

로봇 머니플레이터의 동력학 제어를 위한 AMN 제어기

AMN Controller for Dynamic Control of Robot Manipulators

정재욱, 국태용, 이택중

성균관대학교 전기 전자 컴퓨터공학부(Tel:+82-331-290-7202; Fax:-82-331-290-7191; E-mail:skechy@yurim.skku.ac.kr)

Abstracts In this paper, we present an associative memory network (AMN) controller for dynamic robot control. The purpose of using AMN is to reduce the size of required memory in storing and recalling large of data representing input relationship of nonlinear functions. With the capability, AMN can be used to dynamic robot control, which has nonlinear properties inherently. The proposed AMN control scheme has advantages for the inverse dynamics learning, no limitation of input range, and insensitive of payload change. Computer simulations show the effectiveness and feasibility of proposed scheme.

Keywords Robotics, Manipulators, Learning Control, AMN, FEL

1. 서론

80년대 중반이후 널리 소개되기 시작한 인공 경회로망(artificial neural networks) 이론은 여러가지 비선형 입출력 관계를 학습 또는 훈련을 통하여 근사화 시킬 수 있다는 장점으로 인하여 입출력 사이에 비선형 관계를 가지는 복잡한 제어 시스템, 특히 로봇 머니플레이터 제어에 적용하려는 시도가 이루어져 왔다. 이에 대표적인 것으로 [1, 2]의 CMAC(cerebellar model arithmetic computer)이 있다. CMAC은 정규화된 입력을 복잡한 사상규칙을 통하여 특정 메모리를 지정한 뒤 사상된 메모리의 내용을 합산하여 출력을 만드는 구조로, 소위 LUT(look-up table) 기법을 취하고 있다. CMAC은 정적인 시스템(패턴, 음성, 영상인식분야)에 널리 이용되는 오차 역전과 학습법에 의한 다층 퍼셉트론(perceptron) [3]에 비해 동적 특성이 강한 로봇 제어에 적용이 용이하다는 것과, 학습규칙의 실시간 계산이 가능할 뿐만 아니라 LMS(least mean square) 기법을 근간으로 한 학습규칙의 구조상 지역 최소점(local minima)으로의 수렴이 적다는 장점을 가진다. 그러나 복잡한 사상규칙은 직관적인 이해를 어렵게 만들 뿐만 아니라 일반화 인자, 양자화 레벨과 같은 설계 파라미터들은 일의적인 결정법이 없다는 단점이 존재한다.

이 논문에서는 로봇의 기구학 제어 [3, 7]에 사용했던 연관 사상 소자를 부분적으로 수정한 AMN(associative memory networks)을 로봇의 동력학 제어에 적용하고자 한다. 제안한 AMN 제어기의 학습은 CMAC과 유사한 LMS 기법을 사용하므로써 수렴속도나 실시간 계산 측면에서 장점을 갖도록 하였으며, CMAC과는 다른 입력사상 규칙 (AMR: associative mapping rule)을 사용하므로써 우선 직관적으로 이해하기 쉬우며, 실제 구현시에도 용이한 구조를 가진다. 아울러 입력에 대해 별도의 정규화 과정을 취하지 않으므로 그 범위의 제한이 없으며, 단지 제어 시스템의 정밀도에 따라 이산레벨의 범위가 정해질 뿐이다.

다음 절에서는 제안한 AMN 제어기의 동작과 학습규칙 그리고 비선형 함수에 대한 근사화 능력에 대해 기술할 것이며, 3절에서는 로봇 머니플레이터의 동력학 제어에 대한 AMN의 적용 가능성에 대해 논의하며, 4절에서 제안한 기법의 성능검증을 컴퓨터 시뮬레이션을 통하여 보이며, 마지막으로 결론을 맺도록 한다.

2. AMN의 구조 및 동작원리

2.1 연관사상규칙(AMR: Associative Mapping Rule)

AMN은 종래의 메모리가 가지는 번지 지정방식과는 달리 새로운 연관사상 규칙을 사용하므로써 메모리의 양을 줄이고자 제안한

것으로서, 예를 들면, 종래의 메모리 구조에서는 특정 메모리 장소를 사상하기 위해서는 시계열(time series)에 따른 순서에 입각하여 메모리에 접근하는 방법이지만 AMN은 입력의 시계열이 아닌 그 내용에 의하여(content-addressing) 메모리를 접근하는 방법이다. 우선 n 개의 입력변수 $x_i(t) \in R$ ($1 \leq i \leq n$)와 출력변수 $y(t) \in R$ 가 시공간 $t \in [0, t_f]$ 에서 정의될때, n 은 입력변수의 갯수(5)가 된다. 그리고 입출력 변수는 $1 \leq k \leq n_f$ 의 구간에서 이산 샘플열로 주어지는 데이터라 가정한다. 그러면 입력변수는 M 진수계, L 자리의 부동소수점으로 다음과 같이 나타낼 수 있다.

$$\begin{aligned} x_1(k) &= s_0^k \bar{a}_1^k \bar{a}_2^k \cdots \bar{a}_l^k, \\ x_2(k) &= s_0^k \bar{b}_1^k \bar{b}_2^k \cdots \bar{b}_l^k, \\ &\vdots \\ x_n(k) &= s_0^k \bar{c}_1^k \bar{c}_2^k \cdots \bar{c}_l^k. \end{aligned}$$

여기서 $\{s_0^k, s_1^k, \dots, s_l^k\}$ 는 $\{x_1(k), x_2(k), \dots, x_n(k)\}$ 의 부호를 나타내며 소수점은 생략한 형태로 표시하였다. 위와 같이 표현된 입력변수를 AMN에 적용하기 위해서 양의 값을 갖도록 천이시켜서 해당 메모리에 addressing할 수 있는 번지로 표현한다. 즉, $1 \leq i \leq L$ 의 경우

$$\begin{aligned} a_i^k &= s_0^k \bar{a}_i^k - Base, \\ b_i^k &= s_0^k \bar{b}_i^k - Base, \\ &\vdots \\ c_i^k &= s_0^k \bar{c}_i^k - Base. \end{aligned} \quad (1)$$

여기서

$$Base = \begin{cases} M-1 & \text{for } s_0^k < 0 \\ M & \text{for } s_0^k \geq 0 \end{cases}$$

식(1)과 같이 표현되는 입력변수의 값으로 지정되는 메모리 사상과 이를 통한 출력은 다음 식으로 기술할 수 있다.

$$\begin{aligned} f(\cdot) &= [f_1(\cdot), f_2(\cdot), \dots, f_l(\cdot)]^T, \\ w_i(k) &= f_i(a_i^k, b_i^k, \dots, c_i^k), \\ y(k) &= \sum_{i=1}^L w_i(k). \end{aligned} \quad (2)$$

여기서 $f(\cdot)$ 는 addressing결과로 나타난 번지와 메모리의 사상관계를 나타내며, w 는 해당 메모리의 값(가중치) 또는 내용을 가리킨다. 출력 y 는 입력번지에 의해 사상된 모든 가중치의 합으로 계산된다.

그림(1)은 두 변수에 대한 입력사상과 메모리의 구조를 표현한 것으로 두 변수의 경우 하나의 AMN 블록에 총 $(2M)^2$ 의 메모리 장소가 존재하며, 인접한 데이터는 그렇지 못한 데이터에 비해 많은 메모리 장소를 공유할 수 있음을 알 수 있다. 즉 이러한 사상특성을 종래의 메모리 지정기법에 비해 많은 양의 메모리를 절약할 수 있는 장점을 갖게 된다.

2.2 학습규칙: WUR(weight update rule)

AMN의 가중치 수정규칙 또는 학습규칙은 우선 계산량의 축소를 도모하기 위해 같은 메모리 장소로 사상되는 인접한 두 데이터에 대해 중첩량을 미리 파악하여 중첩된 부분은 기존의 메모리 내용으로 출력을 만드는데 참여하고 그렇지 못한 부분만 수정해 가는 방법을 취한다. 즉, $\{x_1(k), x_2(k), \dots, x_n(k)\}$ 가 $\{x_1(k-1), x_2(k-1), \dots, x_n(k-1)\}$ 의 다음에 나타난 입력이라면 이들간에 많은 유효자리가 서로 같은 값을 가지게 된다. 그리하여 많은 경우 다음과 같은 영이 아닌 수 $\{m_a^k, m_b^k, \dots, m_c^k\}$ 가 존재한다.

$$\begin{cases} a_l^k = a_l^{k-1}, & 1 \leq l \leq m_a^k \\ b_l^k = b_l^{k-1}, & 1 \leq l \leq m_b^k \\ \vdots \\ c_l^k = c_l^{k-1}, & 1 \leq l \leq m_c^k. \end{cases}$$

모든 입력변수에 대한 중첩자리를

$$m_d^k = \min \{m_a^k, m_b^k, \dots, m_c^k\} \quad (3)$$

으로 정의하고, $k = 1$ 인 경우는 $m_d^k = 0$ 으로 하여 가중치 수정신호 $\epsilon(k)$ 로 가중치를 수정해 나간다. 이러한 관계를 정리하면, $m_d^k = 0$ 또는 $m_d^k = L$ 인 경우,

$$w_l^{j-1}(k) = w_l^j(k) + \eta_l \epsilon^j(k) \quad \text{for } 1 \leq l \leq L, \quad (4)$$

여기서 $\sum_{l=1}^L \eta_l = 1$ 이다. $m_d^k > 0$ 인 경우,

$$w_l^{j-1}(k) = \begin{cases} w_l^j(k), & (1 \leq l \leq m_d^k) \\ w_l^j(k-1) + \eta_2 \epsilon^j(k), & (m_d^k + 1 \leq l \leq L) \end{cases} \quad (5)$$

여기서 $\sum_{l=m_d^k+1}^L \eta_2 = 1$ 이고, 초기값은 $w_l^j(0) = 0$ 으로 두며, j 는 반복 학습횟수를 나타낸다. 위의 식에서 나타나는 분포 파라미터인 η_1 와 η_2 은 다음과 같이 정의한다.

$$\eta_1 \equiv \begin{cases} \frac{1}{L} & \text{(Equal Distribution)} \\ \frac{2(L-r-1)}{L(L-1)} & \text{(Reversely Weighted Distribution)} \end{cases}$$

$$\eta_2 \equiv \begin{cases} \frac{1}{L-m_d^k} & \text{(Equal Distribution)} \\ \frac{2(L-r-1)}{(L-m_d^k)(L-m_d^k-1)} & \text{(Reversely Weighted Distribution)} \end{cases}$$

이들은 입력 신호의 분포특성에 따라 균일 분포일때와 내림 가중 분포일때를 고려하여 적용하면 된다.

2.3 비선형 함수 근사화

전술한 AMN의 AMR과 WUR을 이용하여 AMN의 비선형 함수에 대한 근사화 능력을 고찰하고자 한다. 우선 근사화 시킬 목표함수를

$$y_d = e^{-\sin(2.0\pi x)}$$

로 정하고, 여기서 $x \in [0, 20]$ 에 속하는 값으로 정하였다. 이와 같이 원하는 출력신호를 미리 알 수 있는 지도학습(supervisor

learning)에서는 학습신호를 원하는 출력신호와 실제 출력과의 차로 사용될 수 있으므로 $\epsilon^j(k)$ 를

$$\epsilon^j(k) \equiv 3(y_d(k) - y^j(k))$$

로 사용하고, $y^j(k)$ 는 j 번째 학습횟수때의 k 번째 샘플열의 입력으로 나타난 AMN의 출력을 나타내고, β 는 학습속도 또는 수렴속도를 결정하는 양수로서 $0 < \beta \leq 1$ 의 값을 취한다.

그림(2)은 목표함수의 제곱을 도시한 것이고, 그림(3)은 열번째 학습후에 나타난 AMN의 출력 결과이다. 이때 $\beta = 0.1$ 로 정하고, addressing을 위해 입력의 이산레벨은 정수부분 한자리, 소수부분 다섯자리로 하여 모두 6자리의 입력 레벨, 즉 $L = 6$ 인 값을 사용하였다. 그러나 $L = 5$ 인 경우라도 만족할 만한 오차범위내로 수렴됨을 확인할 수 있었고, 이는 입력레벨을 다섯자리나 여섯자리 이상으로 정밀하게 나누어 addressing하지 않고서도 적절한 오차범위내에서 비선형 함수를 근사화 시킬 수 있음을 의미한다.

3. AMN의 로봇 동역학 제어

N 자유도를 가지는 로봇 매니퓰레이터의 동역학식은 다음과 같이 주어진다.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + f_{cr}(\dot{q}) = \tau \quad (6)$$

여기서 $q \in R^n$ 이며, 로봇 관절의 일반화 좌표벡터이다. 그리고 $M(q) \in R^{n \times n}$ 인 positive-definite 관성행렬이고, $C(q, \dot{q})\dot{q}$ 는 구심력 및 전향력(Coriolis force)을 나타내는 벡터이다. 또한 $g(q)$, $f_{cr}(\dot{q})$, τ 는 각각 중력과 마찰력, 입력토크의 벡터들이다.

위와 같은 로봇 동역학에 대해 널리 사용되는 제어 법칙은 PD(proportional-derivative)제어로 다음과 같이 표현할 수 있다 [6].

$$\tau = K_D(\dot{q}_d - \dot{q}) + K_P(q_d - q) \quad (7)$$

여기서 q_d 와 \dot{q}_d 는 로봇의 기준 궤적이고, K_D 와 K_P 는 negative feedback 시스템에서 필요한 양의 값을 가지는 이득 행렬이다. PD제어를 이용할 경우 closed-loop 오차 시스템은

$$\ddot{e} + M^{-1}K_D\dot{e} + M^{-1}K_P e = M(q)^{-1}(M(q)\ddot{q}_d + C(q, \dot{q})\dot{q} + g(q) + f_{cr}(\dot{q}))$$

가 된다. 여기서 $e \equiv q_d - q$, $\dot{e} \equiv \dot{q}_d - \dot{q}$, $\ddot{e} \equiv \ddot{q}_d - \ddot{q}$ 로 정의한다. 수식에서 알 수 있듯이 PD 제어를 사용한 오차 시스템이 안정하기 위해서는 첫 식의 우변이 0으로 수렴해야 하므로, 출력의 가속도를 측정가능하거나 이용할 수 있어야 한다는 단점이 있다. 이의 대안으로 로봇 동역학이 가지는 비선형성을 완전히 파악한 뒤 이를 보상하는 신호를 추가한다면 안정한 제어 특성을 얻을 수 있으나, 동적 시스템에 대한 비선형 관계를 엄밀한 수학적 모델로 표현한다는 것은 쉽지 않다. 그래서 전절에서 기술한 AMN의 비선형 함수 근사화 능력을 이용하여 수학적으로 풀기 어려운 로봇의 동적 비선형성을 AMN으로 보상하고자 한다. 즉,

$$\tau = K_D\dot{e} + K_P e + \tau_{inc} \quad (8)$$

여기서 τ_{inc} 는 AMN의 출력이고, 새로이 구성된 제어 입력을 로봇의 동역학 방정식에 대입하면,

$$\begin{aligned} & \ddot{e} + M^{-1}K_D\dot{e} + M^{-1}K_P e \\ & = M^{-1}(M(q)\ddot{q}_d + C(q, \dot{q})\dot{q} + g(q) + f_{cr}(\dot{q}) - \tau_{inc}) \end{aligned}$$

가 된다.

비선형 함수의 근사화 문제와 같은 지도학습의 경우 원하는 출력신호를 알 수 있기 때문에 AMN의 가중치들을 수정하기 위한 학습신호를 출력오차로 쉽게 적용할 수 있었지만, 식(8)에 사용되는 AMN의 출력은 로봇에서 필요한 제어 입력신호이므로 지도학습처럼 비교할 원하는 출력신호를 이용할 수 없다. 그리하여 이 논문에서는

원하는 출력을 모르는 자율학습법을 채용하며, [4]에서 제안한 FEL(feedback error learning) 기법을 이용한다. 그러면 AMN에 사용되는 학습신호는

$$e^j(k) = \beta(K_f)\dot{e} + K_p e \quad (9)$$

가 되며, 식(9)의 관계로부터

$$\tau_{inv} = M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + g(q_d) - f_{cc}(q_d) \quad (10)$$

로 정리되어 AMN의 출력이 로봇의 역동력학을 학습하면, $e^j(k) = 0$ 으로 수렴한다.

4. 시뮬레이션

AMN의 성능을 분석하기 위해 2 자유도 로봇 머니플레이터의 제적 제어를 수행하였다. 로봇 머니플레이터의 재원은 (표 1)에 제시하였고, 형상모형은 그림 (4)에 나타내었다. 기준케속은 주적문제와 regulation 문제 적합하도록 다음과 같은 정현파를 사용하고 6초후 지정된 위치에서 정지하도록 하였다.

$$q_{d1} = -\frac{\pi}{3}\cos\left(\frac{2\pi}{3}t\right)$$

$$q_{d2} = \frac{\pi}{2}\cos\left(\frac{2\pi}{3}t\right) - \frac{\pi}{4}$$

그리고 두 제어기에서 사용하는 이득은 $K_D = \text{diag}\{50 \ 50\}$, $K_P = \text{diag}\{80 \ 80\}$ 으로 정하고, 링크 2에 10Kg의 부하를 탑재한 상태로 학습을 진행시켰다. AMN 제어기 파라미터들로는 addressing을 위한 기준 위치케속 값에서 매 샘플링 순간마다 정수부 한자리와 소수부 5자리를 취하여($L=6$) 사용하였고, 학습률 $\beta = 0.01$, 샘플링 속도는 5msec로 정하였다. 그리고 실제 머니플레이터 적용이 가능하도록 로봇에 들어가는 전체 입력 토크는 각각 200Nm로 제한한 τ_p 를 사용하였다. 그림 (5)에는 제안한 방식의 제어기를 사용한 전체 제어 시스템을 도시하였고, 제안한 기법의 성능을 비교하기 위해 우선 AMN제어기를 사용하지 않은 PD 제어기만의 결과를 그림 (6)에 제시하였다. 그림 (7)은 AMN을 적용한 첫 학습후의 나타낸 결과로 PD에 비해 오차가 줄어들음을 알 수 있다. 그림 (8)은 20번 학습후의 결과인데, 위치 및 속도 오차가 매우 작은범위로 수렴되어 있음을 알 수 있고, 특히 토크성분을 관찰해 보면, 주된 제어기능이 PD에서 AMN 제어기로 옮겨갔음이 관찰되어, 이로 부터 AMN이 전절에서 기술한 로봇의 역동력학 학습을 제대로 수행할 수 있음을 알 수 있다. 또한 200Nm로 입력토크를 제한하였지만, 결과에서 알 수 있듯이 100Nm범위내에서 한정되고 있으므로 제안한 제어기법은 실제 머니플레이터에 적용하더라도 과도한 토크를 사용하지 않을 뿐만 아니라 무리한 토크출력으로 인한 제어 시스템의 내구성 저하를 방지할 수 있는 장점이 있다.

5. 결론

이 논문에서는 로봇 머니플레이터의 동력학 제어를 위해 새로운 형태의 AMN 제어기를 제시하였다. 제안한 방법은 로봇의 역동력학을 학습할 수 있는 능력을 지니며, 부하변동과 같은 외란적 요소가 가미되더라도 우수한 추적성능을 지닐 수 있다. 또한 CMAC과는 다른 사상규칙인 AMR을 사용하므로 설계 파라미터의 heuristic한 측면을 배제할 수 있어서 실제 적용시 체계적인 구현이 가능하며, 입력범위의 제한은 구현할 제어 시스템이 가질 수 있는 정밀도에만 영향을 받으므로 정규화 과정이 필요하지 않고, 임의의 입력범위에 대해 적용할 수 있는 장점이 있다.

4결의 실험에서 제시한 것처럼 AMN의 입력은 기준 위치 케속만을 사용하는데, 이것은 식(11)에서 제시한 함수인자의 종류와 갯수에서 많은 차이를 보이고 있으나 제시한 AMN은 입력을 함수의 인자로 사용하여 출력을 계산하는 것이 아니라 단지 출력을 담고 있는

메모리의 위치를 지정하는데 사용하므로 굳이 방정식에 제시된 인자를 모두 입력으로 사용할 필요는 없다. 즉, AMN에서는 입력에 따라 구별 가능한 메모리 위치만을 선택할 수 있으면 충분한 학습이 가능하다. 이 결과 모든 함수 인자를 사용할 때 보다 필요한 메모리의 양이 줄어드는 것은 명백한 사실이 된다.

참고문헌

- [1] J. S. Albus, "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller(CMAC)", *Trans. of the ASME, DSC*, pp. 220-227, 1975.
- [2] J. S. Albus, "Data Storage in the Cerebellar Model Articulation Controller(CMAC)", *Trans. of the ASME, DSC*, pp. 228-233, 1975.
- [3] C. Chung and T. Kuc, "Associative Memory Network Controller for Robot Manipulators with Generalized Trajectory Learning Ability", *to appear in IEEE Int. Conf. on SMC*, 1997.
- [4] M. Kawato, "Computational Schemes and Neural Network Models for Formation and Control of Multijoint Arm Trajectory", *Neural Networks for Control*, MIT Press, pp. 197-228, 1988.
- [5] D. E. Rumelhart et al., *Parallel Distributed Processing*, MIT Press, 1986.
- [6] Z. Qu, D. M. Dawson, *Robust Tracking Control of Robot Manipulators*, IEEE Press, New York, 1996.
- [7] 정재욱, 국태용, 이택중, "연관상상 메모리를 이용한 로봇 머니플레이터의 학습제어기 설계", 제 11 회 한국자동제어학술회의 논문집, 제 2 권, pp. 936-939, 1996

표 1 두부하시의 머니플레이터 파라미터 값

Table 1 Parameter values of the unloaded manipulator

링크/성분	m Kg	l[m]	lc[m]	I[Kgm ²]	kf	pf
1	10.0	1.0	0.5	0.833	1.0	4.0
2	5.0	0.7	0.35	0.204	1.0	4.0

역기속 \dot{q} , 정현파 샘플기터, l (관절모멘트), k_f (감속비), pf (Coulomb 마찰)

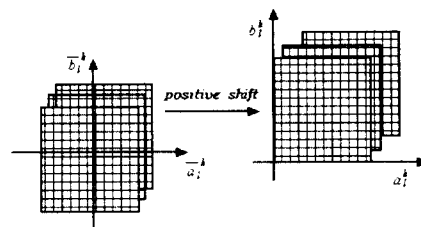


그림 1. AMN의 메모리 사상구조
Fig. 1 Memory mapping structure of AMN

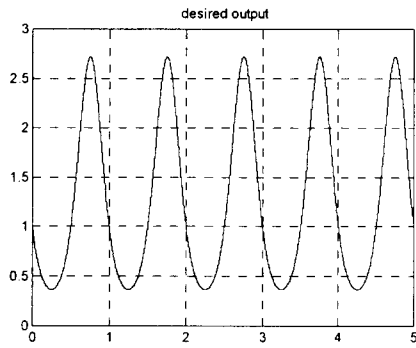


그림 2. 목표 근사화 함수 궤적
Fig. 2 Desired function trajectory

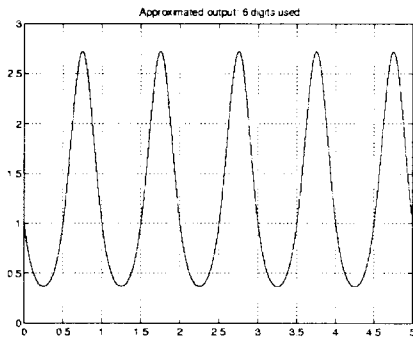


그림 3. 근사화된 함수 궤적
Fig. 3 Approximated function trajectory

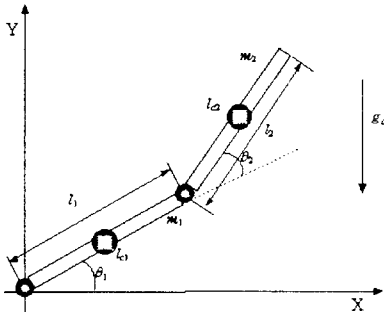


그림 4. 2 자유도 로봇 매니퓰레이터
Fig. 4 Robotic manipulator with 2 degree of freedom

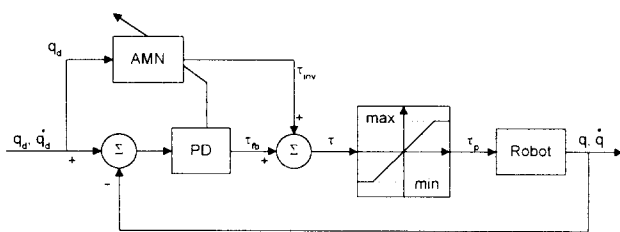


그림 5. 전체 제어 시스템의 블록도
Fig. 5 Block diagram of overall control system

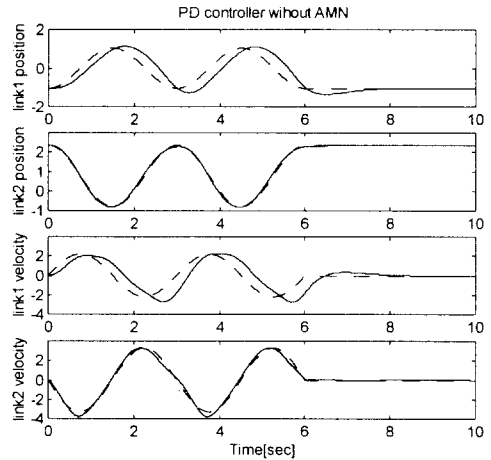


그림 6. PD 제어기의 위치와 속도 궤적
Fig. 6 Position and velocity of PD controller

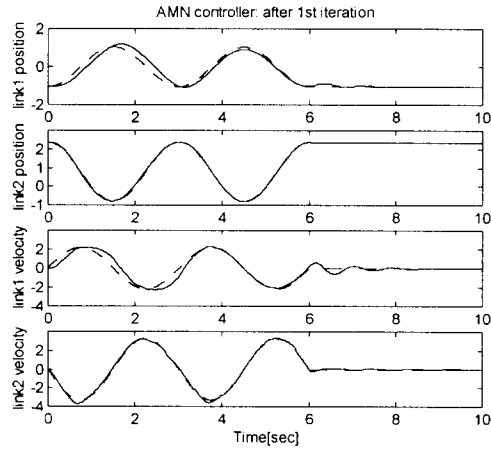


그림 7. 첫번째 학습후 AMN 제어기의 위치와 속도 궤적
Fig. 7 Position and velocity of AMN controller after 1st iteration

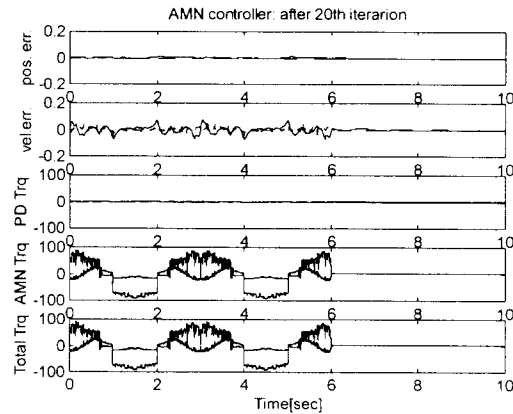


그림 8. 20번째 학습후 AMN 제어기의 위치 및 속도 오차와 토크성분
Fig. 8 Position and/or velocity errors and torques after 20th iteration