

매니플레이터의 신경제어를 위한 새로운 학습 방법

* 경계현, 고명삼, 이범희

서울대학교 제어계측공학과, 자동화시스템공동연구소

A New Training Method for Neuro-Control of a Manipulator

Kye-Hyun Kyung, Bum-Hee Lee, Myoung-Sam Ko

Dept. of Control & Instrumentation Eng. Seoul National University,
Automation and Systems Research Institute

ABSTRACT: A new method to control a robot manipulator by neural networks is proposed. The controller is composed of both a PD controller and a neural network-based feedforward controller. MLP(multi-layer perceptron) neural network is used for the feedforward controller and trained by BP(back-propagation) learning rule. Error terms for BP learning rule are composed of the outputs of a PD controller and the acceleration errors of manipulator joints. We compare the proposed method with existing ones and contrast performances of them by simulation. Also, We discuss the real application of the proposed method in consideration of the learning time of the neural network and the time required for sensing the joint acceleration.

1. 서론

인공신경회로망은 간단한 연산자와 연결 네트워크로 구성되며 연결강도를 학습시킬 수 있다. 또한 일반성과 유연성을 가지며 self organizing 성질을 가지고 있기 때문에 시스템 자체나 외부 환경에서 발생하는 변화에 적응 능력을 갖는다.

그러므로 이러한 특성을 갖는 인공신경회로망을 이용한 매니플레이터 제어는 기존의 로봇 제어방법들이 갖는 문제점들 - computed torque 제어 방법[1,2]: 정확한 동역학 모델이 필요함, 적응제어[3,4,5,6]: 실시간 파라미터 추정을 위한 계산량과 잡음에 대한 민감도가 시스템의 상태 변수가 증가함에 따라 증가함, 반복 학습 제어[7,8]: 단지 반복적인 동작에만 적용할 수 있음 - 을 해결할 수 있다.

이미 인공신경회로망을 사용하여 로봇 매니플레이터를 제어하는 여러 방법들이 연구되어 왔다[9,10,11,12]. Miyamoto 등[9]은 PD 제어기와 Neuton-Euler 운동방정식의 비선형 항들을 입력으로 하는 단층 perceptron으로 구성된 앞먹임 제어기로 로봇 제어기를 구성하였다. 그들은 PD 출력력을 줄이는 방향으로 신경회로망을 학습시키는 Feedback-Error Learning을 제안하였다. 이 제어기의 단점은 신경회로망의 입력을 계산하는데 로봇의 동력학식에 있는 부분식들이 사용되므로 로봇 운동방정식을 알아야 한다는 점과 신경회로망 입력의 계산량이 많은 것에 있다. Miller III 등[10,11]은 CMAC신경회로망[14]을 사용한 로봇 제어 방법을 제안하였는데 이 모델에서 신경회로망은 로봇의 동력학 모델을 형성하도록 on-line으로 학습된다. 이 모델에서 신경회로망의 학습은 매니플레이터의 각 관절

의 위치와 속도뿐만 아니라 가속도까지 이용된다. 이 방법은 CMAC을 사용하므로 많은 기억 용량을 필요로 하며 한 제어주기 동안 두번의 신경회로망 feedforward 계산을 해야 하는 단점이 있다. Ozaki 등[12]은 관성행렬, 원심력과 Coriolis 및 마찰력 벡터 그리고 모델되지 않은 동력학 및 외란등을 다층 perceptron으로 각각 구현하여 computed torque 제어를 수행하였다. 이 방법에서 두 신경회로망은 관절의 위치와 속도를 줄이도록 각각 학습된다. 이 방법은 두개의 신경회로망을 사용하여 제어 및 학습이 복잡하며 또한 이들이 제안한 모델학습(model learning)은 동력학적 지식을 필요로 한다.

기존의 방법들의 문제점은 학습되지 않은 신경회로망 제어기의 수렴에 상당한 시간이 요구되며 또한 부하의 변동에 대해서도 상당한 시간의 재학습이 필요하다는 점이다. Leahy 등[13]은 이러한 문제점을 해결하기 위한 한 방법으로 신경회로망에 의해 부하를 추정하는 방법을 제안하였다. 본 논문에서는 새로운 신경제어기의 학습 방법에 의해 위의 문제점을 해결하고자 한다.

본 논문에서 로봇 제어기는 PD 제어기와 다층 perceptron으로 구성된 feedforward 제어기로 구성된다. 신경회로망은 BP[15]로 학습되는데 PD 출력과 매니플레이터 각 관절의 가속도 오차의 가속화된 합을 극소화 하도록 학습된다. 제안된 신경제어기에서 가속도 오차항은 단지 신경회로망의 학습에만 사용되는데 이에 의해 다음과 같은 이득을 얻을 수 있다. 학습되지 않은 제어기를 사용할 경우와 부하에 변동이 있을 경우에 가속도항은 위치나 속도항에 비해 큰 오차를 보인다. 따라서 이 두 경우 가속도 오차가 추가됨으로서 BP 학습에서 요구되는 에너지가 크게 되어 보다 빠른 수렴을 얻을 수 있다. 즉 가속도 오차항이 추가됨으로써 수렴 속도가 빠르게 되고 부하의 변화에 대해서도 빠르게 적응하게 된다. 또한 제안된 학습 구조는 신경회로망 제어기의 신경회로망 학습과 가속도 검출에 걸리는 시간지연을 고려한 실제 적용을 용이하게 한다.

2장에서는 사용한 신경제어기의 구조를 설명하고 Miller III 등[11]이 사용한 학습 방법, Miyamoto 등[9]이 사용한 학습 방법, 그리고 제안된 학습 방법을 비교한다. 3장에서는 시뮬레이션을 통해 세 방법의 성능을 평가하고, 4장에서는 신경회로망의 학습에 요구되는 시간과 가속도 검출에 걸리는 시간지연을 고려하여 제안된 모델을 실제 매니플레이터의 제어에 적용하는 방법에 대해 토론하고 시뮬레이션을 통해 그 가능성을 검토한다. 5장에는 결론을 제시한다.

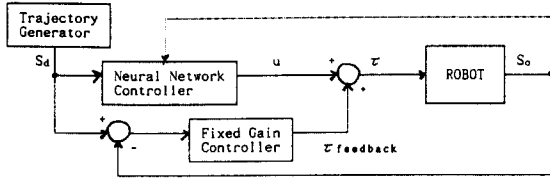


그림1 신경회로망을 사용한 로봇트 매니플레이터 제어기 구성

2. 신경회로망을 사용한 로봇트 제어

일반적으로 n 자유도를 갖는 로봇트 매니플레이터의 동역학은 (1)식으로 표현된다.

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

이 식에서 $\theta \in \mathbb{R}^n$ 는 관절 좌표를 나타내며, $D(\theta) \in \mathbb{R}^{n \times n}$ 는 항상 positive definite인 관성 행렬을, $h(\theta, \dot{\theta}) \in \mathbb{R}^n$ 는 Coriolis와 centrifugal 힘과 관련이 있는 행렬을, 그리고 $c(\theta) \in \mathbb{R}^n$ 는 중력 벡터를 나타낸다. $\tau(t) \in \mathbb{R}^n$ 는 일반화된 힘 벡터를 나타낸다.

그림1은 본 논문에서 구성한 제어기의 구조를 나타낸다. 그림1의 제어기는 다층 perceptron으로 구성된 앞먹임 제어기와 PD 선형되먹임 제어기로 구성된다. 선형되먹임 제어기는 매니플레이터 각 관절의 위치 되먹임 상수대각행렬 $K_p \in \mathbb{R}^{n \times n}$ 와 속도 되먹임 상수대각행렬 $K_v \in \mathbb{R}^{n \times n}$ 로 구성되며 궤적오차를 안정화시키는 역할을 수행한다[16]. n 은 매니플레이터의 자유도를 의미한다. 매니플레이터의 목표궤적과 관측궤적을 입력받는 신경회로망 제어기는 매니플레이터의 궤적을 목표궤적에 수렴시키는 역할을 수행한다[16]. 그림1에서 S_d 와 S_o 는 각각 목표관절궤적과 관측관절궤적을 나타내며, $u \in \mathbb{R}^n$ 와 $\tau_{feedback} \in \mathbb{R}^n$ 은 신경회로망 앞먹임 제어기와 선형되먹임 제어기의 출력 토크를 의미한다. 따라서 그림1의 제어기에서 매니플레이터의 제어입력 $\tau \in \mathbb{R}^n$ 는 (2)식으로 표현된다. (2)식에서 $\theta \in \mathbb{R}^n$ 는 관절좌표들을 의미한다.

$$\begin{aligned} \tau &= u + \tau_{feedback} \\ &= u + K_p(\theta_d - \theta_0) + K_v(\dot{\theta}_d - \dot{\theta}_0) \end{aligned} \quad (2)$$

신경회로망 제어기의 학습은 매 샘플링 시간마다 수행되는데, k 번째 샘플링 시간에서 신경회로망 제어기의 출력 $u(k)$ 는 (3)식과 같이 목표궤적과 관측궤적의 함수로 주어진다.

$$u(k) = g(S_{dk}, S_{ok}) \quad (3)$$

식(3)에서, S_{dk} 와 S_{ok} 는 k 번째 샘플링 순간에서의 궤적 계획기에서 발생된 목표궤적상태와 매니플레이터의 되먹임상태를 각각 나타낸다. 본 논문에서 신경회로망 제어기의 학습은 세가지 방법으로 각각 수행되는데 그 방법들은 다음과 같다.

(1) 방법1 : 첫번째 학습 방법은 Miller III 등[10,11]에 의해 제안된 방법이다. k 번째 샘플링 순간의 시작에서 궤적 계획기에서 발생된 목표궤적 S_{dk} 와 매니플레이터의 되먹임 상태(states) S_{ok} 를 입력받는 로봇트 제어기는 (2)식의 토크 입력 τ 를 매니플레이터에 전달하고 매니플레이터는 입력 토크에 의해 S_{ok+1} 로 이동하게 된다. 신경회로망 제

어기의 학습은 k 번째 샘플링 순간의 끝에서 이루어지는데, 먼저 $\tau' = g(S_{ok+1}, S_{ok})$ 가 계산되면, 신경회로망 제어기는 토크오차 ($\tau - \tau'$)를 극소화 하도록 backpropagation에 의해 학습되게 된다. 이때 τ 는 매니플레이터를 S_{ok} 상태에서 S_{ok+1} 상태로 이동시키는데 요구되는 이상 토크가 되며, τ' 는 같은 경우의 신경회로망 제어기의 실제 토크이므로 이 학습 방법에서 목적함수는 (4)식과 같게 된다.

$$E = \frac{1}{2} \sum (\tau_d - \tau_0)^2 \quad (4)$$

(2) 방법2 : 두번째 학습 방법은 Miyamoto 등[9]에 의해 제안된 방법이다. k 번째 샘플링 순간의 시작에서 (2)식의 토크 입력 τ 가 매니플레이터에 전달되면 매니플레이터는 S_{ok} 상태로부터 S_{ok+1} 상태로 이동 하게 된다. 그러면 k 번째 샘플링 순간의 끝에서 $K_v(\dot{\theta}_{dk} - \dot{\theta}_{ok+1}) + K_p(\theta_{dk} - \theta_{ok+1})$ 가 계산되고 이 값을 backpropagation 시퀀스에 의해 신경회로망 제어기의 학습이 이루어진다. 즉 신경회로망은 고정 이득 되먹임 제어기의 출력을 극소화하도록 학습된다. 그런데 고정 이득 되먹임 제어기의 출력은 이득 상수에 곱해진 매니플레이터 관절의 위치오차와 속도오차의 합으로 구성되므로 이 방법에 의해 신경회로망은 위치오차와 속도오차를 극소화하도록 학습된다. 이 방법에서 관측 가속도는 사용되지 않는다. 따라서 이 방법에서 목적함수는 (5)식이 된다. (5)식에서 ω 는 가중치이다.

$$E = \frac{1}{2} \sum \{ \omega [K_p(\theta_d - \theta_0)^2 + K_v(\dot{\theta}_d - \dot{\theta}_0)^2] \} \quad (5)$$

(3) 방법3 : 신경회로망 제어기의 세번째 학습 방법은 본 논문에서 제안한 방법으로 다음과 같다. 먼저 k 번째 샘플링 순간의 시작에서 (2)식의 토크 입력 τ 가 매니플레이터에 전달되면 매니플레이터는 S_{ok} 상태에서부터 S_{ok+1} 상태로 이동하게 된다. 그러면 k 번째 샘플링 순간의 끝에서 $\alpha(\theta_{dk} - \theta_{ok+1}) + \beta(\dot{\theta}_{dk} - \dot{\theta}_{ok+1}) + \gamma(\ddot{\theta}_{dk} - \ddot{\theta}_{ok+1})$ 가 오차로서 계산되고 이 오차를 backpropagation 시퀀스에 의해 신경회로망 제어기의 학습이 이루어진다. α 와 β 그리고 γ 는 가중치이다. 즉 신경회로망 제어기는 위치오차와 속도오차 그리고 가속도오차의 가중치화된 합을 극소화 하도록 학습된다. 이 방법에서 관측 가속도는 신경회로망의 학습에만 사용된다. 이 방법에서 목적함수는 (6)식으로 표현된다.

$$E = \frac{1}{2} \sum \{ \alpha(\theta_d - \theta_0)^2 + \beta(\dot{\theta}_d - \dot{\theta}_0)^2 + \gamma(\ddot{\theta}_d - \ddot{\theta}_0)^2 \} \quad (6)$$

3. 시뮬레이션 결과

신경회로망을 이용한 학습 제어기의 성능을 평가하기 위해 그림2의 2 자유도를 갖는 로봇트 매니플레이터의 제어에 학습 제어기를 적용하였다. 그림2의 매니플레이터의 동역학은 식(7)으로 표현된다[17].

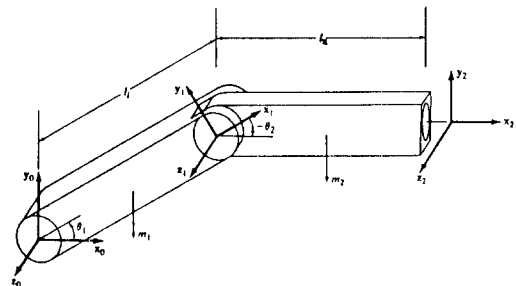


그림2 시뮬레이션을 위한 로봇트 매니플레이터

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} m_1 l_1^2 + \frac{4}{3} m_2 l_2^2 + m_2 C_2 l_1 l_2 & \frac{1}{2} m_2 l_2^2 + \frac{1}{2} m_2 l_1 l_2 C_2 \\ \frac{1}{2} m_2 l_2^2 + \frac{1}{2} m_2 C_2 l_1 l_2 & \frac{1}{2} m_2 l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -\frac{1}{2} m_2 S_2 l_1 l_2 \dot{\theta}_2^2 - m_2 S_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \\ \frac{1}{2} m_2 S_2 l_1 l_2 \dot{\theta}_1^2 \end{bmatrix} + \begin{bmatrix} \frac{1}{2} m_1 g l_1 C_1 + \frac{1}{2} m_2 g l_2 C_2 + m_2 g l_1 C_1 \\ \frac{1}{2} m_2 g l_2 C_2 \end{bmatrix} \quad (7)$$

식(7)에서 $C_1 \equiv \cos(\theta_1)$, $C_2 \equiv \cos(\theta_2)$, $S_2 \equiv \sin(\theta_2)$, 그리고 $C_{12} \equiv \cos(\theta_1 + \theta_2)$ 으로 정의된다. 또한 m_i 와 l_i ($i=1,2$) 는 i 번째 링크의 무게와 길이를 나타내며 g 는 중력 상수를 나타낸다. 시뮬레이션에서 사용한 조건들은 다음과 같이 하였으며, 신경회로망의 구조는 각각 20개와 10개의 뉴런으로 구성된 두 중간층을 갖는 4층 퍼셉트론으로 하였다.

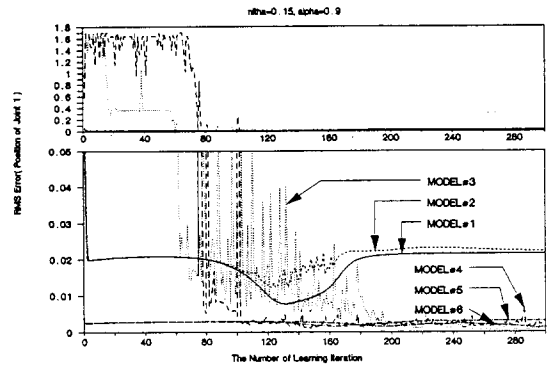
sampling time: $T = 5$ (ms);
 mass of link: $m_1 = 8$ (Kg); $m_2 = 5$ (Kg);
 length of link: $l_1 = 0.7$ (m); $l_2 = 0.5$ (m);
 feedback gain: $K_p = \text{diag}(150, 150)$ (N/rad);
 $K_v = \text{diag}(100, 100)$ (Ns/rad);
 desired trajectory: $\theta_1 = 5\pi/12 \cos(\pi t/2)$ (rad);
 $\theta_2 = -\pi/3 \cos(\pi t/2) + 2\pi/5$ (rad);
 initial condition: $\theta_1(0) = 5\pi/12$ (rad);
 $\theta_2(0) = \pi/15$ (rad);
 $\dot{\theta}_1(0) = \dot{\theta}_2(0) = 0$ (rad/sec);
 momentum in backpropagation learning rule: 0.9 .

신경회로망 제어기는 2.장에서 설명한 세가지의 각 학습 방법에 대해서도 신경회로망의 입력을 어떻게 구성하는가에 따라 그 성능이 달라질 수 있다. 본 논문에서는 신경회로망 제어기의 입력은 첫째 매니플레이터의 목표궤적과 관측 궤적 모두를 입력으로 한 경우와 둘째 매니플레이터 목표 궤적과 관측궤적의 위치와 속도만을 입력으로 한 경우의 두 가지로 구분하였으며, 표1에는 이에 따른 여섯가지 신경회로망 모델을 나타낸다. 표1에서 energy function은 각 방법에 대해 backpropagation 학습 법칙에서 요구되는 목적 함수를 의미한다.

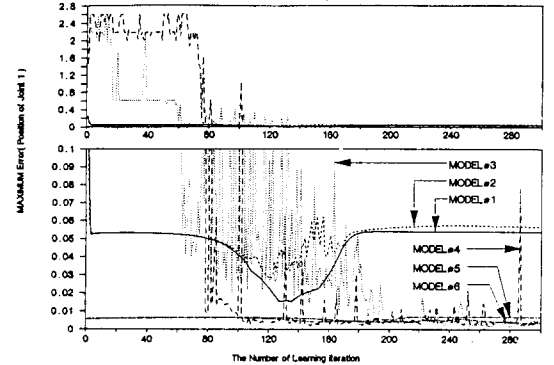
그림3과 그림4는 표1에서 설명한 여섯가지 모델로 신경 제어를 수행한 시뮬레이션 결과를 나타낸다. 신경회로망 제어기의 입력은 매니플레이터의 각 관절 별로 관절 영역, 최대 속도, 최대 가속도, 최대 허용 토크에 대해 0.1과 0.9 사이로 normalization 과정을 통해 구하였다. 한편 이들 그림에서 $nitha(\eta)$ 와 $alpha(\alpha)$ 는 backpropagation 학습 법칙에서 사용되는 학습률(learning ratio)과 모우멘텀 계수를 각각 나타낸다. 시뮬레이션에서 $\omega = 0.01$, $\alpha = 1.5$, $\beta = 1.0$, 그리고 $\gamma = 0.1$ 로 하였다. 즉 K_p 와 K_v 가 각각 150, 100이므로 방법3에서 backpropagation에서 사용된 오차항은 방법2에서 사용된 오차항에 0.1로 가중화된 가속도

표1. 신경회로망 모델 비교

method	model no.	input to NNC	energy function
(1)	1	$\theta_d, \dot{\theta}_d, \ddot{\theta}_d, \theta_o, \dot{\theta}_o, \ddot{\theta}_o$	$\frac{1}{2} \sum (\tau_d - \tau_o)^2$
	2	$\theta_d, \dot{\theta}_d, \theta_o, \dot{\theta}_o$	
(2)	3	$\theta_d, \dot{\theta}_d, \ddot{\theta}_d, \theta_o, \dot{\theta}_o, \ddot{\theta}_o$	$\frac{1}{2} \sum \{ \omega [K_v (\dot{\theta}_d - \dot{\theta}_o)^2 + K_p (\theta_d - \theta_o)^2] \}$
	4	$\theta_d, \dot{\theta}_d, \theta_o, \dot{\theta}_o$	
(3)	5	$\theta_d, \dot{\theta}_d, \ddot{\theta}_d, \theta_o, \dot{\theta}_o, \ddot{\theta}_o$	$\frac{1}{2} \sum \{ \alpha (\theta_d - \theta_o)^2 + \beta (\dot{\theta}_d - \dot{\theta}_o)^2 + \gamma (\ddot{\theta}_d - \ddot{\theta}_o)^2 \}$
	6	$\theta_d, \dot{\theta}_d, \theta_o, \dot{\theta}_o$	

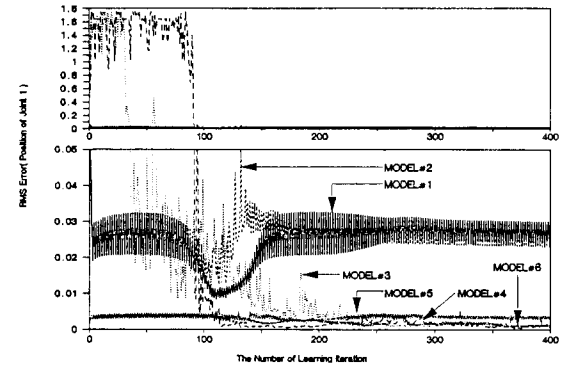


(a) RMS 오차

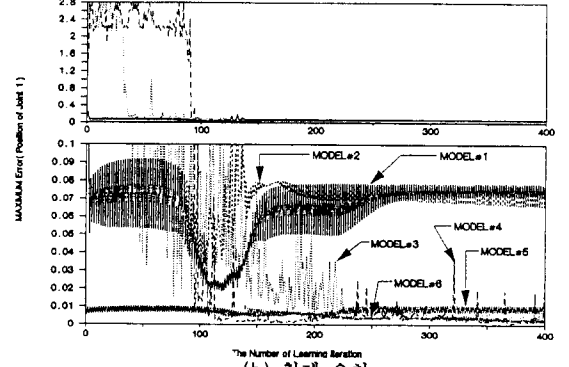


(b) 최대 오차

그림3. 학습 모델에 따른 신경제어기의 학습 곡선 (무부하의 경우)



(a) RMS 오차



(b) 최대 오차

그림4. 학습 모델에 따른 신경제어기의 학습 곡선 (홀수 주기: 무부하, 짝수 주기: 2Kg 부하)

오차량을 더한 것이 된다. 그림3은 무부하 상태에서의 시뮬레이션 결과이고 그림4는 학습의 홀수 주기에는 무부하 상태로 짝수 주기에는 2Kg의 부하를 부가한 상태로 학습한 시뮬레이션 결과이다. 이들 그림으로부터 학습 방법(1)은 비교적 빠른 수렴을 보이나 상당히 큰 수렴 오차를 보임을 알 수 있으며, 학습 방법(2)는 작은 수렴 오차에 비해 긴 수렴 시간을 나타냄을 알 수 있다. 본 논문에서 제안된 학습 방법(3)은 빠른 수렴 시간과 작은 수렴 오차를 보여 가장 우수한 성능을 나타냄을 알 수 있다. 즉 서론에서 설명한 것과 같이 가속도 오차량이 추가됨으로 수렴 속도가 매우 빠르게 된다. 그림3과 그림4로부터 다음의 결과를 얻을 수 있다.

첫째 신경제어기는 Miller III등이 제안한 학습 방법보다 Miyamoto등이 제안한 Feedback-Error Learning에 의해 학습될 때 더 작은 수렴 오차를 보인다.

둘째 학습에서 가속도량은 수렴속도를 빠르게하는데 중요한 역할을 수행한다.

셋째 신경제어기의 성능은 신경회로망 입력에 큰 영향을 받지 않는다.

그림6은 그림3과 그림4의 시뮬레이션에서 학습된 Weight 들을 갖는 신경제어기의 학습되지 않은 궤적에 대한 일반화 (generalization) 특성을 조사하기 위한 시뮬레이션 결과이다. 그림5는 이를 위한 매니플레이터 핸드의 카티시안 이동경로와 궤적을 나타낸다. 일반화 특성을 위한 시뮬레이션에서 매니플레이터는 주어진 원 경로를 첫 회전은 무부하로 두번째 회전은 2Kg의 부하 상태로 추적한다. 이 때 앞의 시뮬레이션에서 다른 두 방법에 비해 상당히 큰 수렴 오차를 나타낸 학습 방법(1)은 제외하였다. 학습 방법(3)의 경우에는 가속도량을 단지 학습시에만 사용한다는 가정하에서 모델#6만을 사용하였는데 그 이유는 4장에서 설명된다. 그림6-(a)로부터 모델#3은 시간이 지남에 따라 점점 오차가 증가함을 볼 수 있는데 이것은 모델#3의 경우 새로운 궤적에 대해 상당히 큰 오차를 보이는 관측 가속도 입력에 대한 신경회로망의 Weight들이 시간이 지남에 따라 관측 가속도 입력에 비례하여 커지게 되어 궤적 오차가 증가하게 되는 것으로 생각된다. 모델#4(그림6-(b))는 무부하 상태로 300주기 동안 사전 학습된 경우 부하 상태와 무부하 상태를 반복하여 학습된 경우보다 궤적 추적 초기에 비교적 큰 오차를 나타내는데 이것은 신경제어기의 일반화 능력에 있어서 전자의 학습 방법이 후자에 비해 불충분하기 때문으로 생각된다. 제안된 모델(그림6-(c))의 경우는 앞의 두 모델에 비해 보다 좋은 결과를 보임을 알 수 있다. 특히 제안된 모델의 경우 전혀 학습되지 않은 상태에서 수행된 궤적 추적 결과(그림6-(d))도 사전 학습된 결과에 비해 떨어지지 않는 성능을 나타냄을 볼 수 있다. 이것은 제안된 학습 방법이 서론에서 설명한 것과 같이 매우 빠른 수렴성을 지님을 나타낸다.

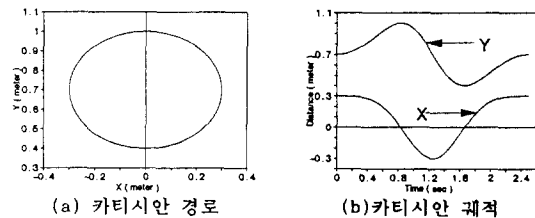


그림5. 신경제어기의 일반화 특성을 위한 경로와 궤적

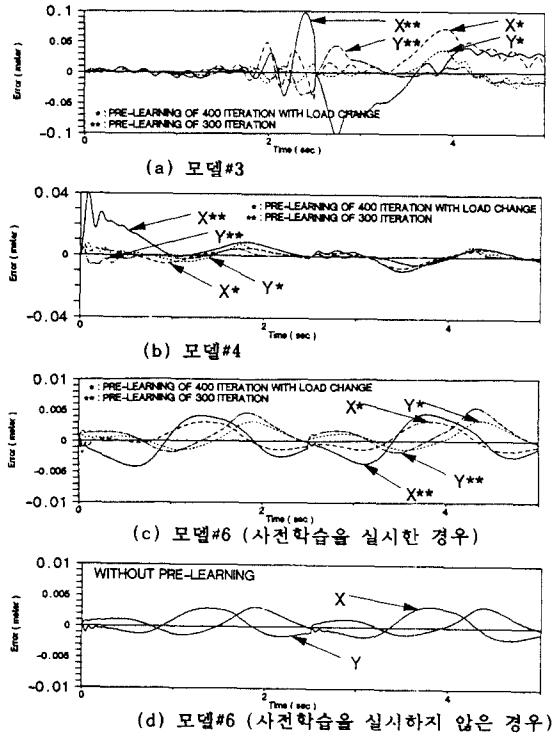
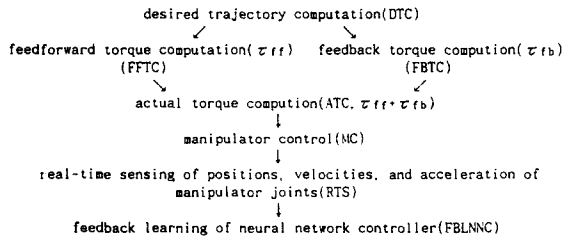


그림6. 그림5의 궤적에 대한 추적 오차

4. 토 의

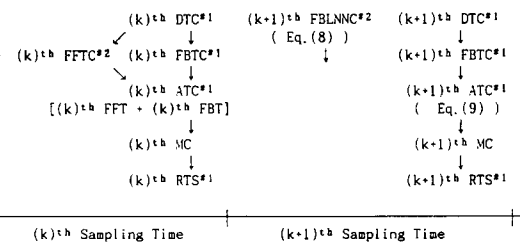
본 논문에서 제안한 방법으로 신경회로망을 사용하여 로보트 매니플레이터를 제어할 때 발생하는 문제점들 중 첫째 주어진 샘플링 시간 내에 신경회로망의 feedforward 계산과 feedback 계산을 모두 해야 한다는 점과 둘째 feedback 계산에서 요구되는 매니플레이터 각 관측의 각가속도의 감지에 현재의 기술로는 시간지연이 있게 된다는 점[8]은 실시간 매니플레이터 제어에 가장 큰 장애가 된다. 즉 제안된 제어기에서는 매 샘플링 시간마다



의 순서로 매니플레이터 제어가 이루어 지는데 특히 신경회로망의 feedback 학습에 소요되는 시간과 각 가속도 감지에서 생기는 시간 지연 때문에 제안된 제어기를 사용한 매니플레이터의 실시간 제어가 어렵다는 것이다.

이들 두가지 문제점들은 신경회로망 전용 프로세서와 제어 프로세서의 두 프로세서를 사용함에 의해 해결될 수 있다.

먼저 신경회로망의 학습에 소요되는 시간때문에 발생하는 문제점은 다음과 같은 계산과정을 통해 해결할 수 있다. 이 때 각가속도 감지에 시간지연이 없다고 가정한다.



#1은 제어 프로세서에 의해 통제되는 프로세스를 의미하며, #2는 신경회로망 전용 프로세서에 의해 통제되는 프로세스를 의미한다. 먼저 (k)번째 샘플링 시간에서 제어 프로세서에 의한 되먹임 토크와 신경회로망 전용 프로세서에 의한 앞먹임 토크의 합에 의해 매니플레이터가 이동하면 (k)번째 샘플링 시간의 끝에서 각 관절의 위치, 속도, 그리고 가속도가 감지된다. 그리고 나서 (k+1)번째 샘플링 시간에서는 신경회로망 전용 프로세서에 의해 그 오차값들의 가중화된 합((k)th ERROR)을 구한다. (k+1)th FBLNNC 를 위한 오차항(ERROR)은 학습이 수행되지 않는 샘플링 시간을 고려하여 (8)식과 같이 구한다.

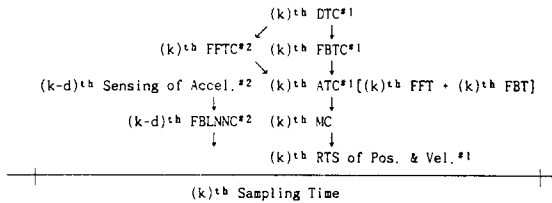
$$ERROR = (k)th ERROR + (k-1)th ERROR \quad (8)$$

또한 (k+1)번째 샘플링 시간에서 제어 프로세서는 토크가 선형이라는 가정하에서 이전의 샘플링 시간에 구한 신경회로망 출력들로부터 (9)식의 보간에 의해 앞먹임 토크를 계산한다. 이때 하나의 샘플링 시간이 신경회로망의 학습에 허용되게 된다.

$$(k+1)th FFT = (k)th FFT + ((k)th FFT - (k-2)th FFT) / 2 \quad (9)$$

그림7은 신경회로망의 학습 시간을 고려하여 그림3과 동일한 조건하에서 수행된 시뮬레이션 결과로 학습시간을 고려하지 않은 경우와 비슷한 성능을 보임을 알 수 있다.

각 가속도 감지에서 요구되는 시간지연 때문에 발생하는 문제점은 다음의 계산과정을 통해 해결될 수 있는데, 이 때 신경회로망의 학습에 걸리는 시간은 문제가 되지 않는다고 가정한다.



즉 매 샘플링 시간마다 목표궤적에 의해 되먹임 토크와 앞먹임 토크가 계산되면 제어 프로세서는 매니플레이터를 제어하게 되며, 신경회로망 전용 프로세서는 (k-d)번째 샘플링 시간의 각 가속도를 감지하여 (k-d)번째 샘플링 시간에서의 가중화된 오차항을 구해 그 값으로 신경회로망을 학습하게 된다. 이 때 샘플링 시간을 T, 목표궤적을 구하는데 걸리는 시간을 T_{DTC}, 그리고 앞먹임 토크를 구하는데 걸리는 시간을 T_{FFTC}라고 하면 가속도의 시간지연에 허용되는 시간은 (d-1)T + T_{DTC} + T_{FFTC}가 된다. 그림8은 가속도 감지 시간지연 고려하여(d=2) 그림3과 동일한 조건하에서 수행된 시뮬레이션 결과로 가속도 감지 시간지연을 고려하지 않은 경우와 비슷한 성능을 보임을 알 수 있다.

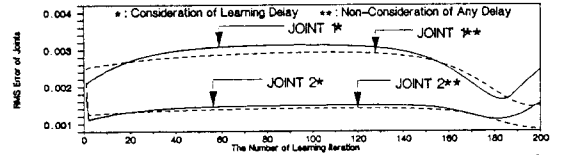


그림7. 신경회로망 학습시간을 고려한 경우의 학습 곡선

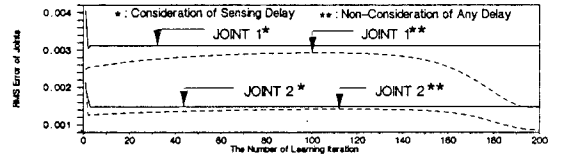


그림8. 가속도 감지 시간지연을 고려한 경우의 학습 곡선

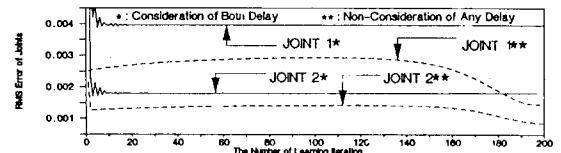
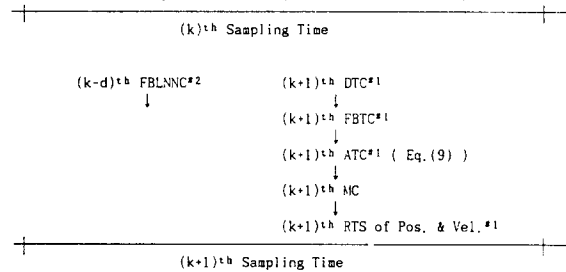
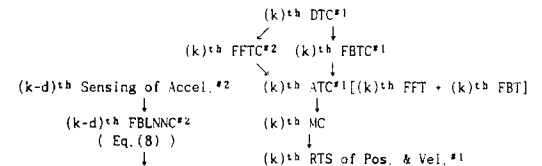


그림9. 신경회로망 학습시간과 가속도 감지 시간지연을 모두 고려한 경우의 학습 곡선

이상의 두 결과들로부터 신경회로망의 학습에 요구되는 시간과 각 가속도 감지에 요구되는 시간지연의 두 문제를 모두 해결할 수 있는 방법은 다음과 같이 제시될 수 있다.



즉 (k)번째 샘플링 시간에서는 목표궤적에 의해 구해진 되먹임 토크와 앞먹임 토크로 제어 프로세서는 매니플레이터를 제어하게 되고 신경회로망 전용 프로세서는 (k-d)번째 샘플링 시간의 각 가속도를 감지하여 (8)식 오차항을 구해 그 값으로 신경회로망을 학습하게 되며, (k+1)번째 샘플링 시간에서는 신경회로망 전용 프로세서는 계속 신경회로망의 학습을 수행하고 제어 프로세서는 (9)식으로 구한 앞먹임 토크와 (k+1)번째 샘플링 시간에서 구한 되먹임 토크에 의해 매니플레이터를 제어하게 된다. 이때 신경회로망의 학습에 허용되는 시간은 가속도의 감지에 소요되는 시간을 T_s라고 할 때 dT - T_{DTC} - T_{FFTC} - T_s가 되며, 가속도의 시간지연에 허용되는 시간은 (d-1)T + T_{DTC} + T_{FFTC}가 된다. 그림9는 신경회로망의 학습 시간과 가속도 감지 시간지연을 모두 고려하여(d=2) 그림3과 동일한 조건하에서 수행된 시뮬레이션

결과이다. 각각의 시간지연을 고려한 경우보다 오차가 보다 크게 나타나지만 여전히 시간지연을 전혀 고려하지 않은 경우와 비슷한 성능을 보임을 알 수 있다.

5. 결 론

본 논문에서는 다층 perceptron 신경회로망을 사용한 로보트 매니플레이터의 제어에 대해 서술하였다. 신경제어기는 고정 이득 제어기로 구성된 선형 피드백 제어기와 신경회로망 앞먹임 제어기로 구성된다. 신경회로망 제어기의 학습은 BP로 수행되었으며, 첫째 Miller III 등[11]이 사용한 학습 방법, 둘째 Miyamoto 등[9]이 사용한 학습 방법, 그리고 세째 본 논문에서 제안한 매니플레이터 각 관절의 위치와 속도 그리고 가속도의 가중화된 오차들을 이용한 학습 방법의 세 방법을 비교하고 시뮬레이션을 통해 제안된 학습 방법을 통해 빠른 수렴성을 얻을 수 있음을 보였다. 또한 본 연구에서는 제안한 방법의 제어를 실제 매니플레이터 제어에 적용할 때 발생하는 두 문제점들 즉 첫째 신경회로망 학습에 긴 시간이 걸린다는 점과 둘째 신경회로망의 학습에서 요구되는 매니플레이터 각 관절의 각가속도의 감지에 현재의 기술로는 시간지연이 있게 된다는 점에 대한 해결 방법을 제시하고 시뮬레이션을 통해 제안된 제어기의 실제 적용 방법의 가능성을 확인하였다.

참고 문헌

1. R.P.Paul, "Modelling, Trajectory Calculation and Servicing of a Computer Controlled Arm," Stanford Artificial Intelligence Lab. Memo AM-177, Nov. 1972.
2. J.Y.S.Luh, M.W.Walker, and R.P.Paul, "Resolved Acceleration Control of Mechanical Manipulator," IEEE Trans. Automat. Contr., Vol.AC-25, pp.468-474, 1980.
3. S.Dubowsky and D.T.DesForges, "The Application of Model Referenced Adaptive Control to Robot Manipulators," Trans. ASME, J. Dyn. Syst., Meas. Contr., Vol.101, pp.193-200, 1979.
4. A.J.Koivo and T.H.Guo, "Adaptive Linear Controller for Robotic Manipulators," IEEE Trans. Automat. Contr., Vol.AC-28, pp.162-171, 1983.
5. C.S.G.Lee and M.J.Chung, "An Adaptive Control Strategy for Mechanical Manipulators," IEEE Trans. Automat. Contr., Vol.AC-29, pp.837-840, 1984.
6. C.S.G.Lee and B.H.Lee, "Resolved Motion Adaptive Control for Mechanical Manipulators," Trans. ASME, J. Dyn. Syst., MEAS. Contr., Vol.106, No.2, pp.134-142, 1984.
7. S.Arimoto, S.Kawamura, and F.Miyazaki, "Bettering Operation of Robots by Learning," J. Robotics Syst., Vol.1, pp.123-140, 1984.
8. S.Kawamura, F.Miyazaki, and S.Arimoto, "Realization of Robot Motion Based on a Learning Method," IEEE Trans. Syst., Man, Cybern., Vol.SMC-18, No.1, pp.126-134, Jan./Feb. 1988.
9. H.Miyamoto, M.Kawato, T.Setoyama, and R.Suzuki, "Feedback-Error-Learning Neural Network for Trajectory Control of a Robotic Manipulator," Neural Networks, Vol.1, pp.251-265, 1988.
10. W.T.Miller, III, "Sensor-based Control of Robotic Manipulators Using a General Learning Algorithm," IEEE J. Robotics Automat., Vol.RA-3, pp.157-165, Apr. 1987.
11. W.T.Miller, III, R.P.Hewes, F.H.Glanz, and L.G.Kraft, III, "Realtime Dynamic Control of an Industrial Manipulator Using a Neural-Network-Based Learning Controller," IEEE Trans. Robotics Automat., Vol.6, No.1, pp.1-9, Feb. 1990.

12. T.Ozaki, T.Suzuki, T.Furuhashi, S.Okuma, and Y.Uchikawa, "Trajectory Control of Robotic Manipulators Using Neural Networks," IEEE Trans. Ind. Electron., vol.38, No.3, pp.195-202, June 1991.
13. M.B.Leahy, Jr., M.A.Johnson, and S.K.Rogers, "Neural Network Payload Estimation for Adaptive Robot Control," IEEE Trans. Neural Networks, vol.2, No.1, pp.93-100, Jan. 1991.
14. J.S.Albus, "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)," Transactions of the ASME, pp.220-227, September 1975.
15. D.E.Rumelhart, J.L.McClelland, and the PDP Research Group, "Learning Internal Representations by Error Propagation," Parallel Distributed Processing: Explorations in the Microstructures of Cognition, vol.1, MA: The MIT Press, pp.318-362, 1986.
16. 경계현, 고명삼, 이범희, "신경회로망을 사용한 로보트 매니플레이터의 학습 제어," '90한국자동제어 학술회의 논문집, vol.1, pp.30-35, 1990.
17. K.S.Fu, R.C.Gonzalez, and C.S.G.Lee, ROBOTICS: Control, Sensing, Vision, and Intelligence, McGraw-Hill Book Company, 1987.