

고차원 CMAC 문제의 소요 기억량 감축

Reducing Memory Requirements of Multidimensional CMAC Problems

권 성 규*
Sunggyu Kwon*

요 약

고차원 입력벡터를 갖는 CMAC 문제에 있어서 과도한 소요 기억량의 문제에 대처하기 위해, 공간에서의 운동을 연속적인 직교 단위 방향 운동으로 분해함으로써 직교좌표계에서의 변위와 관련한 고차원 CMAC 문제를 분할된 문제에 의한 저차원 CMAC들로 구성하는 CMAC 시스템을 연구하였다. 시스템의 구성과 훈련 문제를 논의하기 위해, 평면 운동을 하는 세 관절 매니플레이터의 역기구학 문제를 저차원 CMAC들로 구성된 시스템으로 전산 모사하여 시스템의 성능을 평가하였다.

ABSTRACT

In order to reduce huge memory requirements of multidimensional CMAC problems, building a CMAC system by problem decomposition is investigated. Decomposition is based on resolving a displacement vector in cartesian coordinates into unit vectors that define a few lower-dimensional CMACs in the CMAC system. A CMAC system for an inverse kinematics problem for a planar manipulator was simulated and the performance of the system was evaluated in terms of training and output quality.

I. 서 론

J. S. Albus는 1975년 CMAC(Cerebellar Model Articulation Controller)을 제안하면서, 로봇 팔의 제어에 CMAC을 이용하였다[1]. 그 이후 컴퓨터 비전(Computer Vision)이나 공정 자동차 시스템 개발에 이용한 예도 있지만, CMAC을 이용한 대부분의 연구는 매니플레이터의 제어에 관한 것들이었다. CMAC의 응용분야가 넓어지고 다양한 연구 결과가 발표되면서 [20, 21], CMAC 훈련의 문제점을 인식하게 되

고, CMAC의 훈련을 위한 방법들이 제안되었다[2-5]. 또한 고차원 CMAC 문제에 대처하기 위해 CMAC의 변환 알고리즘과 훈련 및 여러 매개 변수가 CMAC 성능에 미치는 영향 등에 대한 연구 결과들도 발표되고 있다[5-7].

CMAC을 이용한 제어에서는 제어계의 동특성을 분석하여 그 수학적 모델을 나타내는 복잡한 수식을 푸는 방법을 쓰지 않고, 제어함수를 CMAC이 학습하여 기억表(Look-up memory table)에 심고 실제의 작업에서는 제어상태에 해당하는 곳에 심어진 제어함수 값들을 합하여 CMAC의 제어 출력값으로 사용한다.

그런데, CMAC을 제어에 이용하데는 두가지 큰 단

*계명대학교 기계설계학과

점이 있다. 첫째로, 쓸 수 있는 만큼의 기억량을 고려하여 연속적인 입력변수들을 불연속화하여 양자화하므로 CMAC의 출력에는 어느 정도의 오차가 있게 마련이다. 그래서 만약 CMAC이 만드는 기억표의 크기가 매우 제한적이면, 원하는 신뢰도를 달성하기가 어렵다. 둘째로, 기억표의 크기가 CMAC을 구성하는 입력변수의 수에 기하학적으로 비례하여 커진다. 이 문제는 비록 메모리가 매우 싸지고 기억장치 기술이 괄목할만하게 발전하고 있다 하더라도, CMAC을 구성하는 입력변수의 수가 많을 때는 기억량이 엄청나게 커지므로, 심각한 문제가 아닐 수 없다. 이것은 기억표를 이용하는 다른 제어 방법에서도 마찬가지 문제이다.

기억표를 이용하는 제어 방법에서 소요되는 기억량을 줄이기 위해서, Configuration Space Method에서는[8] 운동방정식의 계수들이 단지 매니플레이터 팔의 자세에만 의존하는 특별한 방정식을 구성하고, 값이 0이 되는 계수들을 방정식에서 제거하여 기억소요량을 줄였다. 또한 매니플레이터의 힘 제어기에서 상태 입력 변수들의 양자화(quantization)를 최적화하여 소요 기억량을 최소화하는 방법도 제시하였다[9].

반면 많은 입력변수들로 인한 고차원의 복잡한 제어 문제들은 적은 수의 입력변수로 정의되는 덜 복잡한 저차원의 문제들로 분할하여 고차원의 대규모 기억표 대신 여러 저차원의 소규모 기억표를 구성하는 방법도 있다. Dynamic Legged System[10]의 제어 문제에서는, 문제를 따로 풀 수 있는 두 문제로 나누고, 그렇게 해서 나누어진 두 무리의 시스템 상태 변수들로 적당한 크기의 두개의 기억표를 만들었다. 7 자유도를 갖는 의학적 인공 팔의 제어 문제에서는[11], "Principle of Minimum Interaction"에 따라, 시스템을 자유도가 각각 1인 일곱 하부 시스템으로 나누어 제어하는 계층 제어 시스템을 제안하였다. 또한 자율주행 로봇을 위한 계층 제어 시스템에서는 [12] "Levels of Competence"를 정의하고 그 기준에 따라 문제를 계층별로 분할하고, 분할된 문제에 대한 제어기를 각각 구축하였다.

CMAC도 제어 수준의 계층별로 문제를 실제적으로 분할할 수 있는 구조로 되어있는데, Albus는 그의 초기 논문들[13, 14]에서 여러 입력변수들로 정의되는

복잡한 제어 문제에 CMAC을 이용하기 위한 제어 문제의 계층적 분할의 개념을 설명하였다[13, 14]. 최근의 연구 예로는 10 자유도를 갖는 두발 로봇(Biped Walking Robot)의 동적 균형을 위한 학습 제어 시스템을 구축하는 문제에서[15] 시스템내의 제어 문제의 계층에 따라 세개의 CMAC 제어기를 사용하였다.

이 논문에서는 CMAC을 정의하는 고차원 입력벡터로 인한 과도한 소요 기억량의 문제에 대처하기 위해서 공간에서의 운동을 연속적인 직교 단위 방향 운동으로 분할하여 변위와 관련된 고차원 벡터를 저차원 벡터들의 합으로 분해한다. 그래서, 고차원 벡터에 의한 고차원 CMAC 기억표 대신 여러 저차원 기억표들을 갖는 분할된 문제로 소요 기억량을 줄일 수 있는 방안을 연구한다. 평면 매니플레이터의 역기구학 문제를 예로 들어 문제의 분할로 구성된 두 층 CMAC 시스템을 훈련하고 성능을 평가하였다.

II. 다층 CMAC 시스템

앞에서 언급했듯이, CMAC을 제어기로 사용하는 데 있어서의 큰 단점 중의 하나는 CMAC을 구성하는 제어변수의 수가 많을 때 소요 기억량이 과도하게 증대하는 것이다. 예를 들어, 적은 수의 되먹임(feedback) 변수들을 사용하는 간단한 매니플레이터의 경우에 있어서는 단 하나의 CMAC이 기본적인 운동 지령을 받아서 관절 구동 신호를 직접 생산하는 것이 가능하다. 그러나, 여러 되먹임변수들을 갖는 다관절의 더 복잡한 매니플레이터의 제어에 있어서는 고차원 입력벡터로 정의되는 단 하나의 대규모 CMAC으로는 실제적으로 불가능하다.

다층 CMAC 시스템은 많은 입력변수로 정의되는 고차원 CMAC 문제의 과도한 소요 기억량이나 훈련의 어려움을 극복하기 위해 복잡한 제어 문제를 계층 개념으로 분할하여 분할된 여러 저차원의 문제에 대한 저차원의 CMAC들이 층별로 구성된 시스템을 말한다. [11]에서는 Organization, Coordination, Hardware Control 수준의 개념으로 계층을 분할하였다. Fig. 1 [13]은 분할된 제어 문제에 대해서 CMAC이 층별로 조직된 제어 구조를 보여준다. 이 그림에서 각 CMAC은 위층의 입력에 대해서 아래층으로 일련의 출력을 내보낸다. 이 출력 신호들은 위층의 입력 지령 뿐 아

나라 되먹임변수들의 함수들이다. 예를 들어, 다관절의 복잡한 매니플레이터에 대해서는 최상위의 CMAC은 기본적인 운동 지령을 하달하고, 중간층의 CMAC은 그 지령에 응해서 x, y, z 속도 성분들을 생산하며, 최하위의 CMAC들이 실제로 액츄에이터를 제어하도록 문제를 분할하는 것이 최선의 방법이 될 것이다.

그러나 고차원의 CMAC 문제에 대해서 다층 CMAC 시스템을 구성하기 위한 일반적인 다층화 알고리즘은 없을 뿐더러, 비록 어떤 문제에 대한 제어 함수를 해석적인 함수로 표현한다고 하더라도 제어 문제의 분할은 직접적인 수학적 과정은 아니다. 예를 들어, 고차의 제어 함수가 CMAC 정의역에서 해석적(analytic)이라면 무한 급수를 사용하여 제어 함수를 전개하는 것이 가능하다. 그러나, 제어함수의 도함수들을 구하여야 하는 어려움 때문에 고차제어 함수의 무한 급수 전개는 실제적이지 못하다.

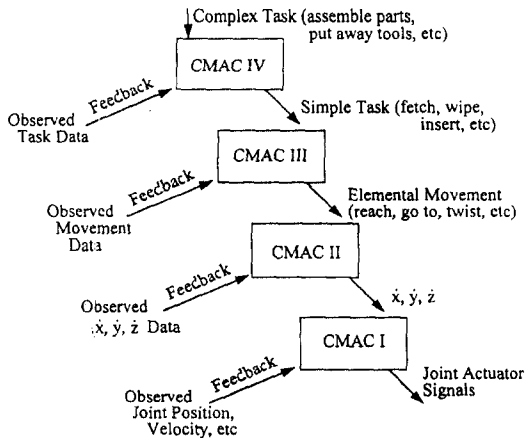


Fig. 1 A hierarchical structure of CMAC controllers[13].

반면, 제어 문제의 성격을 분석함으로써 고차원 제어 문제에 대해서 다층 CMAC 시스템을 구성할 수 있는지를 탐색할 수 있다. 예를 들면, 최상위의 지령이 얼마나 복잡하고, 각 계층에서의 되먹임 변수들이 몇개나 있고, 어떤 유형이며, 어떤 훈련 방법을 사용할 수 있을 것인지[13] 등에 따라 분석하면 될 것이다. 또한 어떤 층의 저차원 제어 함수가 더 단순한 제어 함수들의 결합으로 구성될 수 있다면 그 층에서 또

다른 제어 문제의 분할이 가능할 것이다.

고차원 CMAC 문제를 계층 관계에 있는 여러개의 저차원 CMAC 문제로 분할하면 무엇보다도 고차원 기억표로 인한 과도한 소요 기억량의 부담을 덜 수 있다. 이것은 CMAC 기억표의 차원은 입력변수의 개수에 의해 정해지므로, 규모가 큰 고차원 기억표 하나를 갖는 원래 문제가 규모가 작은 저차원의 기억표 여럿을 갖는 분할된 문제로 되기 때문이다.

다층 CMAC 시스템을 구성하면 시스템내의 모든 CMAC을 훈련해야 한다. 먼저, 다층 CMAC 시스템내의 모든 CMAC을 훈련하기 위해 원래 문제의 입출력 관계에 의한 전체적인 변환을 고려해 보자. 그러면, 각 훈련 주기마다 시스템 전체의 입력변수와 출력값의 관계에 따라 시스템의 모든 CMAC들이 학습해야 하므로 한 CMAC의 훈련에 이웃 CMAC들이 관련된다. 그런데, 원래 문제에 의한 입력변수와 출력값 관계만으로는 시스템내의 모든 CMAC을 훈련할 수 없다. 시스템내의 CMAC들을 원래의 문제를 계층별로 분할한 저차원 문제들의 입력변수들과 출력값의 관계로 정의하였기 때문이다. 그래서, 원래 문제에 의한 전체 입력변수와 시스템의 출력값 사이의 관계 외에 각 CMAC에 대한 입력변수와 출력값 사이의 관계가 설정되어 있어서 모든 훈련 주기에 학습기준으로 이용할 수 있어야 한다. 그러므로, 시스템의 모든 CMAC들을 각 훈련 주기마다 동시에 훈련한다는 것은 다층 CMAC 시스템의 장점을 살리는 훈련 방법이라고 할 수 없다.

전체적인 입출력 관계에 의한 훈련 대신, 하나 하나의 CMAC을 별개로 훈련하고 모든 CMAC의 개별 학습이 끝난 뒤 전체적인 입출력 관계를 고려하는 훈련방법에 대해서 생각해보자. 원래의 문제를 계층별로 분할한 저차원 문제에 대한 입력변수와 출력값 사이의 관계에 따라 각 CMAC을 훈련한다. 필요한 경우 이웃 계층간의 CMAC이 서로 학습에 연루될 수도 있을 것이다. 이 방법에 의하면, 훈련은 최상층에서 시작하여 첫 층의 CMAC이 학습을 끝내면, 그 CMAC의 출력값을 측정하고 그 범위를 확인하여 그 출력값을 다음 층 CMAC의 한 입력변수로서 사용할 수 있을 것이다. 비록 이 방법이 각 계층 CMAC의 학습으로 인해 훈련 시간이 길어질 수는 있으나 신뢰할 수 있는 훈련을 달성할 수 있을 것이다.

III. 문제의 분할과 소요 기억량 감소

문제의 분할로 인한 소요 기억량의 감소를 알아보기 위해, 3차원 운동이 가능한 어떤 매니플레이터에 대해서 3차원 공간에서 한 제어 주기 동안의 직선 운동을 생각해 보자. Fig. 2에서 직교좌표 (Cx, Cy, Cz)로 표시되는 점 C와 좌표 (Px, Py, Pz)인 점 P가 현재의 매니플레이터의 손과 궤적점의 좌표를 나타낸다고 하자. 한 폐회로 제어 주기에서 매니플레이터의 손을 어떤 점 C에서 점 P로 이동하기 위한 지령을 해당 제어 계층(예, Fig. 1의 CMACi)에서 "(Cx, Cy, Cz)에서 (Px, Py, Pz)로 이동"이라고 할 때, 점 P를 (Cx, Cy, Cz)와 점 P의 C에 대한 상대 위치 ($\Delta x, \Delta y, \Delta z$)로 정의하자. 그러면, 이 여섯 변수들은 한 제어 주기 동안의 매니플레이터 손의 위치에 대한 역기구학 문제를 정의하는 충분한 정보이고, Fig. 3에서처럼, 6차원 CMAC를 정의한다.

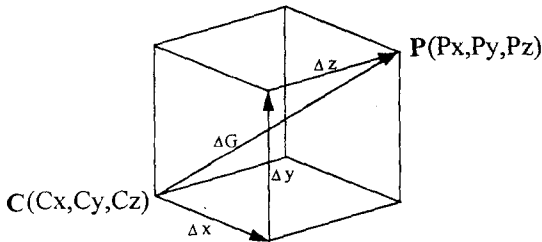


Fig. 2 Decomposition of vector ΔG using three unit vectors.

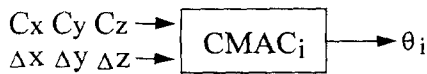


Fig. 3 Scheme of a 6-D CMAC with 6 input variables.

Fig. 3에서, Cx, Cy, Cz, $\Delta x, \Delta y, \Delta z$ 로 정의한 6차원 CMACi는 관절 i에 대한 역기구학 해를 학습하여 6차원 기억표에 저장할 것이다. 기억표의 규모는 여섯 입력변수들의 해상도와 K[i]의 선택에 따라 정해진다. 이 매니플레이터의 작업공간의 크기는 임의의 단위로 (352, 200, 200)이다. 한 제어 주기 동안 이동 가능한 거리를 고려하여 한 점에 대해서 어떤 궤적점을 상대적으로 정의할 수 있는 공간의 크기는 (16, 16, 16)으로 정한다. 입력변수들의 해상도는 1로, K는 8로 하자. 이렇게 정의한 CMAC에 대한 기억표의 계산은 Table 1에 나와있다. Table 1에서 Si는 입력 변수, Range는 입력 변수 값의 범위, ri는 i번째 입력 변수의 불연속화 해상도, iNj는 입력 변수 Si의 j번째 중간 변수 축이 불연속화에 의해 분할된 마디[5] 수이다. Table 1에 의하면, 각 중간변수층에 대해 821, 340개의 기억 장소를 갖는 6차원 기억표가 있어야 한다. K=8이므로, 그런 기억표가 여덟개 있다. 그러면, 한 관절 각도 값에 대해서 821,340 * 8 = 6,570,720개의 기억 장소가 필요하고, 여섯 관절 매니플레이터라면, 여섯 관절에 대해서는 6 * 6,570,720 = 39,424,320개의 기억 장소가 필요하다.

이제, 6차원 문제로 인한 소요 기억량을 줄이기 위해, 점 C에서 점 P로 이동하는 제어 문제를 분할한다. 공간에서의 운동을 x, y, z 방향으로 세 개의 연속하는 직교 단위 방향 운동으로 분할하는 것은, 운동에 대한 변위 벡터를 직교좌표의 단위 방향으로 분해하는 것이다. Fig. 2에서 C는 손의 현재 위치를, P는 경로상의 한 궤적점의 위치를 나타낸다. 벡터 ΔG 는 세 단위 벡터로 분해할 수 있는데, Fig. 2와 같이하면, 벡터 Δx , 벡터 Δy 및 벡터 Δz 합으로 나타낼 수 있다. 각 단위 방향 운동은 네개의 성분을 갖는 벡터로 정의할 수 있는데, x 방향으로 (Cx, Cy, Cz, Δx), y 방

Table 1. Required memory locations for a 6-D CMAC.

Si	Range	ri	iN1	iN2	iN3	iN4	iN5	iN6	iN7	iN8
S1	[0, 352]	1	45	45	45	45	45	45	45	45
S2	[0, 200]	1	26	26	26	26	26	26	26	26
S3	[0, 200]	1	26	26	26	26	26	26	26	26
S4	[-8, 8]	1	3	3	3	3	3	3	3	3
S5	[-8, 8]	1	3	3	3	3	3	3	3	3
S6	[-8, 8]	1	3	3	3	3	3	3	3	3
			821,340	821,340	821,340	821,340	821,340	821,340	821,340	821,340

향으로는 (Px, Cy, Cz, Δy), z 방향으로는 (Px, Py, Cz, Δz)로 정의한다. 그러면, Fig. 4의 배열처럼, 매니플레이터 기준좌표계의 단위 방향 순서에 따라, 훈련을 마친 CMAC_ix는 매니플레이터 손을 (Cx, Cy, Cz)에서 (Cx', Cy', Cz')로 CMAC_iy는 (Cx', Cy', Cz')에서 (Cx'', Cy'', Cz'')로, CMAC_iz는 (Cx'', Cy'', Cz'')에서 (Px, Py, Pz)로 운동하는데 필요한 관절 i에 대한 역기구학 문제의 해를 기억하고 있을 것이다. Fig. 4는 Fig. 1과 비슷한 구조로, Fig. 1의 CMAC I이 담당하는 문제가 다시 분할된 것으로 생각할 수 있다.

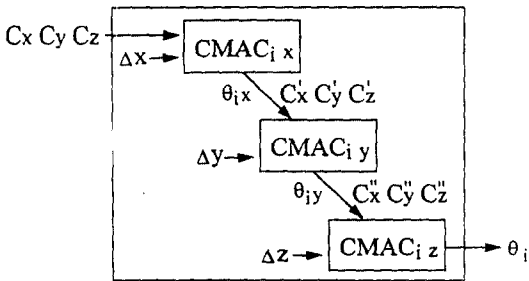


Fig. 4 Three-layered CMAC system for the problem of CMAC of Fig. 3.

분할 운동에 의한 CMAC 시스템의 소요 기억량을 계산해 보자. 3차원 운동을 단위 방향 운동으로 분할하면, 작업공간 (352, 200, 200)에 대해서 한 단위 방향으로의 이동거리가 양과 음의 방향으로 각각 8이므로, 한 점에 대해서 어떤 격자점을 상대적으로 정의할 수 있는 공간은 직선이 되고, 직선의 길이는 16이 된다. 입력변수들의 해상도를 각각 1로, K는 8로 하자. Table 2에 소요되는 기억량이 계산되어 있다. Table 2에 의하면, 한 중간변수층에 대해 91,260개의 기억 장소를 갖는 4차원 기억표가 있어야 한다. K=8이므로, 한 단위 방향 운동을 위해 그런 기억표가 여

덟개 있어야 하고, 세 단위 방향이므로, $91,260 \cdot 8 \cdot 3 = 2,190,240$ 개의 기억 장소가, 한 관절의 각도 값에 대해 소요된다. 여섯 관절에 대해서는 $6 \cdot 2,190,240 = 13,141,440$ 개의 기억 장소가 필요하다.

Table 1과 Table 2에 의한 소요 기억량을 비교해 보면, 6차원 CMAC에 대해서는, 소요 기억량이 39,424,320이고, 분할 운동에 의한 4차원 CMAC들로 구성된 시스템에 대해서는 13,141,440 있으면 된다. CMAC 시스템에 의한 소요 기억량은 6차원 CMAC에서 소요되는 기억량의 1/3이다. 이와 같이, 3차원 공간에서의 운동을 정의하는 벡터를 분해하여, 3차원 운동과 관련한 CMAC의 입력변수의 개수를 줄임으로서, 기억 소요량을 대폭 감축할 수 있다. 소요 기억량 감소 외의, 다층 CMAC 시스템의 훈련과 성능에 관련된 문제는 다음에 논의하기로 한다.

IV. 다층 CMAC 시스템의 훈련

문제의 분할로 인한 CMAC 시스템의 훈련 문제를 논의하기 위하여, 3 자유도를 갖는 평면 매니플레이터의 2차원 운동을 위한 역기구학 문제를 고려해 보자. 사실, DOS용 Fortran 프로그램으로, III절에서 가정한, 3차원 운동이 가능한 6 관절 매니플레이터의 역기구학 문제에 대한 CMAC 시스템을 전산모사하기가 불가능하다. 비록 2차원 운동의 평면 매니플레이터라고는 하지만, 3차원 운동의 분할로 인한 훈련 문제를 2차원 운동의 분할로 인한 훈련 문제에서 그대로 관찰할 수 있기 때문에, 다층 CMAC 시스템의 훈련 문제를 고려하는 데는 별 문제가 없다. 그러나, 단지 2차원 운동을 단위 방향의 운동으로 분할하면, 3차원 운동의 분할에서와 같은 소요 기억량의 감소는 기대할 수 없음에 유의할 필요는 있다.

3 자유도 평면형 매니플레이터는 일반적인 평면 운

Table 2. Required memory locations for 4-D CMACs.

Si	Range	ri	iN1	iN2	iN3	iN4	iN5	iN6	iN7	iN8
S1	[0, 352)	1	45	45	45	45	45	45	45	45
S2	[0, 200)	1	26	26	26	26	26	26	26	26
S3	[0, 200)	1	26	26	26	26	26	26	26	26
S4	[-8, 8)	1	3	3	3	3	3	3	3	3
			91,260	91,260	91,260	91,260	91,260	91,260	91,260	91,260

동을 할 수 있는 가장 간단한 기구로서, 세 관절각도 값이 주어지면 평면상의 손의 위치가 결정되며, 끝 관절각도가 마지막 링크의 방위각이 된다. 평면형 매니플레이터의 실제 작업 면적은 링크 길이와 관절의 운동 범위에 달려 있다. 따라서, 평면형 매니플레이터가 작업 면적 내의 장애물을 피할 수 있는 등의 작업을 능숙하게 수행할 수 있게 하기 위해서는, 경우에 따라서 3 자유도 이상이 요구되기도 한다. 중량물 취급을 위한 동력매니플레이터[16]의 마지막 세 링크를 한 매니플레이터로 간주하여 예를 들어보자. 이 매니플레이터의 재원으로, 첫번째 링크의 길이는 400mm, 둘째는 300mm, 셋째는 475mm이고, 손의 길이는 125mm이다. 각 관절의 운동 범위는, 첫째 관절이 270도, 둘째 관절도 270도, 셋째 관절은 연속적인 원운동이 가능하다.

이 매니플레이터에 대한 역기구학 문제는 다음과 같다: 매니플레이터가 어떤 자세로 있을 때, 그 손끝을 다른 곳으로 옮기려고 한다. 이때 그곳의 위치만 주고 마지막 링크의 방위는 주지 않는 대신, 매니플레이터의 출발점에서의 자세를 가능한대로 유지하면서 목표점으로 운동하도록 하는 정성적 제한 조건을 준다. 마지막 링크의 방위와 매니플레이터 자세에 대한 제한 조건을 만족하면서, 손이 원하는 곳에 위치할 수 있도록 세 관절 각도 값을 정하는 것이 문제이다. 문제를 푸는 방법은 여러가지가 있을 수 있지만, 여기서는 기하학적인 방법을 쓰는데, 이 방법에 대해서는 [17]에 자세히 설명되어 있다. Fig. 5는 이 방법에 의해서, 한 목표점을 향해 직선 경로로 움직일 때의 관절각도들의 변화를 보여준다. 각도 값의 변화가 연속적이고 순하게 변하므로, 이 기하학적인 방법이 CMAC이 담당하기에 적절한 함수를 만들어 낼 수 있고, 그래서 기하학적인 방법이 이 문제에 대해서 CMAC을 위한 적절한 해법임을 알 수 있다. 물론, 해석적이거나 기하학적 해법이 있는 문제를 굳이 CMAC으로 모사할 필요는 없지만, Random Problem[18]의 성격을 갖는 CMAC 문제에 대해서도, 여기서 논의하는 방법을 그대로 적용할 수 있을 것이다.

Fig. 6에서, $C(C_x, C_y)$ 와 $P(P_x, P_y)$ 는 현재의 손과 궤적점의 위치를 나타낸다. C에서 P로의 2차원 운동을 3차원에서와 같은 방법으로 분할하면, x와 y 방향으로 두개의 연속되는 직교 단위 방향 운동으로 분할

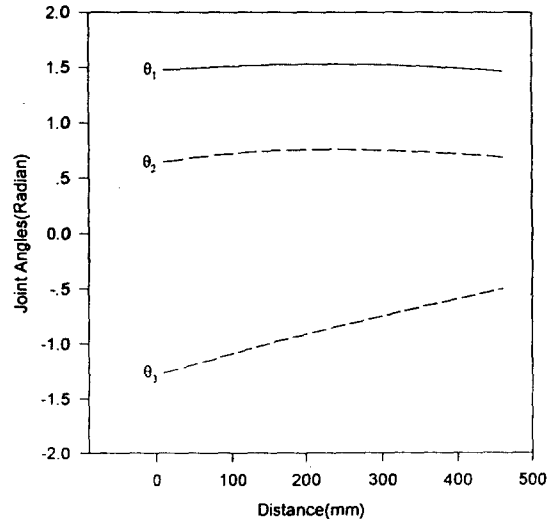


Fig. 5 Joint angle changes by the geometrical solution method when the hand moves along a straight line path.

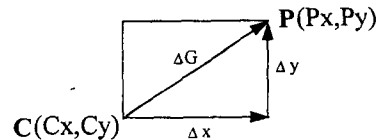


Fig. 6 Decomposition of vector ΔG using two unit vectors.

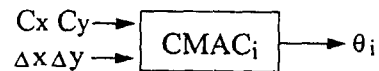


Fig. 7 Scheme of a 4-D CMAC with 4 input variables.

수 있다. 이 그림에서 벡터 ΔG 는 두 단위 방향으로 분해할 수 있으므로, 벡터 Δx 와 벡터 Δy 의 합으로 나타낼 수 있다.

Fig. 7은 네개의 입력변수 $C_x, C_y, \Delta x, \Delta y$ 를 갖는 원래의 문제에 대한 CMAC의 입출력 관계를 보여주고, Fig. 8은 C에서 P로의 운동을 두 단위 방향으로 분할했을 경우의 CMAC 시스템을 보여준다. (C_x, C_y) 에서 x 방향으로 Δx 만큼 이동했을 때의 관절 i의 θ_{ix} 를 $CMAC_{ix}$ 에서 구한다. x 방향으로 Δx 만큼 이동한 새로운 위치 (C'_x, C'_y) 에서, y 방향으로 Δy 만

Table 3. Comparison of required memory locations by a 6-D CMAC with those by three 4-D CMACs system.

K	4	5	6	7	8
6-D	115,744,500	45,790,440	21,969,096	11,943,171	6,570,720
3:4-D	13,889,340	7,564,500	4,656,312	3,107,457	2,190,240
감축율	0.880	0.835	0.788	0.740	0.667

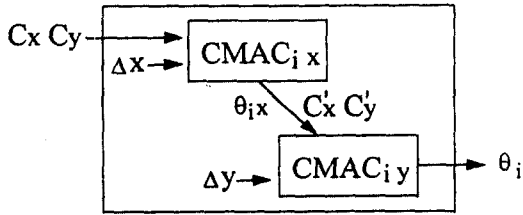


Fig. 8 Two-layered CMAC system for the problem of CMAC of Fig. 7

큼 이동했을 때의 관절 i 의 각도 값을 $CMAC_{iy}$ 에서 구하면, 결국, (P_x, P_y) 에 대한 CMAC 시스템의 출력 값 θ_i 가 된다. $CMAC_{ix}$ 는 이웃간훈련 방법[19]으로 입력변수 C_x, C_y 와 Δx 에 대해서 손이 (C_x, C_y) 에서 x 방향으로 Δx 운동할 때의 관절 i 의 각도 값을 학습하였다. 마찬가지로, $CMAC_{iy}$ 는 이웃간훈련 방법으로 입력변수 (C'_x, C'_y) 와 Δy 에 대해서 손이 (C'_x, C'_y) 에서 y 방향으로 Δy 운동할 때의 관절 i 의 각도 값을 학습하였다. 시스템의 두 CMAC은 작은 $\Delta x, \Delta y$ 값들에 대해서 작동한다는 제약 조건을 전제로, CMAC 매개 변수들을 설계하는데, 이것은 그런 제약조건 하에서 원래의 4차원의 문제를 분할하였기 때문이다.

V. 결과 및 토의

다층 CMAC 시스템의 주된 목적은 많은 수의 입력 변수로 인하여 입력벡터의 차원이 클 때 요구되는 기억량의 크기를 감축하기 위한 것이다. Table 3은 III 절에서 가정한 3차원 운동이 가능한 매니플레이터의 한 관절의 각도 값에 대해 요구되는 소요 기억량을 여러 K 값에 대해서 계산한 표이다. 6-D행은 보통 운동에 의한 여섯개의 입력변수로된 6차원 CMAC의 소요 기억량을, 3:4-D행은 분할 운동에 의한 네개의 입력변수로된 4차원 CMAC 세개의 소요 기억량을

합한 양을 나타낸다. 물론, 두 경우의 CMAC 매개 변수들은 $r_i=1, K=8$ 로 같을 때이다. 이 Table에서 소요 기억량이 감축되는 율은 K 값이 작아질수록 더 커지는 것을 알 수 있다.

이제, 다층 CMAC 시스템의 훈련과 그 성능에 대해서 고찰해보자. Fig. 8에서 $CMAC_{ix}$ 에 의해서 손이 P_x 에 정확히 이른다고는 볼 수 없고 C'_x 에 이르렀다고 하자. 이때, y 의 좌표도 C_y 가 아닌 C'_y 로 달라질 것이다. C'_x, C'_y 와 $\Delta y (=P_y - C'_y)$ 로 $CMAC_{iy}$ 에서 출력값을 받는다. 이런 출력 과정을 반복하여, C'_x 와 C'_y 가 P_x 와 P_y 에 가장 가까운 각도 값을 그 실행 주기의 출력값으로 한다. 이렇게 하면, 비록 매니플레이터의 손이 목표 지점에 상당한 정확도로 이른다 하더라도 분할 운동에 의한 매니플레이터의 자세는 보통 운동에 의한 것과는 다를 것이며, 목표 지점에 이르기 위해 출력과정을 많이 반복할수록 이런 현상은 더 심해질 것이다. 일반적으로, 한 제어 주기에 의한 이동 거리가 크면 보통 운동에 의한 관절 각도의 값들과 분할 운동에 의한 값들이 많이 다를 것으로 예상할 수 있다.

Fig. 9는 Fig. 4가 나타내는 합수(exact)를 Fig. 7의 4차원 CMAC이 이웃간훈련으로 학습한 결과(4-D)를 보여준다. 가로축은 한 출발점에서 어떤 목표점으로 이동한 거리를 나타내고, 세로축은 세 관절들의 각도 값을 나타낸다. Fig. 10은 Fig. 9에 의해 손이 평면상에서 지나간 경로를 보여준다. 이 그림에서 직선 경로를 따르는 궤적점들은 손이 목표점에 도달할 때까지, 각 제어 주기마다 경로를 따라 최대 8mm를 움직이도록 정하였다. Fig. 9와 Fig. 10을 보면, 비록 손이 평면의 직선 경로에 있는 모든 궤적점을 아주 가까이 지나가긴 했지만 운동하는 거리가 커질수록 CMAC의 학습모델이 나타내는 관절들의 각도 변화(exact)와 차이가 커짐을 알 수 있다. 이것은 한 궤적점에 대해서 손이 최고로 가까이 이를 때까지 CMAC이 계속

출력하도록 했기 때문인 것으로 생각되는데, 그동안 매니플레이터의 자세는 계속 달라졌을 것이다.

Fig. 11은 Fig. 4가 나타내는 함수를 Fig. 8의 3차원 CMAC 두개로 구성된 다층 CMAC 시스템이 학습한 결과를 보여준다. Fig. 12는 Fig. 10에 의해 손이 평면 상에서 지나간 경로를 보여준다. Fig. 12에 의한 경로가 Fig. 10에 의한 경로보다 직선에서 벗어나는 정도가 좀더 심함을 볼 수 있다. Fig. 9(4-D)와 Fig. 11(3-D)을 비교하면, 관절 각도의 변화가 거의 같음을 볼 수 있는데, 이것은 주어진 매니플레이터의 역기구학 문제에 대해서, 4차원 CMAC 한개에 의한 학습과 3차원 CMAC 두개에 의한 학습이 거의 같은 성능을 보여준다는 의미이다. 한 예에 불과하지만, 고차원 CMAC에 비교하여 다층 CMAC 시스템이 비록 제어 기로서의 성능이 약간 떨어진다 하더라도 소요 기억량이 굉장히 감소되는 사실은 특기할 만하다.

또한 다층 CMAC 시스템의 성능이 단 하나의 고차원 CMAC에 의한 성능에 비해 다소 떨어진다고 하더라도, 고차원 CMAC 하나에 주어진 소요 기억량을 기준으로 다층 시스템의 CMAC들을 정의하는 각 변수 축의 해상도를 더 낮게 하거나 더 작은 K 값을 사용하면, 단 하나의 고차원 CMAC에 의한 성능보다 더 나은 다층 CMAC 시스템을 구성할 수 있는 가능

성도 있을 것이다. 이런 가능성은 CMAC이 더 나은 해상도와 더 작은 K에 대해서 좋은 학습 성능을 갖는다는 일반적인 특성에 바탕을 둔 고찰이다. 예를 들어, Table 3에서 6-D행의 K=8에서 소요 기억량이 6,570,720이고, 3:4-D행의 K=5에서 7,564,500 K=6

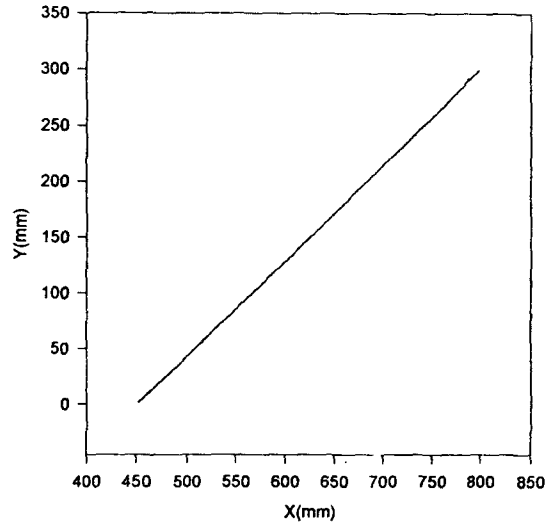


Fig. 10 Path along which the hand moves by a 4-D CMAC.

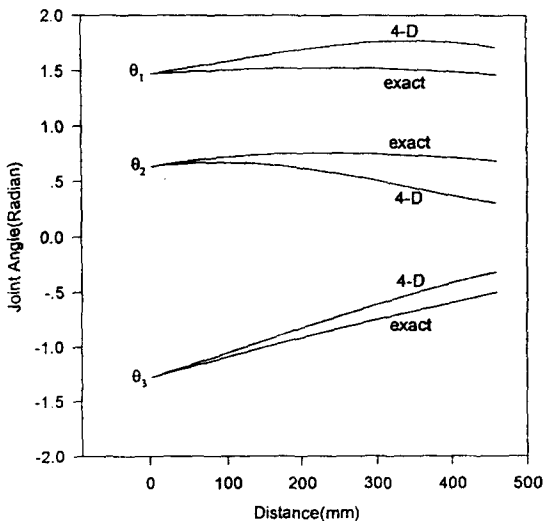


Fig. 9 Joint angle changes by a 4-D CMAC when the hand moves along a straight line path.

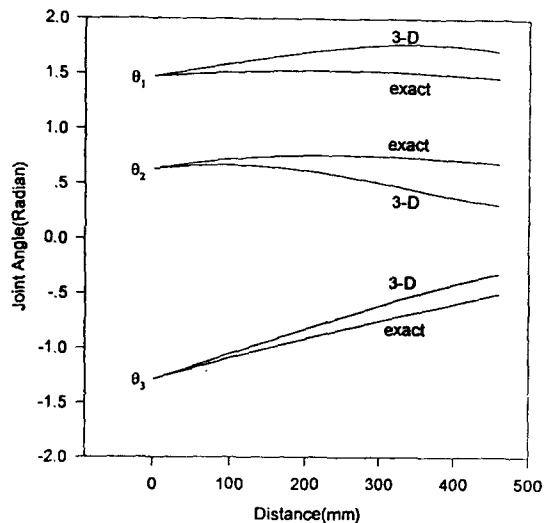


Fig. 11 Joint angle changes by two-layered CMAC system when the hand moves along a straight line path.

Table 4. Required memory locations by 3-D CMAC with lower resolutions

Si	Range	ri	iN1	iN2	iN3	iN4	iN5	iN6	iN7	iN8
S1	[0, 352)	0.75	59	60	60	60	60	60	59	59
S2	[0, 200)	0.75	34	35	35	34	34	34	34	34
S3	[0, 200)	0.75	34	35	35	34	34	34	34	34
S4	[-8, 8)	0.75	3	4	4	4	4	4	3	3
			204,612	294,000	294,000	277,440	277,440	277,440	204,612	204,612

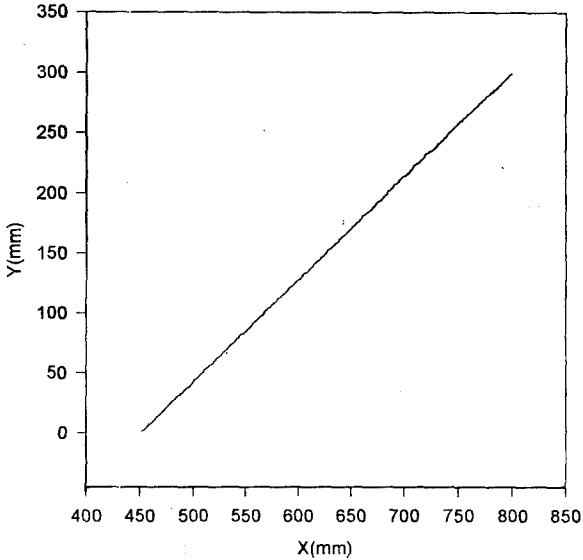


Fig. 12 Path along which the hand moves by two-layered CMAC system.

에서는 4,656,312이다. 이론적으로는 $K=8$ 로 된 6차원 CMAC에 주어지는 만큼의 기억 장소가 있다면 $K=6$ 으로된 4차원 CMAC 시스템을 구성할 수 있다. 이제, 6차원 CMAC에 주어지는 만큼의 기억 장소가 있을 때 K 는 8을 갖되 얼마나 작은 해상도를 갖는 4차원 CMAC 시스템을 구성할 수 있는지 보자. Table 2에서 $r_i=1$ 이므로 $r_i=0.75$ 로 하면, 한 단위 방향에 대해서 한 관절 각도 값을 위해 Table 4에 의해, $612 \cdot 3 + 294,000 \cdot 2 + 277,440 \cdot 3 = 2,034,156$ 개의 기억 장소가 소요된다. 세 단위 방향이 있어야 하므로, 모두 6,102,468개의 기억 장소가 소요된다. 이 규모는 Table 1에 의한 6,570,720보다 조금 적다.

또한 다층 시스템의 CMAC들이 원래 문제에 의한 고차원 제어 함수와 비교할 때 더 단순한 함수들을 담당할 것이므로 그 단순한 함수들의 함수 값의 변화

정도가 더 순하여 CMAC이 제어 함수를 더 잘 일반화할 수 있다면, 시스템의 성능이 더 나아질 것으로 기대할 수도 있을 것이다.

VI. 결론 및 전망

입력변수의 개수가 많은 고차원 CMAC 문제에 대해서 저차원 CMAC들로 다층 CMAC 시스템을 구성하여 고차원 CMAC으로 인한 과도한 소요 기억량을 감축할 수 있음을 관찰하였다. 3 차원 직교좌표 공간에서의 운동을 서로 직교하는 세 단위 방향 운동으로 분할하는 예를 들어, 분할된 운동에 대한 다층 시스템을 구성하였다. 또한, 평면 매니퓰레이터의 역기구학 문제에 대한 전산 모사를 통해, 다층 CMAC 시스템의 훈련과 성능을 소요 기억량의 관점에서 고찰하였다.

제어 문제의 분할과 그에 따른 입력벡터의 분할로 다층 CMAC 시스템을 구성하고 원하는 제어함수를 훈련할 수 있다면, 다층 CMAC 시스템으로 CMAC 기억표의 크기를 줄일 수 있는 것은 분명하다. 이 논문에서 예를 든 변위벡터의 분해로 인한 운동 문제의 분할 기법은 직교좌표, 원통형, 또는 구형좌표계들로 표시되는 위치 정보와 연관되고 그 함수 값이 연속적으로 부드럽게 변하는 제어 함수의 특성을 갖는 다른 제어 문제에도 응용할 수 있을 것이다. 지금까지 CMAC을 이용한 연구 분야를 보면, Fig. 1과 같이 구성된 대규모 문제의 한 층에서 CMAC이 제어기로서 역할 한 예가 많은데, 이 논문에서 논의한 기법을 Fig. 1의 CMAC I에 적용할 수 있을 것이다.

다층 CMAC 시스템을 구성하는 개별 CMAC의 학습 모델은 다층화의 바탕이 된 문제의 분할에 근거하여야 하고, 그에 따른 적절한 훈련 방법을 마련하여야 할 것이다. 시스템의 성능 조율을 위한 훈련에 대

해서는 훈련 시간이 상당히 걸릴 것이므로 기존의 훈련 방법 외에 문제의 특성에 맞는 적절한 훈련 방법을 개발하여야 할 것으로 생각된다.

참 고 문 헌

1. J. S. Albus, "A New Approach to Manipulator Control; The Cerebellar Model Articulation Controller(CMAC)," *Journal of Dynamic Systems, Measurement, and Control, Transactions of the ASME, Series G*, Vol. 97, No. 3, pp. 220-227, September, 1975.
2. S. Manglevhedakar, *An Adaptive Hierarchical Model for Computer Vision*, MSME Thesis, Department of Mechanical Engineering, Louisiana State University, 1986.
3. 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.
4. D. Ellison, "On the Convergence of the Multidimensional Albus Perceptron," *The International Journal of Robotics Research*, Vol. 10, No. 4, pp. 338-357, August 1991.
5. M. Brown and C. Harris, *Neurofuzzy Adaptive Modelling and Control*, Prentice Hall, 1994.
6. S. Kwon, *An Adaptive Control System for Biological and Robotic Simulations*, Ph.D. Dissertation, Dept. of Mechanical Engineering, Louisiana State University, December 1990.
7. G. A. Larsen, S. Cetinkunt, and A. Donmez, "CMAC Neural Network Control for High Precision Motion Control in the Presence of Large Friction," *Transaction of the ASME Journal of Dynamics Systems, Measurement, and Control*, Vol. 117, pp. 415-420, September 1995.
8. M. H. Raibert and B. K. P. Horn, "Manipulator Control Using the Configuration Space Method," *The Industrial Robot*, Vol. 5, No. 2, pp. 69-73, June 1978.
9. J. Simons, H. V. Brussel, J. D. Schutter, and J. Verhaert, "A Self-Learning Automation with Variable Resolution for High Precision Assembly by Industrial Robots," *IEEE Transactions on Automatic Control*, Vol. AC-27, No. 5, pp. 1109-1113, October 1982.
10. M. H. Raibert and F. C. Wimberly, "Tabular Control of Balance in a Dynamic Legged System," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-7, No. 6, pp. 407-420, June 1984.
11. G. N. Saridis and H. E. Stephanou, "A Hierarchical Approach to the Control of a Prosthetic Arm," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-7, No. 6, pp. 407-420, June 1977.
12. R. D. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, pp. 14-23, March 1986.
13. J. S. Albus, "Data Storage in the Cerebellar Model Articulation Controller(CMAC)," *Journal of Dynamic Systems, Measurement, and Control, Transactions of the ASME, Series G*, Vol. 97, No. 3, pp. 228-233, September 1975.
14. J. S. Albus, H. G. McGain, and R. Lumina, "NASA/NBS Reference Model for Telerobot Control System Architecture(NASEREM)," *National Bureau of Standards Technical Notes 1235*, National Bureau of Standards, Washington, DC, July 1987.
15. W. T. Miller III, "Real-Time Neural Network Control of a Biped Walking Robot," *IEEE Control Systems*, pp. 41-48, February 1994.
16. *Description and Operating Instructions for Power Manipulator A1055*, BB-K12038-8, HWM Hans Walischmiller GmbH, MBHh.
17. Y. S. Park, W. T. Chung, S. Kwon, J. S. Yoon, J. Lee, and H. S. Cho, "Neural Network Application to Camera-Robot Kinematic Conversion," *Proceedings of '93 International Conference on Advanced Robotics ('93 ICAR)*, pp. 55-60, Tokyo,

- Japan, November 1-2, 1993.
18. Y. S. Abu-Mostafa and D. Psaltis, "Optical Neural Computers," *Scientific American*, Vol. 256, pp. 66-73, March 1987.
 19. 권성규, "CMAC을 위한 이웃간 訓練 方法," 대한 기계학회논문집 제16권 제10호, pp. 1816-1823, 1992.
 20. J. Lee and B. M. Kramer, "Analysis of Machine Degradation Using a Neural Network Based Pattern Discrimination Model," *Journal of Manufacturing Systems*, Vol. 12, No. 5, pp. 379-387, 1993.
 21. H. Shiraishi, S. L. Ipri, and D. Cho, "CMAC Neural Network Controller for Fuel-Injection Systems," *IEEE Transactions on Control Systems Technology*, Vol. 3, No. 1, pp. 32-38, March 1995.



권 성 규(Sunggyu Kwon)정회원

1955년 3월 29일생

1980년:연세대학교 기계공학과
졸업(학사)

1983년:연세대학교 대학원 기계
공학과 졸업(석사)

1990년:미국 루이지애나주립대학
교 기계공학과 졸업(Ph.D.)

1991년:한국원자력연구소 원격장치기술실 선임연구원

1995년:계명대학교 기계설계학과

※관심분야:CMAC 응용 및 지능제어, CAD, 로봇
시스템 개발