

## AMN을 이용한 반복학습 제어기의 성능개선

### Performance Improvement of Repetitive Learning Controller Using AMN

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 learning of robot trajectories. We use AMN controller in order to improve the performance of conventional learning control, e.g. RCL, which had studied by Sadegh et al. Computer simulations show the feasibility and effectiveness of the proposed AMN controller.

**Keywords** Robotics, Manipulators, Learning Control, AMN, Lyapunov

#### 1. 서론

산업현장에서 널리 사용되는 로봇 머니플레이터에 대한 제어기법 중 반복학습 제어기법은 로봇이 수행해야하는 작업계획의 종류가 일정하다는 점에 착안하여 원하는 궤적을 로봇에 적용한 뒤 그 결과에 따라 나타난 오차를 피드백하여 학습규칙에 따라 새로운 제어신호를 만드는 방식이다.

Arimoto [1] 등은 출력오차를 학습규칙에 사용하여 로봇의 출력을 개선시킬 수 있음을 보였고, 최근 Sadegh [4, 5] 등은 DCAL 기법을 수정하여 새로운 학습규칙(RCL)을 제시하였다. DCAL은 다른 적응제어 기법과는 달리 로봇 동역학의 회귀행렬식에 실제 궤적이 아닌 기준 궤적을 사용하므로써 로봇 적응제어 분야의 난점인 실시간 계산문제를 해결하였고, 이 기법에 사용된 회귀행렬식이 기준 궤적을 사용한다는 점을 이용하여 이의 결과를 가상의 기준 입력토크로 간주한 학습규칙(RCL)을 유도하고, 제어식의 조건에 따라 출력오차를 수렴시킬 뿐만 아니라 속도오차가 유계됨을 증명하였다. 그러나 이들 학습제어 기법은 원천적으로 제어 시스템의 샘플링 속도에 그 성능이 크게 좌우될 뿐만 아니라 고속의 샘플링 속도를 가질 수록 이전 주기 또는 학습횟수에서 계산했던 학습제어기의 출력을 더 많이 저장해야 하므로 메모리의 증가를 피할 수 없는 단점을 갖게 되었고, 특히 Sadegh의 경우 전체 제어 시스템의 안정도가 보장되기 위해서는 큰 제어기 이득을 가져야 하는 단점을 안고 있다.

이 논문에서는 AMN(Associative Memory Network) [2, 6]을 이용하여 Sadegh가 제시한 학습제어 구조를 개선하고자 한다. 제안한 AMN의 메모리 구조는 종래의 메모리와는 다른 addressing을 취하므로써 제어 시스템의 샘플링 속도가 빨라지더라도 메모리의 크기가 증가되지 않을 뿐만 아니라 Sadegh의 학습제어기와 비교하여 느린 샘플링 속도와 동일한 제어기 이득에서도 만족할 만한 제어 성능을 가지며, 아울러 머니플레이터의 부하변동과 같은 외란적 요소에도 강한 특성을 가진다.

다음 절에서는 Sadegh의 학습 제어기를 간략히 기술하며, 3절에서는 AMN의 구조와 학습 제어에 적용될 수 있는 근거를 마련하며, 4절에서는 제안한 구조의 제어 성능의 개선효과를 Sadegh의 제어기와 비교하여 설명하고, 마지막으로 결론을 내리고자 한다.

#### 2. Sadegh의 학습제어기

##### 2.1 로봇 머니플레이터의 동역학

N 자유도를 가지는 로봇 머니플레이터의 동력학식은 다음과 같이 주어진다.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = f_c(\dot{q}) = \tau \quad (1)$$

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  $g(q)$ ,  $f_c(\dot{q})$ ,  $\tau$ 는 각각 중력과 마찰력, 입력토크의 벡터들을 표시한다. 제어기를 설계하기 위해 식(1)의 로봇 시스템은 다음과 같은 성질을 만족한다고 가정한다 [3].

성질 1) 관성행렬  $M(q)$ 는 positive-definite과 symmetric 성질을 가지며 모든  $q$ 에 대해 유계되어 있다.

성질 2) 행렬  $\dot{M}(q) - 2C(q, \dot{q})$ 는 skew-symmetric 하다.

##### 2.2 RCL: Repetitive Control Law

Sadegh의 반복학습 제어기법(RCL)은 기준궤적이 주기적이라는 가정하에서 출발한다. 즉,

$$M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + g(q_d) + f_c(\dot{q}_d) = \tau_d \quad (2)$$

여기서  $q_d$ ,  $\dot{q}_d$ ,  $\ddot{q}_d$ 는 각각 기준궤적의 위치, 속도, 가속도이며,  $\tau_d$  바람직한 제어입력이 되고, 따라서 식(2)는 바람직한 동력학이라 할 수 있으며, 특히 기준궤적이 주기성을 가지게 되면 동력학식 내부에 미지의 상수 파라미터가 있더라도  $\tau_d$ 는 다음 식처럼 주기(T)성 또는 반복성을 갖게 된다.

$$\tau_d(t) = \tau_d(t - T) \quad (3)$$

식(2)로 주어진 동력학의 반복성을 이용하면,

$$\tau(t) = \hat{\tau}_d(t) + k_r r + k_p e + k_d \|e\|^2 r \quad (4)$$

과 같은 RCL(Repetitive Control Law)을 구성할 수 있고, 여기서  $\hat{\tau}_d(t)$ 는 학습항으로서  $\tau_d(t)$ 를 보상하는데 사용하며  $e \equiv q_d - q$ ,  $r \equiv \dot{e} - \dot{e}_0$ 로 정의한다. 그리고  $k_r, k_p, k_d$ 는 양수 이득을 나타낸다. 학습항은 주기적으로 매 샘플마다

$$\hat{\tau}_d(t) = \hat{\tau}_d(t - T) + k_L r \quad (5)$$

에 의한 학습규칙으로 수정되어 가며, 여기서  $k_L$ 은 양수 이득을 나타낸다. 이러한 학습항을 바람직한 학습항과의 차이값을 다음과 같이 정의하고, 양변에 적절한 대수적인 조작을 하면,

$$\begin{aligned} \hat{\tau}_d(t) &\equiv \tau_d(t) - \hat{\tau}_d(t) \\ \tau_d(t) - \hat{\tau}_d(t) &= \tau_d(t) - \hat{\tau}_d(t - T) - k_L r \\ \hat{\tau}_d(t) &= \hat{\tau}_d(t - T) - k_L r \end{aligned} \quad (6)$$

과 같은 학습오차 수정규칙을 유도할 수 있다.

시스템의 안정도를 분석하기 위해서 다음과 같은 Lyapunov-like 함수를 선정하자.

$$V = \frac{1}{2}r^T D(q)r + \frac{1}{2}k_p e^T e + \frac{1}{2k_L} \int_{t-T}^t \tilde{\tau}_d^T(\delta)\tilde{\tau}_d(\delta)d\delta \quad (7)$$

이를 시간에 대해 미분하면,

$$\begin{aligned} \dot{V} &\leq -\lambda_3 \|x\|^2 \\ &\leq 0 \end{aligned} \quad (8)$$

여기서  $\lambda_3$ 는 양수로  $\lambda_{min}(Q_0)$ 으로 주어지며, 이때

$$Q_0 \equiv \begin{bmatrix} k_p - \zeta_2/4 & -\zeta_1/2 \\ -\zeta_1/2 & k_c - 1/2k_L - \zeta_3 - \zeta_1/4 \end{bmatrix}$$

$$x \equiv \begin{bmatrix} \|e\| \\ \|r\| \end{bmatrix}$$

로 정의된다. 이러한 관계에서  $Q_0$ 가 positive-definite 성질을 유지하게끔 적절한 이득을 선정한다면 위치 및 속도 오차가 수렴 및 유계가 되도록 만들 수 있다. 즉, 주어진 시스템은 점근적으로 안정하게 된다.

이상의 RCL에 의한 제어기법은 안정한 제어 시스템을 유지할 뿐만 아니라 계산량이 다른 적응 제어기법에 비해 상당히 적음을 알 수 있다. 그러나 식(4)에서 알 수 있듯이 입력 토크를 계산하기 위해 필요한 학습항은 시스템의 샘플링 속도가 빨라짐에 따라(샘플링의 수가 많아질 수록) 상당한 양의 메모리가 필요하다. 아울러 안정도 증명시 시스템에 인가될 수 있는 외란요소(부하 변동 등)에 대한 고려가 없다는 것과 적정 수준의 큰 제어기 이득을 선정하지 않으면  $Q_0$ 행렬의 positive-definite 성질이 만족되지 않아 불안정해질 수 있다. 다음 절에서 RCL의 계산량에 대한 장점과 후술한 단점을 취사선택하여 새로운 형태의 메모리 구조로써 이를 극복하는 방법을 제시하고자 한다.

### 3. AMN의 구조 및 동작원리

#### 3.1 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 AMR은 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 시구간  $t \in [0, t_f]$ 에서 정의될때,  $n$ 은 입력변수의 갯수( $\gamma$ )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.

$$\begin{aligned} x_1(k) &= s_a^k a_1^k a_2^k \cdots a_l^k, \\ x_2(k) &= s_b^k b_1^k b_2^k \cdots b_l^k, \\ &\vdots \\ x_n(k) &= s_c^k c_1^k c_2^k \cdots c_l^k, \end{aligned}$$

여기서  $\{s_a^k, s_b^k, \dots, s_c^k\}$ 는  $\{x_1(k), x_2(k), \dots, x_n(k)\}$ 의 부호를 나타내며 소수점은 생략한 형태로 표시하였다. 위와 같이 표현된 입력변수를 AMR에 적용하기 위해서 각 자리를 양의 값을 갖도록 천이시켜서 해당 메모리에 addressing할 수 있는 번지로 표현한다. 즉,  $1 \leq l \leq L$ 의 경우

$$\begin{aligned} a_l^k &= s_a^k a_l^k + Base, \\ b_l^k &= s_b^k b_l^k + Base, \\ &\vdots \end{aligned}$$

$$c_l^k = s_c^k c_l^k + Base. \quad (9)$$

여기서

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

식(15)와 같이 표현되는 입력변수의 값으로 지정되는 메모리 사상과 이를 통한 출력은 다음 식으로 기술할 수 있다.

$$\begin{aligned} f(\cdot) &= [f_1(\cdot), f_2(\cdot), \dots, f_L(\cdot)]^T, \\ w_l(k) &= f_l(a_l^k, b_l^k, \dots, c_l^k), \\ y(k) &= \sum_{l=1}^L w_l(k), \end{aligned} \quad (10)$$

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.

#### 3.2 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.

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

그리하여 모든 입력변수에 대한 중첩자리를

$$m_d^k = \min \{m_a^k, m_b^k, \dots, m_c^k\} \quad (11)$$

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

$$w_l(k+T) = w_l(k) + \eta_1 \epsilon(k) \quad \text{for } 1 \leq l \leq L, \quad (12)$$

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

$$w_l(k+T) = \begin{cases} w_l(k), & (1 \leq l \leq m_d^k) \\ w_l(k-1) + \eta_2 \epsilon(k), & (m_d^k + 1 \leq l \leq L) \end{cases} \quad (13)$$

여기서  $\sum_{l=m_d^k+1}^L \eta_l = 1$ 이고, 초기값은  $w_l(0) = 0$ 으로 둔다. 위의 식에서 나타나는 분포 파라미터인  $\eta_1$ 과  $\eta_2$ 은 다음과 같이 정하였다.

$$\begin{aligned} \eta_1 &\equiv \frac{2(L-l+1)}{L(L+1)} \\ \eta_2 &\equiv \frac{2(L-l+1)}{(L-m_d^k)(L-m_d^k+1)} \end{aligned}$$

이상의 AMN의 동작특성을 이용하면 2절에서 기술한 Sadegh의 학습제어기 부분중 메모리 블록만을 대체 시키므로써(그림 2) 적용할 수 있고, 학습규칙의 경우 식(18)과 (19)의  $\epsilon(k)$ 를 Sadegh가 제시하였던 식(5)를 바로 이용하여 AMN의 메모리 수정규칙에 적용할 수 있다. 즉,

$$\epsilon(k) = \beta \cdot k_L \cdot r(k) \quad (14)$$

여기서  $\beta$ 는 적절한 학습속도 또는 수렴속도를 지정하는 양수로서  $0 < \beta \leq 1$ 의 범위내에서 정한다. 그러나 AMN의 입력에는 각 메모리 장소를 지정하기 위한 번지 값으로 기준좌측의 위치신호( $q_d$ )를 사용한다.

#### 4. 적용 예: 2 자유도 머니플레이터 제어

전술한 AMN과 RCL의 성능 비교를 위해 2 자유도 로봇 머니플레이터의 궤적 제어를 수행하였다. 로봇 머니플레이터의 궤적은 (표 1)에 제시하였고, 형상모형은 (그림 3)에 나타내었다. 기준좌측은주기  $2\pi$ 를 갖는  $\sin(t)$ 를 각 관절에 인가하였다. 그리고 두 제어기에서 사용하는 이득은 모두  $k_v = k_p = k_L = 50$ 으로 같았고, RCL의 경우 몇번의 실험결과를 통하여 적절한 샘플링 속도를  $1msec$ 로 한 반면, AMN에서는 실제 구현시 적절한 속도인  $5msec$ 로 하였고, 학습률  $\beta = 0.1$ ,  $L = 5$ 로서 정수부는 한자리, 소수부는 4자리로 한정하여 addressing에 사용하였고, 입력은 기준좌측의 위치신호만을 인가하였다. 실험에는 기준좌측의 위치와 속도를 무부하 상태에서 추종하는 부분과 두 제어기 모두 충분히 학습한 상태에서 링크 2에 부하 10Kg을 인가하는 부분(50초후)을 함께 수행하여 그 결과를 조사하였다. 실험결과(그림 4, 5) 두 제어기 모두 50초이내의 무부하 상태에서는 만족할만한 위치 및 속도 오차를 가지지만, AMN을 사용한 제어기가 더 작은 영역으로 수렴됨을 볼 수 있고, 50초 이후의 부하인가시 RCL은 위치 및 속도오차가 급속히 증가되는 반면, AMN 제어기는 증가된 오차가 다시 무부하 상태의 오차범위내로 수렴됨을 알 수 있다. 그림(6, 7)은 각 제어기의 출력 토크파형을 제시하였는데, 여기에서 RCL은 부하인가시 전체 토크 성분이 서서히 증가하는 추세를 보이고 있으며, AMN은 부하가 인가된 순간 큰 토크로 증가한 후 다시 부하를 감당할 만한 수준의 토크량으로 안정됨을 볼 수 있다. RCL의 경우처럼 증가추세의 토크는 로봇에 사용하는 구동기의 정격을 넘어설 수 있는 불안정 요인이 될 수 있다.

AMN의 토크파형 결과에서 알 수 있는 사실은 로봇을 제어하는 주제어기의 역할이 선형 제어기(PD)에서 AMN으로 옮겨간다는 점이다. 이는 AMN이 충분히 학습한 상태라면 제어대상의 역동력학을 학습하는 능력을 지녔다는 것으로 파악할 수 있고, Sadegh가 제시한 바람직한 토크  $\tau_d$ 에 RCL보다 가깝다는 것으로 판단할 수 있다.

#### 5. 결론

이 논문에서는 기존의 학습제어기를 개선시킬 수 있는 새로운 형태의 메모리 사상구조를 갖는 AMN 학습제어기를 제시하였다. 이 기법에서는 기존의 학습제어기에서 필요한 메모리 양 ( $N \times \text{int}[T/\text{Sampling rate}] = 2 \times \text{int}[2\pi/0.001] = 12566$ )에 비해 현저하게 적은 양( $N \times L \times (2M)^2 = 2 \times 5 \times 20^2 = 4000$ )으로 구성할 수 있으며, 이러한 메모리 양은 제어 시스템의 샘플링 속도나 기준좌측의 주기와는 무관하고, 단지 설계할 시스템에서 필요한 데이터의 정밀도에 따라 적절히 갖출 수 있으며, 실험결과로 미루어 과도한 데이터의 정밀도(L)가 전체 성능에 큰 영향을 주지는 않는다. 그리고 고속의 샘플링 속도가 적용되더라도 오히려 AMN의 기본 특성에 비추어 유사한 메모리 장소로의 사상이 일어나( $m_i^t$ ) 수정규칙에 대한 계산량이나 메모리 양의 증가가 요구되지 않는 특성을 지니고 있다. 아울러 로봇 제어에서 알 수 있듯이 부하 변동과 같은 외란요소에 뛰어난 대처능력을 갖추고 있을 뿐만 아니라 실제 구현시 타당한

출력토크를 생성하므로써 구동기의 내구성을 증가시키는 요인으로 작용할 수 있다.

#### 참고문헌

- [1] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering Operation of Robots by Learning", *J. of Robotic Systems*, vol. 1, pp. 123-140, 1984.
- [2] C. Chung, T. Kuc, "Associative Memory Network Controller for Robot Manipulators with Generalized Trajectory Learning Ability", to appear in *IEEE Int. Conf. on SMC*, 1997.
- [3] F.L. Lewis, C.T. Abdallah, and D.M. Dawson, *Control of Robot Manipulators*, Macmillan Pub., New York, 1993.
- [4] N. Sadegh and R. Horowitz, "Stability and Robustness Analysis of a Class of Adaptive Controller for Robotic Manipulators", *Int. J. of Robotics Research*, vol. 9, pp. 74-92, 1990.
- [5] N. Sadegh, R. Horowitz, W. Kao, and M. Tomizuka, "A Unified Approach to the Design of Adaptive and Repetitive Controllers for Robotic Manipulators", *Trans. of the ASME, DSC*, vol. 112, pp. 618-629, 1990.
- [6] 정재욱, 국태용, 이택중, "연관사상 메모리를 이용한 로봇 머니플레이터의 학습제어기 설계", 제 11 회 한국자동제어학회 논문집, 제 2 권, pp. 936-939, 1996

표 1 무부하시의 머니플레이터 파라미터 값  
Table 1 Parameter values of the unloaded manipulator

링크/성분	$m$ [Kg]	$l$ [m]	$lc$ [m]	$I$ [ $Kgm^2$ ]	$k_f$	$pf$
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

여기서  $l_i$  (질량중심거리),  $I$  (관성모멘트),  $k_f$  (점성마찰),  $pf$  (Coulomb 마찰)

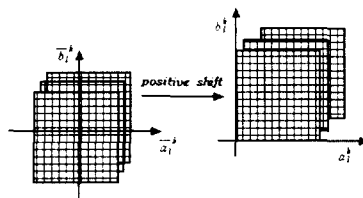


그림 1. AMN의 메모리 사상구조  
Fig. 1 Memory mapping structure of AMN

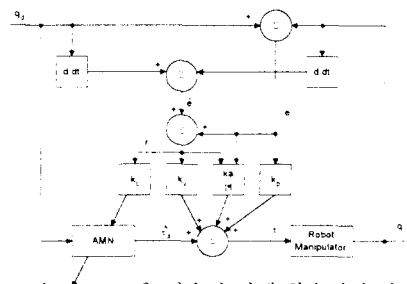


그림 2. AMN을 이용한 전체 학습제어 시스템  
Fig. 2 Overall learning control system using AMN

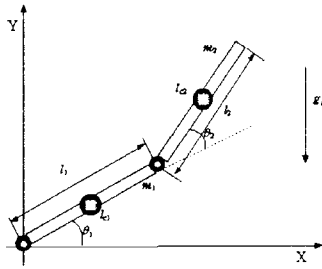


그림 3. 2 자유도 로봇 매니플레이터  
Fig. 3 Robotic manipulator with 2 degree of freedom

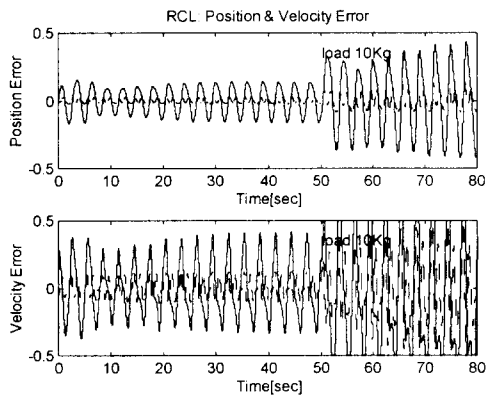


그림 4. RCL 제어기의 위치와 속도 오차(50초 이후 부하변동)  
Fig. 4 Position and velocity errors of RCL controller(payload change after 50 sec)

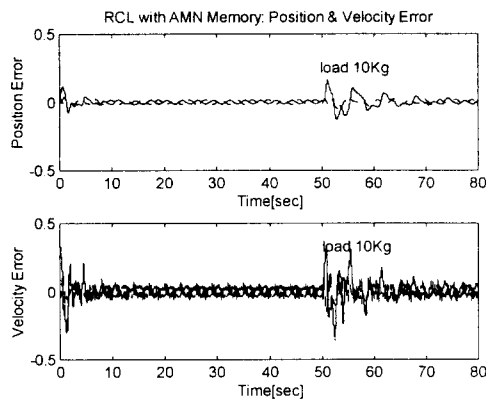


그림 5. AMN 제어기의 위치와 속도 오차(50초 이후 부하변동)  
Fig. 5 Position and velocity errors of AMN controller(payload change after 50 sec)

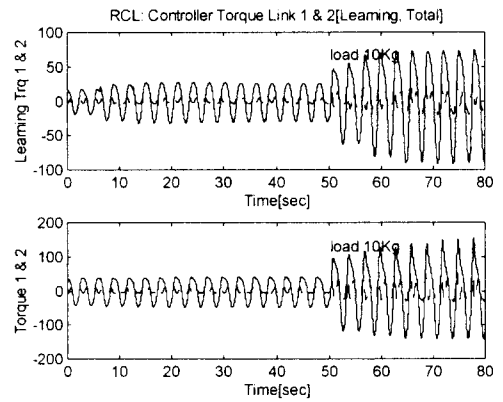


그림 6. RCL 제어기의 학습토크와 전체토크  
Fig. 6 Learning torques and overall torques of RCL controller

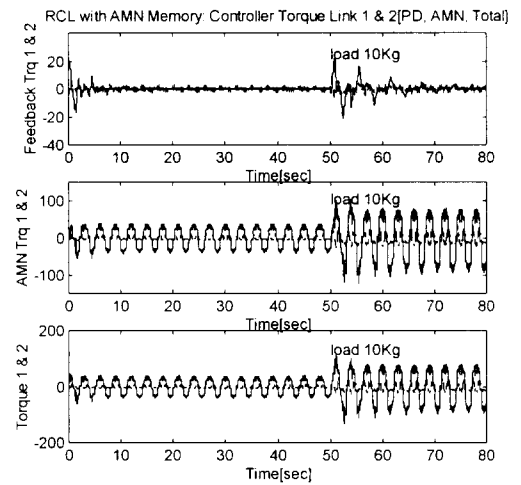


그림 7. AMN 제어기의 PD, 학습 및 전체토크  
Fig. 7 PD, AMN, and overall torques of AMN controller