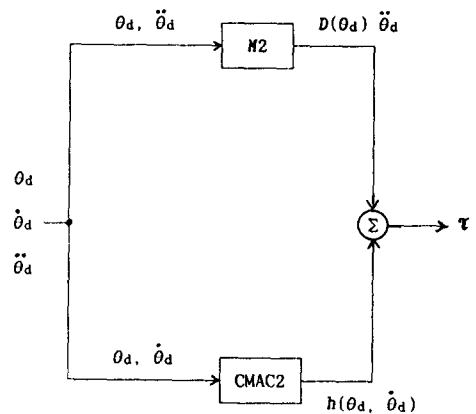


그림 5. CMAC2 와 M2 를 이용한 M1 회로망의 구성