

다층 신경회로 및 역전달 학습방법에 의한 로봇 팔의 다이내믹 제어

Dynamic Control of Robot Manipulators Using Multilayer Neural Networks and Error Backpropagation

吳世泳* · 柳淵植**
(Se-Young Oh · Yeon-Sik Ryu)

요 약

본 논문에서는 다층 신경회로를 이용한 PUMA560의 다이내믹 제어를 제안한다. 역전달 신경회로를 사용하였다. 신경회로는 임의의 비선형 변환을 수행할 수 있으므로 모터의 Feedforward 토크 발생기로 사용하였다. PD 보조 제어가 신경회로와 함께 사용되며, 신경회로의 학습에 사용된다. 신경회로는 로봇 팔의 현재 상태와 제환 PD오차 토크에 의하여 학습된다. BP회로를 Feedforward 토크 발생기로 사용함에 따라 로봇 팔의 다이내믹 특성에 대한 정보를 미리 필요로 하지 않으며, 그러한 정보들은 신경회로의 연결 값 사이에 함축적으로 저장된다. 다른 실험에서, 신경회로는 속도 센서를 사용하지 않고, 각 관절의 원하는 현재, 과거, 미래의 위치 값들로 학습되었다. BP회로는 이러한 시간의 변화에 따른 위치 값들로부터 각 관절의 속도와 가속도 성분을 추출하게 된다. 컴퓨터 모의 실험을 통하여 신경제어기의 강력한 특성들, 환경의 변화에 대한 적응성, 특정 잡음에 대한 강건성, 학습을 통한 성능의 향상을 보인다.

Abstract- A controller using a multilayer neural network is proposed to the dynamic control of a PUMA 560 robot arm. This controller is developed based on an error back-propagation (BP) neural network. Since the neural network can model an arbitrary nonlinear mapping, it is used as a commanded feedforward torque generator. A Proportional Derivative(PD) feedback controller is used in parallel with the feedforward neural network to train the system. The neural network was trained by the current state of the manipulator as well as the PD feedback error torque. No a priori knowledge on system dynamics is needed and this information is rather implicitly stored in the interconnection weights of the neural network. In another experiment, the neural network was trained with the current, past and future positions only without any use of velocity sensors. Form this time window of position values, BP network implicitly filters out the velocity and acceleration components for each joint. Computer simulation demonstrates such powerful characteristics of the neurocontroller as adaptation to changing environments, robustness to sensor noise, and continuous performance improvement with self-learning.

1. 서 론

최근 신경회로(Neural Network)를 이용한 제어(Neurocontrol)에 관한 연구가 활발히 진행되고 있다. 신경회로는 인간의 지능을 모방하여 구현하므로 지능제어(Intelligent Control)연구에 적합하리라는 생각 때문이다. 신경회로는 병렬 분산처리(Parallel Distributed Processing)에 의거하여 임의의 입출력 데이터 변환(Mapping)을 수행하므로 모든 계산을 대치할 수 있고, 정확한 계산 알고리즘에 의하지 않고 학습(Learning)에 의한 보간으로 출력을 발생하므로 선형 계산보다는 비선형 계산에 적합하다. 일반적으로 대부분의 제어문제는 제어 대상 즉 플랜트(Plant)의 입력에 대한 출력이 명확히 정의되므로 신경회로를 이용하여 비선형 제어문제에 응용하면 기존의 방법으로 처리하기 힘든 문제를 해결할 수 있으리라 생각된다.

일반적으로 기구학(Kinematics)을 통한 제어는 제어 대상의 다이내믹 특성(Dynamic Characteristics)을 고려하지 않으므로 역학적 조건이 변하는 상태에서 고속으로 움직이는 로봇트 팔의 정확한 다이내믹 제어에는 충분하지 못하다. 따라서 정확한 제어를 위해서는 로봇트 팔의 역학적 특성을 고려해야 한다.

기존의 다이내믹 제어(Dynamic Control) 방법중 로봇트의 다이내믹 모델로부터 원하는 궤적에 대한 관절 토크를 발생시키는 토크계산제어(Computed torque) 방법은 궤환 제어기의 Feed-forward 입력으로 동작하여 정확한 비선형 다이내믹 모델이 주어진 경우에는 훌륭한 성과를 기대할 수 있다. 그러나, 현재 다이내믹 모델 알고리즘에서 가장 효율적인 Recursive Newton-Euler 알고리즘도 6축의 경우, HP-9000-300-320 마이크로 컴퓨터에서 40msec[1]의 시간이 걸리고, 관절의 증가에 따라 계산시간이 선형으로 증가하므로 실시간 제어는 아직 힘들다. 또한 로봇트의 경우 정확한 모델을 얻는 것이 불가능하며, 로봇트 자체의 특성 변화에 따라 모델의 수정이 필요하다[2]. 따라서, 모델을 필요로 하지 않는 방법으로 적응제어(Adaptive Control)와 신경제어(Neurocontrol)를 이용하는 방법이 있다. 그러나 적응제어의 경우 Parameter Identification에 따른 많

은 계산양으로 실시간 제어가 힘들고, 계산오차에 대한 민감도와 상태변수가 증가함에 따라 크게 증가하는 측정오차가 문제점으로 알려져 있다. 이에 반해 신경회로는 병렬성(Parallelism)에 의한

- 1) 실시간 제어
- 2) Fault tolerance
- 3) 잡음에 대한 강건성(Robustness)

과 학습능력에 따른

- 1) 플랜트나 환경모델의 불필요
- 2) 제어 환경의 변화에 대한 적응성
- 3) 센서 데이터 합성(Sensor Data Fusion)의 용이성
- 4) 학습을 통한 제어능력의 향상
- 5) 어려운 제어 이론 지식 불필요
- 6) 간단한 Programming

등의 특성을 이용하여 로봇트 팔을 제어하면 기존의 많은 문제점들에 대한 해결방안을 얻을 수 있으리라 본다.

현재까지 신경회로는 크게 패턴인식, 음성인식, 최적화 문제, 화상처리, 제어 분야에 응용되어 왔다. 이중 제어문제에의 응용은 주로 기구학적 제어에 치중되어 있고, 다이내믹제어에 응용한 연구 결과는 그리 많지 않다. 그중 Albus의 CMAC[3]을 이용한 Miller의 2축 로봇트 팔 제어[1], Miyamoto가 단층 신경회로를 이용하여 3축 로봇트팔을 제어한 경우[2]와 Goldberg가 다층 신경회로 또는 Backpropagation(BP)회로를 사용하여 Direct-Drive Arm의 2축에 대해 "pick-and-place" 작업을 off-line 학습을 하여 신경회로의 일반화 특성을 보인 경우[4]가 대표적이다. Miller의 CMAC을 사용한 제어는 Table Lookup 방식을 사용함에 따라 정보의 입출력이 매우 빠르다는 장점이 있는 반면 제어하고자 하는 로봇트의 관절이 증가할수록 사용하는 기억소자의 양이 크게 증가하는 문제점이 있다. 단층 신경회로를 이용하여 역 다이내믹스(Inverse Dynamics)문제의 해를 구한 Miyamoto의 방법은 단층 신경회로만으로는 비 선형변환을 수행할 수 없으므로 입력단의 PE(Processing Element)들은 비 선형변환을 위한 보조시스템으로 부터 입력을 얻고 있다. 즉, 로봇트의 다이내믹 방정식에서 얻은 정보로 보조 시스템을 구성하고 신경회로는 다이내믹 방정식의 계수 값을 구하는 것이 된다. 이러한 방식은 제어 대상이 바뀌는 경우 새로운 보조 시스템을 구성하여야 하는 문제점이 있고, 그에 따라 제어 대상의

*正 會 員 : 浦項工大 電子電氣工學科 副教授 · 工博
 **正 會 員 : (株)金星社 情報機器研究所 컴퓨터 2室 研究員

接受日字 : 1990年 2月 10日
 1次修正 : 1990年 10月 30日

다이나믹 특성을 미리 알고 있어야 한다는 단점이 있다.

본 논문에서는 제어분야에 많이 응용되어온 다층 신경회로를 이용한 PUMA560의 다이나믹 제어에 관하여 연구하였다. 다층 신경회로는 BP학습을 하여, 임의의 입출력사이의 비선형 함수관계를 모델링할 수 있으므로 센서로부터 얻은 로봇 팔의 실제상태와 원하는 상태로부터 필요한 관절 Actuator 명령을 발생하도록 할 수 있다. 또한 제어 대상의 다이나믹 특성에 관한 정보를 여러 연결강도(Interconnection Weight)에 분산 저장함에 따라 Table Lookup 방식의 제어[1]에 비하여 적은 기억소자를 사용하여도 타당한 결과를 얻을 수 있다.

일반적으로 Feedforward 제어기는 위치와 속도 그리고 가속도로 구성된 원하는 상태의 값을 입력으로 하여 역 다이나믹스로부터 필요한 토크를 얻고 있다. 이를 위해서는 로봇의 역 다이나믹스를 알아야 한다. 그러나 BP회로를 사용함에 따라 로봇팔의 다이나믹 특성에 대한 정보를 미리 필요로 하지 않으며, BP회로를 Hardware로 구현할 경우 Feedforward 토크 값을 매우 빠르게 얻을 수 있으므로 실시간 제어에 적합하다.

본 논문에서의 BP제어 방식은 인가하는 입력의 종류에 따라 크게 둘로 나눌 수 있다. 첫째로, 로봇의 현재 위치와 속도 그리고 원하는 가속도를 신경회로 입력으로 사용할 수 있다. 둘째는, BP회로에 가고자 하는 관절 위치의 시간 Window 즉 과거, 현재, 미래값을 신경회로 입력으로 인가하고 제어에 필요한 속도와 가속도에 대한 정보는 BP회로 스스로가 추출하여 Feedforward 토크를 발생하는 방식이다[4]. 이렇게 함으로써 각 관절의 위치 센서만으로 로봇의 다이나믹 제어를 실현할 수 있게 된다.

컴퓨터 시뮬레이션(Simulation)에서는 End-effector의 부하(Payload)를 변화 시킴으로써 환경 변화에 대한 적응성을 실험하고, 각 관절 센서잡음에 대한 강건성을 알아보며, 다층 신경회로의 구조변화에 따른 성능 변화를 알아본다. 궁극적으로는 BP회로가 반복작업을 통하여 로봇 팔의 다이나믹 특성을 학습함에 따라 제어능력이 향상됨을 보인다.

2. 다층 신경회로의 구조 및 학습

2.1 다층 신경회로의 구조

그림 1처럼 입력단과 출력단 사이에 PE들이 하

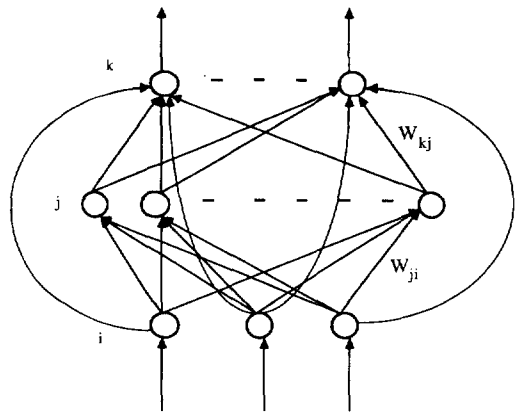


그림 1 다층 신경회로의 구조

Fig. 1 An architecture of multilayer neural networks.

나 이상의 중간단을 형성하는 경우 다층 신경회로라고 한다. w_{ji} 를 PE_i 에서 PE_j 로의 연결 강도라 할 때, PE_j 의 출력 O_j 는

$$o_j = f(\text{net}_j) \tag{2-1}$$

$$\text{net}_j = \sum_{i=1}^n (w_{ji} \cdot O_i) \tag{2-2}$$

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{2-3}$$

로 결정된다.

2.2 다층 신경회로의 학습

다층 신경회로는 임의의 입력에 대하여 원하는 출력으로 변환되도록 할 수 있다[5, 6]. 이는 입력단과 출력단 사이에 있는 중간단 PE에서의 비선형 변환에 의한 특성으로서, 임의의 입출력 영역에 대한 투사문제의 경우 입력단과 출력단은 선형활성화 함수를 갖고, 중간단의 PE는 비선형 활성화 함수를 갖는다. 제어에 필요한 토크의 값 영역을 모르는 경우에 출력단 PE의 활성화 함수를 임의의 기울기를 갖는 선형함수로 사용함으로써 적당한 활성화 영역을 선정할 수 있고, 그에 따라 PE사이의 연결강도의 조정이 용이하여 진다. 비선형 활성화 함수는 단조 증가하며, 미분 가능한 함수인 Sigmoid 형태를 많이 사용하고 있다.

다층 신경망 회로의 학습은 널리 알려진 바와 같이 Error BP rule[7, 8]에 의해 실행된다. Error BP rule은 임의의 입력에 대한 출력을 회로 외부에서 가르쳐주는 Supervised 학습방법에 의하여 원하는 출력으로 변환되도록 다음 식의 Cost function

$$E = \frac{1}{2} \cdot \sum (d_k - o_k)^2$$

d_k : 출력단 PE_k의 원하는 출력 값 (2-4)

o_k : 출력단 PE_k의 실제의 출력 값

가 연결공간(weight space)에서 Gradient Descent 방법으로 감소하도록 PE와 PE사이의 연결 강도를 조정하는 방법이다. 임의의 PE_k와 PE_j사이의 연결강도변화는

$$\Delta w_{kj} = -\eta \cdot \frac{\partial E}{\partial w_{kj}} \quad (2-5)$$

η : 학습율 (Learning rate)

일어난다. 다층 신경회로의 연결강도의 변화는 출력단과 중간단의 PE에서 다르게 정의된다. 출력단의 임의의 PE_k에 대한 연결강도 변화량은

$$\Delta w_{kj} = \eta \cdot (d_k - o_k) \cdot f'_k(\text{net}_k) \cdot o_j \quad (2-6)$$

이다. 또한 출력단과 직접 관련이 없는 PE의 사이의 연결 강도의 변화량은

$$\Delta w_{ji} = \eta \cdot o_i \cdot f'_j(\text{net}_j) \sum_k \delta_k w_{kj} \quad (2-7)$$

$$\Delta w_{ji}(n+1) = \eta \cdot \delta_j \cdot o_i + \alpha \cdot \Delta w_{ji}(n) \quad (2-8)$$

α : 운동량 (Momentum)

PE들의 연결 공간에서 과거 연결상태를 현재 연결강도의 진행 방향에 포함할 경우, 식(2-8)과 같이 하여 연결 강도의 고주파적 변화를 막을 수 있다.

3. 다층 신경회로를 이용한 로봇 팔의 3축 다이내믹제어

2장에서 설명한 다층 신경회로를 이용하여 제어 시스템을 구성하면 그림 2와 같다. S_0 는 측정된 관절 위치(θ_0)와 속도($\dot{\theta}_0$)로 구성되고 S_d 는 원하는 관절 위치(θ_d)와 속도($\dot{\theta}_d$)로 구성된다. BP회로는 Feedforward 제어기로 동작하도록 학습하며, 상수 이득을 갖는 PD 보조 제어기를 갖는다. 운동설계(Motion Design)에서는 원하는 관절 제어를 발생시켜 보조 제어기에 인가하고 로봇 팔의 상태를 센서로부터 입력받아 BP회로에 인가한다. 로봇팔의 다이내믹 모델은 Newton-Euler 방정식에 의해 다음과 같이 표시할 수 있다[9].

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + K(q)^T k = T \quad (3-1)$$

여기서,

$H(q)$: $N \times N$ 관성행렬

$C(q, \dot{q})$: $N \times N$ 원심력 및 Coriolis 행렬

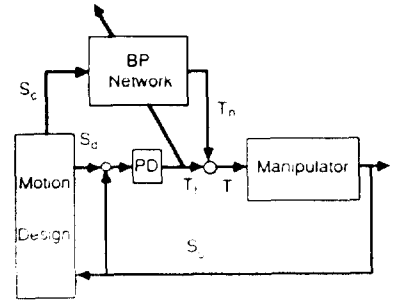


그림 2 다층 신경회로를 이용한 제어 구조
Fig. 2 The control scheme using multilayer neural networks.

$G(q)$: $N \times 1$ 중력 벡터

$K(q)$: $6 \times N$ Jacobian 행렬로서 링크 N에 작용하는 외부의 힘과 모멘트에 의해 각 관절에 발생하는 토크.

k : 6×1 벡터, 외부에서 링크 N에 인가되는 힘과 모멘트로 구성

T : $N \times 1$ 관절 토크 벡터

q : $N \times 1$ 관절 위치 벡터

식(3-1)에서 보듯이 각 관절의 토크는 각 관절의 위치, 속도 그리고 가속도와 비선형 함수 관계에 있다. 이러한 관점에서 BP회로가 임의의 함수를 모델하는 특성을 이용하여 로봇 팔의 역 다이내믹 모델을 학습하면 로봇 팔이 임의의 상태에서 다른 상태로 이동하는데 필요한 토크를 발생시킬 수 있다. BP회로는 로봇의 다이내믹특성을 전혀 모르는 것으로 가정하여 초기에 임의의 작은 연결강도를 갖는다. BP회로는 보조 제어기의 오차 출력을 감소하는 방향으로 식(3-2), (3-3)에 의해 학습한다. 그러므로 학습 초기에 로봇 팔의 제어에 필요한 토크의 대부분이 고정 이득을 갖는 보조 제어기에 의하여 공급되지만 학습에 따라 대부분의 토크는 점차적으로 BP회로에 의해 공급되고 보조 제어기는 작은 오차에 대한 보상만을 하게 된다.

$$T_n = k_p(\theta_{dn} - \theta_{on}) + k_v(\dot{\theta}_{dn} - \dot{\theta}_{on}) \quad (3-2)$$

$$\Delta w_{kj} = \eta \cdot \text{slope} \cdot T_{jk} \cdot o_j \quad (3-3)$$

$k = 1, 2, 3$

BP회로를 사용함에 따라 CMAC을 사용하는 경우 보다 좀더 분산 저장을 하므로 적은수의 기억소자를 사용하고, 단층 회로를 이용한 경우와는 달리 플랜트의 다이내믹특성 모델링을 위한 비선

표 1. 제안한 방법과 기존의 방법 비교

Table 1 Comparison of proposed methods with others

	Miller	Kawato	K. Goldberg	제안한 구조
모 델	CMAC	SLP	MLP	MLP
제어 대상	2축 평면	3축 공간 PUMA 260	2축 공간 CMU DD ARM II	3축 공간 PUMA 560
입력 변수	$\theta_o, \dot{\theta}_o, \ddot{\theta}_o$	$\theta_a, \dot{\theta}_a, \ddot{\theta}_a$	θ_a	$\theta_o, \dot{\theta}_o, \ddot{\theta}_o, \theta_a$
훈련방법	Supervised	Feedback error	Supervised	Feedback Error
샘플링 주기	10ms	10ms	---	5ms
회로 복잡도	200 weight	129 weights	420 weights	869 weights
훈련 상태	on-line	on-line	off-line	on-line
보조 제어기	P	PD	P	PD

형 변환 보조 시스템을 필요로 하지 않는다. 그 비교를 표 1에 나타내었다. BP로 인가되는 입력 $S_c(t + \Delta t)$ 를 두가지 방법으로 구성을 달리하여 시뮬레이션을 행하였고 Δt 는 샘플링주기(Sampling Period)이다.

3.1 위치, 속도 센서를 사용하는 경우

운동 설계에서는 각 관절의 센서로부터 각 관절의 위치와 속도를 입력으로 받고, 현재시간 t 에서의 원하는 상태(S_d)를 PD 보조제어기로 출력한다. 각 관절의 현재상태와 $5\Delta t$ 뒤의 원하는 위치로부터 원하는 가속도를 얻어 현재상태와 함께 BP의 입력, $S_c = (\theta_c, \dot{\theta}_c, \ddot{\theta}_c)$ 으로 인가한다. 즉, 각 관절의 상태

$$\begin{aligned} \theta_c(t + \Delta t) &= \theta_o(t) \\ \dot{\theta}_c(t + \Delta t) &= \dot{\theta}_o(t) \\ \ddot{\theta}_c &= \frac{2}{(5\Delta t)^2} \{ \theta_d(t + 5\Delta t) - \theta_o(t) - \dot{\theta}_o(t) \cdot 5\Delta t \} \end{aligned} \tag{3-4}$$

가 BP로 인가되므로, 3축 제어의 경우 입력단 PE는 9개가 된다. BP회로는 현재상태로부터 원하는 위치로 이동하는데 필요한 Nominal Feedforward 관절 토크(T_n)를 발생한다. PD 제어기의 출력(T_r)은 현재 시간에서의 오차를 보정하여 Nominal 토크와 함께 로봇트 팔에 인가되며, BP회로의 출력단 PE의 오차로 인식된다.

즉, Position, Velocity는 현재의 로봇트 상태를 반영하여 그대로 신경회로에 인가하고 앞으로 가야할 위치와 현 상태로부터 목표지점에 도달하는데 필요한 Acceleration의 값을 신경회로에 제공한다.

즉 로봇트의 현 상태에 기초하고 몇 Sampling 뒤의 위치에 기초한 Acceleration으로부터 신경회

로가 Torque를 생성하여 로봇트에 인가하고 만일 로봇트에 인가한 결과 원하는 궤적에서 이탈하였다면, 점차적으로 원하는 궤적으로 접근하도록 신경회로를 학습시킨다.

3.2 위치 센서만을 사용하는 경우

운동 설계에서 각 관절의 원하는 궤적에 대한 정보를 얻을 수 있으므로, BP의 입력으로 각 관절의 과거, 현재, 미래의 원하는 위치를 인가하여, BP회로가 위치성분으로부터 속도 및 가속도 성분을 추출하여 Feedforward 제어기로 동작하도록 한다. 즉 BP회로의 입력은 $(\theta_d(t - n\Delta t), \dots, \theta_d(t), \dots, \theta_d(t + m\Delta t))$ 등이다. 여기서 n, m 은 선택이 가능한 상수이다.

4. 컴퓨터 시뮬레이션

신경회로를 이용한 문제의 경우 그 특성을 충분히 살리기 위해서는 신경 컴퓨터를 사용하여야 하지만 아직 신경 Hardware 구현이 안된 상태이므로 VAX8800 에서 모의 실험을 행하였고, 프로그램은 C 언어로 구성되었다. 시뮬레이션에 사용된 BP회로는 2단의 중간단(각각 35개, 8개의 PE)을 사용하며, 출력단은 선형 활성화 함수(기울기: 0.0167, 절편: 0.5)를 사용한다. BP의 학습율은 0.1, 운동량은 0.1을 사용하였다. PD제어기는 위치 이득을 100, 속도 이득을 10으로 하였다.

원하는 궤적은 각 joint의 궤적을 다음과 같은 시간의 함수로 정의하였음.

$$\begin{aligned} \theta_1 &= -\frac{\pi}{4} \cdot \cos\left(\frac{\pi}{6}t\right) \\ \theta_2 &= \frac{\pi}{8} + \frac{\pi}{8} \cdot \cos\left(\frac{\pi}{3}\right) - \frac{\pi}{8} \cdot \cos\left(\frac{\pi}{1.5}t\right) \end{aligned}$$

$$\theta_3 = \frac{\pi}{4} - \frac{\pi}{4} \cdot \cos\left(\frac{\pi}{3}t\right) - \frac{\pi}{4} \cdot \cos\left(\frac{\pi}{1.5}t\right)$$

4.1 위치와 속도 센서를 이용한 제어

4.1.1 다층 신경회로를 이용한 제어

본 실험에서는 BP회로의 입력단은 각 관절의 상태로서 9 PE를 사용하였고, 출력단은 각 관절 Actuator에 인가되는 Feedforward 토크를 발생하므로 3개의 PE를 사용한다.

그림 3-1(a)는 PD 제어기만을 이용하여 제어하는 경우의 관절 2의 궤적을 나타내고있다. 그림 3-1(b)는 첫번째 시도과정의 관절 2의 궤적을 나타낸다. 초기 연결 상태의 경우보다 학습을 통한 결과의 개선을 볼 수 있다.

그림 3-1(c)는 41번째 시도에서의 관절 2의 궤적을 나타낸다. 전혀 학습하지 않은 경우와 처음 학습한 경우보다 학습을 통하여 원하는 궤적을 점점 더 가까와 짐을 알 수 있다.

그림 3-2(a)는 PD 제어기만을 이용하여 제어하는 경우의 관절 2의 각 속도 궤적을 나타내고 있다. 그림 3-2(b)는 첫번째 시도과정의 관절 2의 각 속도 궤적을 나타낸다. 초기 연결 상태의 경우보다 학습을 통한 결과의 개선을 볼 수 있다.

그림 3-2(c)는 41번째 시도에서의 관절 2의 각 속도 궤적을 나타낸다. 전혀 학습하지 않은 경우와 처음 학습한 경우보다 학습을 통하여 원하는 궤적에 점점 더 가까와 짐을 알 수 있다.

그림 3-3(a)는 PD 제어기만을 이용하여 제어하는 경우의 관절 2의 각 가속도 궤적을 나타내고 있다. 그림 3-3(b)는 첫번째 시도과정의 관절 2의 각 가속도 궤적을 나타낸다. 초기 연결상태의 경우보다 학습을 통한 결과의 개선을 볼 수 있다.

그림 3-3(c)는 41번째 시도에서의 관절 2의 각 가속도 궤적을 나타낸다. 전혀 학습하지 않은 경우와 처음 학습한 경우보다 학습을 통하여 원하는 궤적에 점점 더 가까와 짐을 알 수 있다.

그림 4는 전체시도에 따른 관절 궤적의 RMS오차의 감소를 보여주고 있다. 학습을 반복함에 따라 원하는 궤적으로 접근함을 알 수 있다.

그림 5-1은 학습의 진행에 따른 케환 보조 제어기 출력의 변화를 나타낸다.

그림 5-2는 학습의 진행에 따른 신경회로의 출력의 변화를 나타낸다.

즉, BP회로는 케환 오차가 감소하는 방향으로 학습을 하여 궁극적으로 다이나믹제어에 필요한

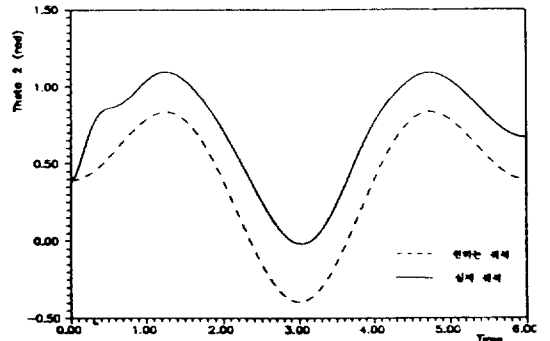


그림 3-1(a) 전혀 학습을 하지 않은 경우의 관절 2의 궤적

Fig. 3-1(a) The joint trajectory at the first attempt

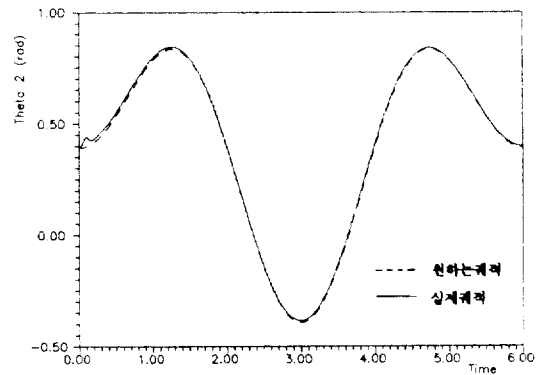


그림 3-1(b) 처음 시도 경우의 관절2의 궤적
Fig. 3-1(b) The joint 2 trajectory at the first attempt

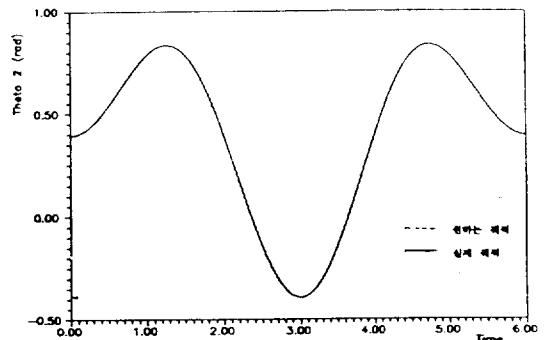


그림 3-1(c) 41회 시도 경우의 관절 2의 궤적
Fig. 3-1(c) The joint 2 trajectory at the 41th attempt

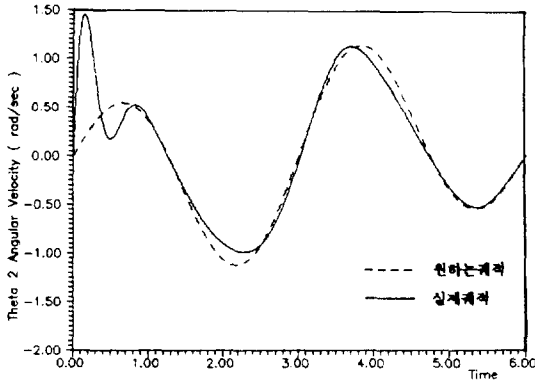


그림 3-2(a) 전혀 학습을 하지 않은 경우의 관절 2의 각 속도 궤적
 Fig. 3-2(a) The joint 2 angular velocity trajectory without learning

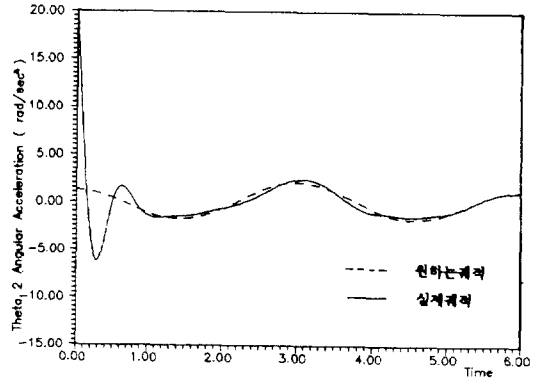


그림 3-3(a) 전혀 학습을 하지 않은 경우의 관절 2의 각 가속도 궤적
 Fig. 3-3(a) The joint 2 angular acceleration trajectory without learning

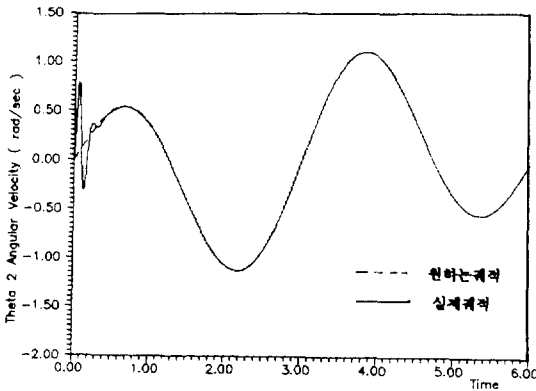


그림 3-2(b) 처음 시도 경우의 관절 2의 각 속도 궤적
 Fig. 3-2(b) The joint 2 angular velocity trajectory at the first attempt

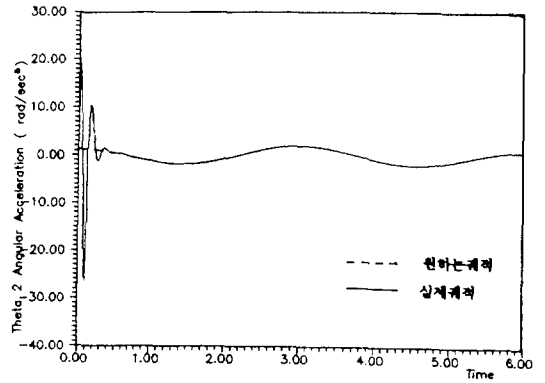


그림 3-3(b) 처음 시도 경우의 관절 2의 각 가속도 궤적
 Fig. 3-3(b) The joint 2 angular acceleration trajectory at the first attempt

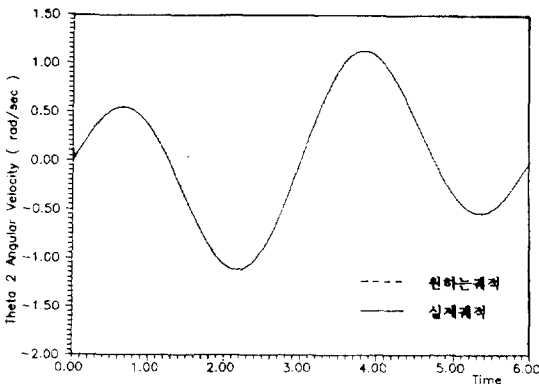


그림 3-2(c) 41회 시도 경우의 관절 2의 각속도 궤적
 Fig. 3-2(c) The joint 2 angular velocity trajectory at the 41th attempt

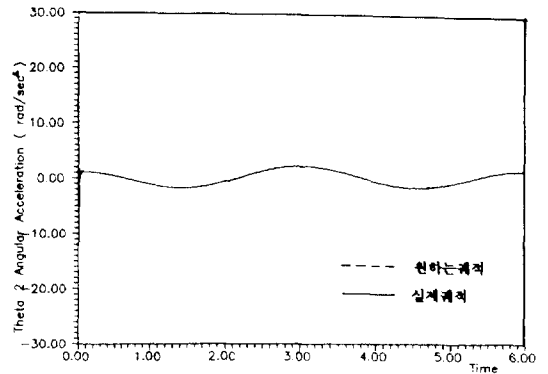


그림 3-3(c) 41회 시도 경우의 관절2의 각 가속도 궤적
 Fig. 3-3(c) The joint 2 angular acceleration trajectory at the 41th attempt

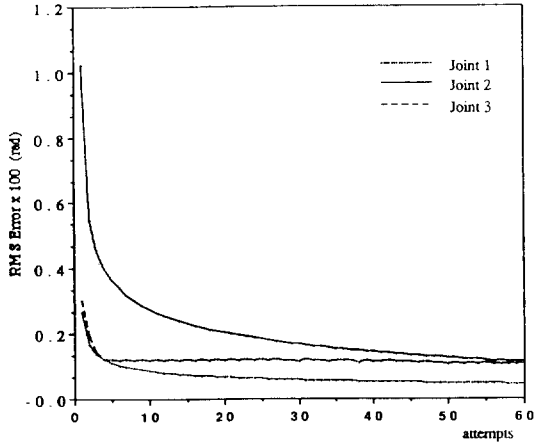


그림 4 학습에 따른 각 관절 궤적의 RMS오차의 변화

Fig. 4 The RMS joint trajectory error plot for each joint trajectory with learning

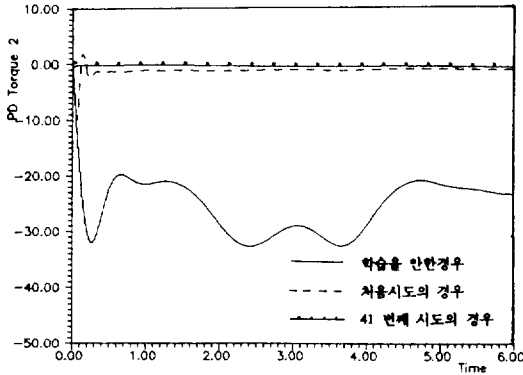


그림 5-1 학습에 따른 관절2의 궤환 토크의 변화
Fig. 5-1 The evolution of the feedback torque for joint 2 with learning

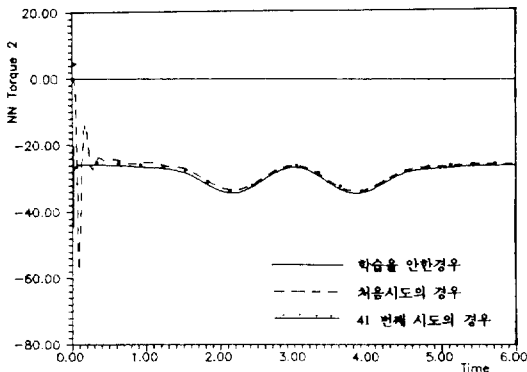


그림 5-2 학습에 따른 관절2의 신경회로 토크의 변화

Fig. 5-2 The evolution of the neural network torque for joint 2 with learning

다층 신경회로 및 역전달 학습방법에 의한 로봇트 팔의 다이내믹 제어

대부분의 토크를 발생하게 되며, 궤환 오차의 감소를 통하여 BP회로가 로봇트의 다이내믹 특성을 잘 학습하고 있음을 알 수 있다.

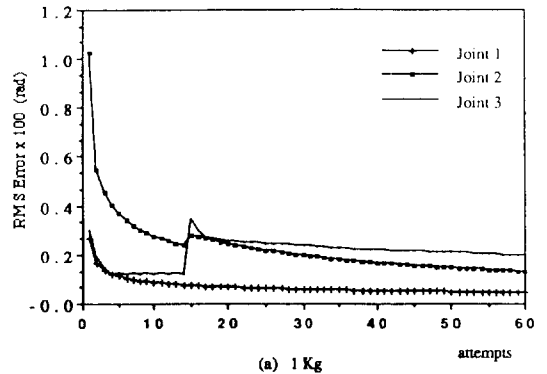
4.1.2 부하의 변화에 따른 적응

본 실험에서는 로봇트의 다이내믹 특성 변화에 따른 신경회로의 적응성을 알아보도록 한다. 로봇트가 학습도중 End-effector의 부하가 변화되는 경우, BP는 이미 학습하였던 것을 바탕으로 새로운 변화를 극복하도록 학습한다.

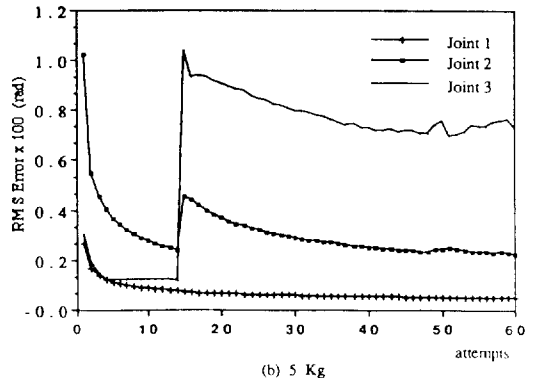
그림 6은 부하가 각각 1-Kg, 5Kg으로 변화된 경우의 결과를 나타낸다. 15번째의 시도에서 부하가 변하며 관절에서의 오차가 증가한 후 학습에 따라 제어환경의 변화에 따른 다이내믹 특성에 적응하고 있음을 볼 수 있다.

4.1.3 다층 신경회로의 구조변화에 따른 성능의 변화

본 실험에서는 BP회로의 구조를 변화시켜 보았



(a) 1 Kg



(b) 5 Kg

그림 6 부하의 변화에 따른 적응 실험

Fig. 6 Adaptability experiment for varying payloads

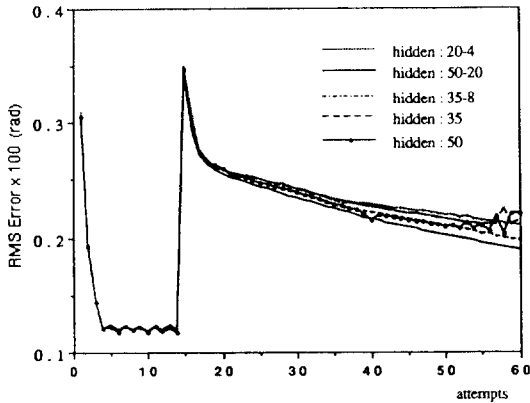


그림 7 다층 신경회로의 구조변화에 따른 성능 비교

Fig. 7 Performance comparison for different architectures of multilayer perceptron

다. 중간 PE의 수를 증가 혹은 감소하거나, 중간 단의 수를 변화시키면서 일정 시점에서 End-effector의 부하를 변화시킴으로써 다이내믹 특성 변화에 대한 여러 구조들의 특성을 보였다. 작업을 반복 수행하여 BP회로가 학습을 하는 도중, 어느 시점에서 End-effector의 부하를 1Kg으로 변화를 시켜 이전에 학습한 다이내믹 특성이 변화하는 경우 BP회로의 동작특성을 보고자 한다. 실험한 구조는 모두 4가지로서 입출력단은 변화가 없고, 각 중간단을 40개, 20개를 쓴 경우, 35개, 8개를 쓴 경우, 20개, 4개를 쓴 경우와 1단의 중간단을 사용하는 경우로써 각각 34개와 40개를 사용하여 실험하였다.

그림 7은 매 시도에 따른 관절 3개의 RMS오차의 변화를 나타낸다. 15번째 시도를 시작할 때 부하가 변화하였고, 그 이후의 시도는 부하의 변화에 적응하는 과정을 보이고 있다. 앞서서도 지적했듯이 아직은 어떤 문제에 필요한 최적의 PE 수는 알 수 없으나, 결과적으로 중간단의 증가나 각 단의 사용 PE증가는 학습의 특성을 좀 더 일반화하여 안정한 오차의 감소를 나타내고 있다.

4.1.4 측정 잡음에 대한 강건성

본 실험에서는 잡음에 대한 신경회로의 특성을 보고자 한다. 일반적으로 센서는 잡음에 노출되어 있다고 보는 것이 타당하다. 본 모의 실험에서는 속도센서가 잡음에 노출될 경우에 한하여 실험하였다.

그림 2에서 보았듯이 BP회로는 궤환 오차에 의

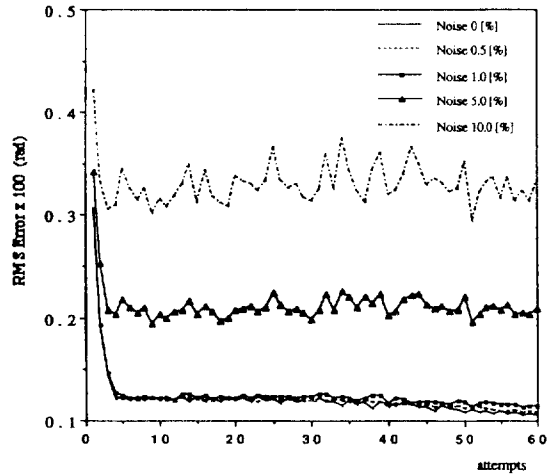


그림 8 센서 잡음에 대한 관절 3 궤적의 강건성
Fig. 8 Robustness of the joint 3 trajectory against sensor noise

하여 학습을 하고 있다. 즉 센서가 잡음을 많이 포함하는 경우 BP는 잡음에 오염된 정보를 통하여 학습을 하게 된다. 실험은 정확히 측정된 값에 대해 0.5%, 1%, 5%, 10%의 잡음이 존재하는 경우에 대하여 실행했다. 그림 8은 잡음에 대해 큰 특성의 변화를 보이는 관절 3의 경우를 보았다. 1%에서 5%사이의 잡음에 대해서는 BP회로가 견실하게 동작함을 볼 수 있다. (여기서 $x\%$ 라는 값은 정확히 측정된 값에 대해 $-x < \text{잡음}(\%) < x$ 의 Pseudo random 값이 더하여 진 것을 의미한다)

이는 시간의 경과하여 센서가 노후됨에 따라 정확한 정보의 전달이 힘든 경우 어느 정도까지는 BP회로가 센서의 부정확성에 잘 적응함을 보여 준다.

4.1.5 일반화 특성

본 실험에서는 신경회로의 큰 장점 중 하나인 일반화 특성을 알아보려고 한다. 일반화 특성을 알아보는 방법으로서, 각 관절이 궤적 A와 궤적 B를 번갈아 학습하고, 어느 정도 학습한 뒤 전혀 학습을 한 바 없는 궤적 C를 따라 가도록 하였다. 물론 궤적 C를 따르는 동안에 BP는 학습을 한다. 여기서, θ_{ai} 는 i 관절의 궤적을 의미한다.

관절 궤적 A의 식 :

$$\theta_{a1} = -\frac{\pi}{3} \cos\left(\frac{\pi}{3} t\right)$$

$$\theta_{a2} = \frac{\pi}{10} + \frac{\pi}{10} \cos\left(\frac{\pi}{3}t\right) - \frac{\pi}{10} \cos\left(\frac{\pi}{1.5}t\right) \quad (4-1)$$

$$\theta_{a3} = \frac{\pi}{6} - \frac{\pi}{6} \cos\left(\frac{\pi}{3}t\right) - \frac{\pi}{6} \cos\left(\frac{\pi}{1.5}t\right)$$

관절 제적 B의 식 :

$$\theta_{b1} = -\frac{\pi}{3} \cos\left(\frac{\pi}{3}t\right)$$

$$\theta_{b2} = \frac{\pi}{6} + \frac{\pi}{6} \cos\left(\frac{\pi}{3}t\right) - \frac{\pi}{6} \cos\left(\frac{\pi}{1.5}t\right) \quad (4-2)$$

$$\theta_{b3} = \frac{\pi}{3} - \frac{\pi}{3} \cos\left(\frac{\pi}{3}t\right) - \frac{\pi}{3} \cos\left(\frac{\pi}{1.5}t\right)$$

관절 제적 C의 식 :

$$\theta_{c1} = -\frac{\pi}{4} \cos\left(\frac{\pi}{3}t\right)$$

$$\theta_{c2} = \frac{\pi}{8} + \frac{\pi}{8} \cos\left(\frac{\pi}{3}t\right) - \frac{\pi}{8} \cos\left(\frac{\pi}{1.5}t\right) \quad (4-3)$$

$$\theta_{c3} = \frac{\pi}{4} - \frac{\pi}{4} \cos\left(\frac{\pi}{3}t\right) - \frac{\pi}{4} \cos\left(\frac{\pi}{1.5}t\right)$$

위의 식에서 보듯, 각 관절 제적 C는 제적 A와 제적 B사이에 존재한다. 그림 9는 제적 A와 제적 B를 번갈아가며 학습하다가 21회부터 제적 C로 학습함에 따른 관절 제적의 RMS오차의 변화를 나타낸다. 어느 정도 학습이 된후 제적 C를 인가한 결과 전혀 학습을 하지 않은 경우보다 오차가 감소하고 있음을 볼 수 있다. 즉 BP회로는 자신이 학습하지 않은 영역에 대해서도 어느 정도 일반화하고 있음을 볼 수 있다.

서로 다른 제적을 번갈아 학습함에 따라 오차가 진동을 하고 있으나, 같은 제적을 반복할 때는 오차가 매끄럽게 감소하고 있음을 볼 수 있다.

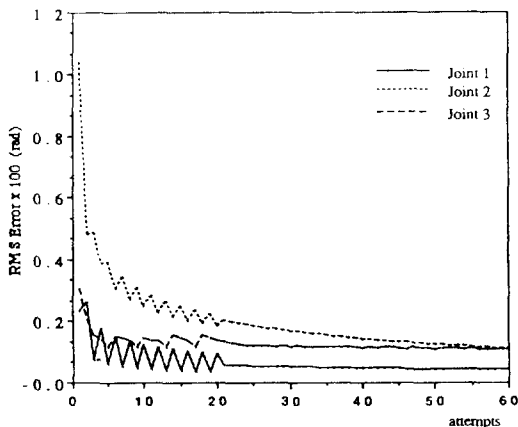


그림 9 다른 제적간의 일반화 특성
Fig. 9 Generalization among different trajectories

4.2 위치 센서만을 이용한 제어

본 실험에서는 BP회로의 입력으로 원하는 관절 제적으로 부터 얻은 위치 벡터를 인가하였다. 운동설계에서는 각 관절의 위치 벡터(S_c)와 오차보정을 위한 S_a 를 발생한다. 보조 제어기로 PD를 사용하지만 속도 성분은 관절의 위치 변화로 부터 수치 해석적으로 얻으므로 관절 위치센서만을 사용하여 위치에 대한 정보를 얻는다. 위치 벡터 S_c 는 3.2 에서와 같이 각 관절의 원하는 위치 값들

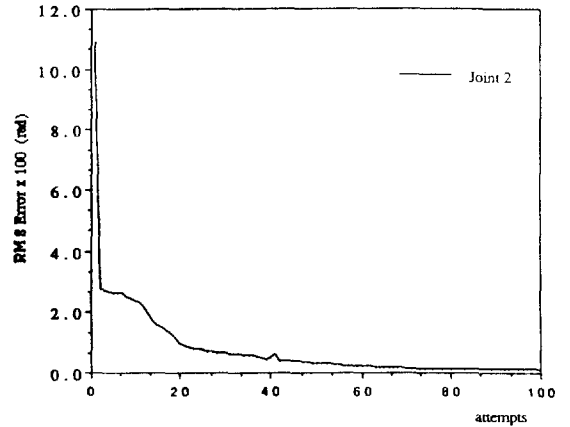


그림 10 위치 센서만 사용한 제어 실험의 RMS 제적오차
Fig. 10 The RMS trajectory error plot when only the position sensor is used

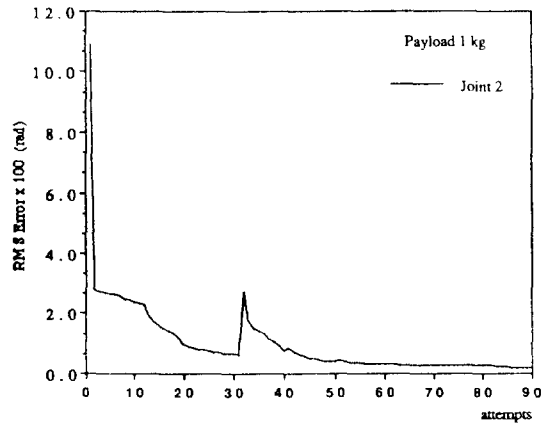


그림 11 위치 센서만 사용한 제어 실험에서 부하가 변하는 경우의 RMS제적오차
Fig. 11 The RMS trajectory error plot for varying payloads when only the position sensor is used

로 구성한다. BP회로는 로봇트 팔에 대한 다이내믹 정보는 전혀 모르는 것으로 가정하며, 과거, 현재, 미래의 목표 위치로부터 BP회로가 역 다이내믹모델링에 필요한 속도, 가속도성분을 추출하여 Feedforward 토크를 발생하도록 한다.

입력단은 33개의 PE를 사용하였다. 입력 값의 추출을 연속적인 값으로 하지 않음으로써 적은 PE로 넓은 상태 공간의 속도 및 가속도의 정보를 얻도록 하였다. 그림 10 은 관절 2 의 제적 RMS 오차를 나타내며, 그림 11 은 부하가 변하는 경우의 RMS 오차를 보여준다. 학습에 따라 관절제적의 RMS 오차는 감소하고 있으며, 부하의 변화에도 잘 적응하고 있음을 볼 수 있다.

5. 요약 및 결론

본 논문에서는 BP회로를 이용하여, PUMA 560 의 3 축에 대한 다이내믹 특성을 고려하여 각 관절의 제적을 제어하였다. BP회로의 비선형 변환 특성을 이용하여 로봇트 팔의 다이내믹 특성을 학습하게 하여 Feedforward 제어로 사용하였다. BP회로는 로봇트의 다이내믹 특성을 모르는 상태에서 PD 보조 제어기의 출력을 출력단 오차로 사용하며, on-line 학습을 통하여 제어기의 성능을 향상시킨다. 부하의 변화에 따른 적응 특성과 측정 잡음에 대한 강건성, 학습한 지식을 기초로 하는 일반화 특성등을 모의 실험을 통하여 기대하는 결과를 얻었다.

앞으로의 연구는 전체 제어 시스템의 안정도 (Stability)에 대한 연구와 함께 실제 로봇트 팔에 적용하여 시뮬레이션이 포함하지 못한 문제들에 대한 성능을 입증하고 또한 6 축 로봇트의 제어로 확장할 예정이다. 6 축의 경우에 있어서도 사용되는 PE는 3 축 경우 보다 크게 증가하지 않으므로 실시간 제어가 가능하리라 본다. 6 축에 대한 결과물을 얻는 경우 산업 로봇트에 응용하여 산업 및 공장 자동화 산업에 획기적으로 기여할 수 있다고 본다.

참 고 문 헌

- [1] W. Thomas Miller III, Filson H. Glanz, and L. Gordon Kraft III, "Application of a General Learning Algorithm to the Control of Robotic Manipulators", The International Journal of Robotics Research, Vol. 6, No. 2, pp. 84~98, Summer 1987.
- [2] Hiroyuki Miyamoto, Mitsuo Kawato, Tohru Setoyama, and Ryoji Suzuki, "Feedback-Error-Learning Neural Network for Trajectory Control a Robotic Manipulator", Neural Network, Vol. 1, No. 3, pp. 251~265, 1988.
- [3] Albus, J. S., "A New Approach to Manipulator Control the Cerebellar Model Articulation Controller (CMAC)", ASME Journal of Dynamic System Measurement and Control pp. 220~227, Sept 1975.
- [4] Ken Goldberg and Barak Pearlmutter, "Using a Neural Network to Learn the Dynamics of the CMU Direct-Drive Arm II", Poster session, IEEE Conference on Neural Networks, 1989.
- [5] Richard P. Lippmann, "An Introduction to Computing with Neural Nets", IEEE ASSP MAGINE, pp. 4~22, April 1987.
- [6] Ken-Ich funahashi, "On the Approximate Realization of Continuous Mappings by Neural Networks", Neural Networks, Vol. 2, 1989.
- [7] Yoh-Han Pao, "Adaptive Pattern Recognition and Neural Networks", Addison Wesley, 1989.
- [8] David E. Rumelhart and James L. McClelland, "Parallel Distributed Processing", MIT Press, 1987.
- [9] John J. Craig, "Introduction to Robotics: Mechanics & Control", Addison Wesley, 1986.