신경회로를 이용한 6축 로보트의 역동력학적 토크 제어

°조 문 증 오 세 영
포항 공대 전자 전기 공학과

An Inverse Dynamic Torque Control of a Six-Jointed Robot Arm Using Neural Networks

Moon-Jeung Joe and Se-Young Oh
Dept. of EE, Pohang Institute of Science and Technology

Abstract

Neural network is a computational model of the biological nervous system developed to exploit its intelligence and parallelism. Applying neural networks to robots creates many advantages over conventional control methods such as learning, real-time control, and continuous performance improvement through training and adaptation.

In this paper, dynamic control of a six-link robot will be presented using neural networks. The neural network model used in this paper is the backpropagation network. Simulated control of the PUMA 560 arm shows that it can move at high speed as well as adapt to unforseen load changes and sensor noise. The results are compared with the conventional PD control scheme.

1. 서론

기존의 로보트 Dynamic 제어에 있어서는 적당한 Dynamic Model에 대한 수치적 계산을 통한 제어를 하기 때문에 계속적으로 Dynamic 특성이 변화 하는 로보트의 경우 정확한 제어가 어려워지고, 따라서 로보트의 제작도 이러한 문제를 생각하여 구조를 정밀하게 제작하여 모델링하기 쉽게 하여야 한다. 그러나, 로보트의 정확한 Parameter를 구하는 것도 불가능 하지만 부품의 노후에서 오는 변화나 센서 잡음은 예측할수 없다. 특히, Kinematics만 고려한 현재의 로보트 제어 방법은 부하를 첨가하는 경우나, 고속동작에서 Dynamics를 고려한 제어보다 비효율적이다는 것은 잘 알려진 사실이다.

따라서, 이러한 문제를 해결하기 위하여 신경회로를 이용하여 생물체에서 이루어지는 Sensory Motor Control을 로보트 제어에 적용하고자 한다. 신경회로를 이용한 제어(신경제어, Neurocontrol)는 종래의 제어방법에서 볼 수 없는 다음과 같은 특징이 있다[1].

1) 신경제어는 모든계산이 각 뉴론(Neuron) 또는 PE(Processing Element)에서 독립적으로 병렬처리 된다.

따라서 실시간 제어가 가능하고 몇개의 PE의 고장으로 전체의 출력에 별로 영향이 없다. 또한, 입력 정보의 잡음은 출력의 영향을 크게 미치지 않는다.

2) 신경제어는 단지 학습으로 제어방법을 배우게 된다.

따라서 System Model이 필요 없어 Modeling하기 어려운 복잡한 System의 제어도 가능하며 System Parameter가 변화 하여도 적응 능력이 있다. 계속적인 학습은 System의 성능을 계속 개선시킨다.

현재까지 발표된 신경회로를 이용한 로보트의 Dynamic 제어는 아직 많지 않다. 그중 Miller가 Table Lookup 방식의 CMAC을 이용한 2축 로보트의 Dynamic Control이 있고[2], Miyamoto가 모의실험으로 3축 로보트를 Dynamics에 관계하는 Newton Euler Term을 입력으로 신경 회로가 제어하고 PD Controller로 학습하는 Feedback-Error-Learning이 있다[3]. Guez는 2축 로보트의 Dynamics Model을 Backpropagation Network이 학습을 통해 알아내는 방식으로 제어하는것을 보였다[4]. CMAC 방식은 많은 기억 용량이 필요한 것이 단점이고 Feedback-Error-Learning와 Guez는 어느 정도의 로보트 Dynamics에 대한 Term을 알아야 한다는 단점이 있다.

또한 산업용 로보트의 적용하기 위해서는 6축 제어를 필요로 한다. 그런데, 위에서 사용한 방식은 축수가 작은 경우 가능

한 방식이나 6축 정도의 경우 대단히 많은 Term을 필요하게 되고 따라서 많은 연산을 필요로 하게 된다. 특히 Miyamoto의 경우 필요한 Term을 구하는 것이 어려워지는 문제점이 있다.

본 논문에서는 처음으로 6축 로보트의 Dynamic 제어를 시도하였다. 로보트를 완전한 Black Box로 가정하고 Full Connection된 다층 신경회로(Multilayer Perceptron; Backpropagation 학습 사용)와 PD Controller를 병렬로 사용하여 궤환오차(Feedback Error)를 줄이는 방향으로 신경회로를 학습 시킨다. 신경회로는 원하는 Joint 각도-각속도와 현재 로보트의 각도-각속도만을 받아 필요한 Joint Torque를 발생한다.

로보트는 두가지 모드로 작동된다. 먼저 학습 모드(Learning Mode)에서는 로보트의 특성을 학습으로 알아내게 된다. 그후 제어 모드(Control Mode)에서는 이미 학습된 지식을 바탕으로 로보트를 동작 시키는 모드이다. 학습 모드는 로보트 공장에서 하게 되고 제어 모드만 실제 제어기에 넣으면 된다. 로보트의 성능이 떨어지게 되거나 새로운 작업이 부여 될때 다시 학습을 시키면 된다.

본 논문은 2장에서 학습 알고리듬(Algorithm)을 유도하고, 3장에서 제어구조를 설명한다. 4장에서 모의 실험결과를 보인다.

2. 다층 신경회로

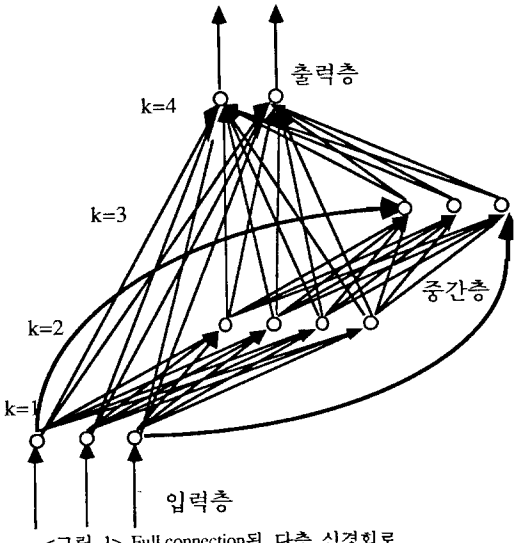
지금까지 발표된 많은 신경회로 모델이 있지만 제어 문제에 적용할 수 있도록 Nonlinear, Continuous Mapping에 적합한 모델은 Multilayer Perceptron이다[5]. 이 다층 신경망회로는 Neuron을 수학적으로 모델하여 이 Neuron을 여러층으로 쌓은 구조로 되어 있다. 각 입력신호를 받는 입력층과 이 입력층 신호를 비선형 Mapping 하는 중간층과 출력을 내는 출력층으로 이루어져 있다. 본 논문에서 사용한 구조는 Full Connection 된 다층 신경회로를 사용한다(그림. 1).

W^k_{ji}를 k번째층에 PE_i(하위층 i번째 PE)에서 PE_j(상위층 j번째 PE)로의 weight라 할 때 PE_j의 출력 O_j은 다음과 같이 표시된다.

$$net_j = \sum_{i=0} W^k_{ji} O^k_i \qquad (2\text{-}1)$$

$$O^{k+1}_j = f(\, net_j \,) \qquad (2\text{-}2)$$

여기서 함수 f(.)는 활성함수(Activation Function)라 하고, 보통의 경우 Sigmoid 함수를 사용한다. 식 (2-2) 의 출력은 다음층



<그림 1> Full connection된 다층 신경회로
<Fig. 1> Fully connected multilayer perceptron

으로 이동하게 되고 중간단 PE 에서의 비선형 변환에 의한 특성으로 임의의 입출력에 대한 비선형 Mapping이 가능하게 된다. 제어 문제와 같이 적당한 출력의 범위를 알수 없을 경우 출력층의 활성함수는 적당한 기울기를 가지는 선형함수를 사용할수 있다. 신경제어기의 출력 τ_n은 다음식과 같이 k=4의 층에서 나오게 된다.(식 2-3)

$$\tau_n = O^4 \qquad (2\text{-}3)$$

다층 신경회로의 학습은 Backpropagation rule(BP rule)에 의해 실행된다[5]. 먼저 로보트의 토크 오차에 해당하는 Energy Function을 정의 하면

$$E = \frac{1}{2} \cdot \sum (\tau_d - \tau_n)^2 \qquad (2\text{-}4)$$

τ_d : 원하는 출력 토크값
τ_n : 실제의 출력 토크값

가 된다. 그러나, τ_d는 구할 수 없으므로 위식은 사용할 수 없다. 따라서, 궤환 제어기의 출력을 Energy Function으로 사용한다. PD 제어기만 사용할 경우 PD제어기의 출력 τ_f는 식 (2-5)과 같이 주어진다.

$$\tau_f = k_p (q_d - q) + k_v (\dot{q}_d - \dot{q}) \qquad (2\text{-}5)$$

따라서 Energy Function을 다음과 같이 다시 정의한다.

$$E = \frac{1}{2} \cdot \sum \tau_f{}^2 \qquad (2\text{-}6)$$

식(2-6)은 오차가 0가 되면 E=0가 되므로 합당한 Energy Function이라 할수 있다.

Energy가 감소하도록 weight를 바꾸는 것은 Gradient Descent

Rule을 사용한다.

상위층 PE_m와 하위층 PE_j 사이의 weight의 변화는

$$\Delta w_{mj}^k = -\eta \cdot \frac{\partial E}{\partial w_{mj}^k}$$

$$= -\eta \cdot \frac{\partial E}{\partial net_m} \cdot \frac{\partial net_m}{\partial w_{mj}^k}$$

$$= -\eta \cdot \frac{\partial E}{\partial net_m} \cdot o_j^k \qquad (2\text{-}7)$$

이다.

위식에서 δ_m를 다음과 같이 정의 한다.

$$\delta_m^k = -\frac{\partial E}{\partial net_m} \qquad (2\text{-}8)$$

그러므로 weight의 변화는 다음식이 된다.

$$\Delta w_{mj}^k = \eta \cdot \delta_m^k \cdot o_j^k \qquad (2\text{-}9)$$

출력층의 경우 δ_m는 다음식과 같이 유도된다.

$$\delta_m^k = \frac{\partial E}{\partial o_m^{k+1}} \cdot \frac{\partial o_m^{k+1}}{\partial net_m}$$

$$= \frac{\partial E}{\partial o_m^{k+1}} \cdot f'(net_m)$$

$$= \tau_f \cdot f'(net_m) \qquad (2\text{-}10)$$

또한, 출력단 아래의 하위단의 상위층 PE_j와 하위층 PE_i사이의 δ_j는 다음과 같다.

$$\delta_j^{k-1} = \frac{\partial E}{\partial o_j^k} \cdot f'(net_j)$$

$$= \frac{\partial E}{\partial net_j} \cdot \frac{\partial net_j}{\partial o_j^k} \cdot f'(net_j)$$

$$= \frac{\partial E}{\partial net_j} \cdot \frac{\partial}{\partial o_j^k}\left(\sum w_{mj}^k \cdot o_j^k\right) \cdot f'(net_j)$$

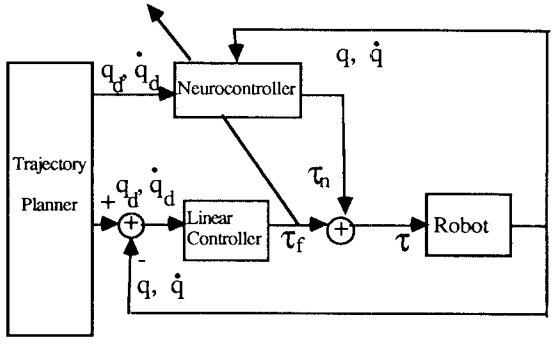
$$= \sum_{m=0} (\delta_m^k \cdot w_{mj}^k) \cdot f'(net_j) \qquad (2\text{-}11)$$

따라서, 출력단 이하의 하위층의 weight의 변화는 식 (2-12)가 된다.

$$\Delta w_j^{k-1} = \eta \cdot \sum_{m=0} (\delta_m^k \cdot w_{mj}^k) \cdot f'(net_j) \cdot o_i^{k-1} \qquad (2\text{-}12)$$

이때 학습출력에서의 오차는 출력층으로 부터 입력층까지 전파되면서 weight를 바꾸게 된다.

3. 신경제어 구조



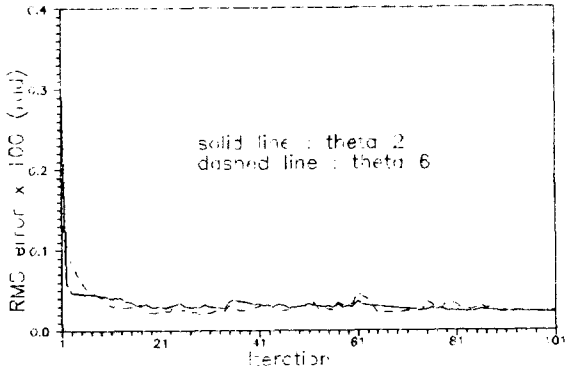
<그림 2> 신경제어 구조
<Fig. 2> Neurocontrol Architecture

이 신경회로를 로보트 제어에 응용하기 위해 <그림 2>와 같은 제어 구조를 제안한다. 우리가 원하는 경로가 주어지면 역기구학해를 풀어 각도와 각속도가 주어지고 현재 로보트의 각도와 각속도와의 차이에 의해 선형제어기 즉 PD Controller가 동작하여 궤환 토크 τ_f가 발생하고 신경제어기는 원하는 각도와 각속도, 현재 각도와 각속도를 입력으로 하는 다층 신경회로를 통과하여 Nominal Torque τ_n를 발생한다. 로보트는 두 Torque의 합으로 움직이게 된다. 이것은 순간적인 변화에서는 PD Controller가 주로 동작하게 하게 되고 평상시에는 신경제어기가 제어를 주로 담당하게 된다. 즉 급격한 변화에서는 신경회로의 학습할 시간이 없기 때문에 PD Controller를 병렬로 사용하게 된다. 학습 모드에서 신경제어기의 weight는 오차에 비례하도록 학습 시켜야 하므로 PD의 출력이 곧 신경회로의 학습신호가 된다. 이 구조의 장점은 Unsupervised Learning 방식으로 학습에 필요한 Torque 샘플을 제시할 필요가 없다. 또한 신경제어기의 학습 Algorithm과 구현이 간단하여 계산속도가 빠르게 된다. 또한 하나의 축의 오차가 신경회로의 내부연결에 의해 다른 축에 영향을 끼치게 되어 모든 축이 협동하여 오차를 줄이게 된다. 따라서 한축에 가해진 부하에 대한 적응 능력이 있고 오차의 수렴속도가 빠르다. 학습에 의한 능력으로 로보트의 Dynamic Model 없이 단지 오차만으로 제어가 가능하다. 이러한 특성을 보이기 위해 PUMA 560의 6축 Dynamic 제어를 모의 실험으로 보인다. 학습 모드에서는 Backpropagation rule에 의해 weight를 변화시키고, 제어 모드에서는 weight의 변화는 없고 학습에서 얻어진 weight로 제어출력을 계산한다.

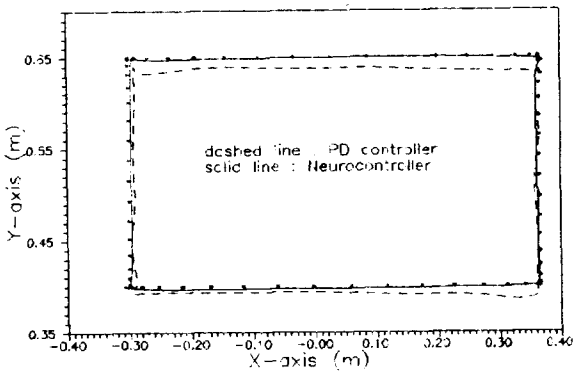
4. 실험 결과

4.1 일반적인 제어

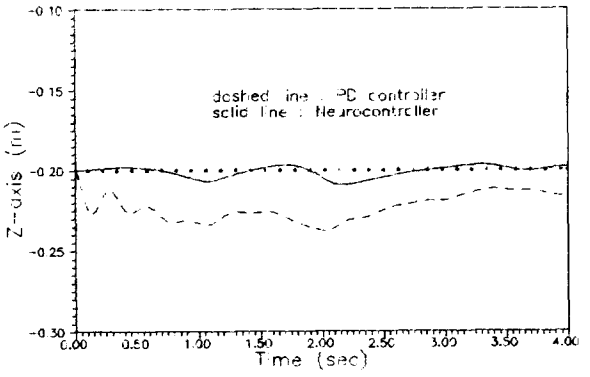
모의 실험은 주어진 사각형 경로를 움직여 신경제어와 일반적인 제어방법인 PD Control을 비교하는 것으로 하였다. 각 변을 움직이는 시간은 1초가 되고, 따라서 전체를 움직이는 시간은 4초가 된다. Sampling Time은 10msec로 하였다. <그림 3>은 학습 모드에서 시도에 따른 Joint 각도의 RMS 오차를 나타낸다. 학습이 계속될수록 오차가 계속 감소됨을 보여준다. <그림 4>는 100번 학습 이후에 제어 모드로 동작 시킨 결과와 PD Controller의 경우를 보인것이다. PD Controller의 경우 학습이 없으므로 여러번 시도해도 결과는 같다. 반면에 신경제어의 경우 계속적인 시도로 성능이 향상될 수 있다. 그림(a)는 X-Y축에서 볼때이고 그림(b)는 Z축과 시간축으로 본 경우이다. 점은 원하는 경로이고 실선은 로보트가 움직인 경로, 점선은 PD Controller가 움직인것을 나타낸다.



<그림 3> 학습에 따른 Joint 각도의 RMS 오차
<Fig. 3> RMS Joint Position Error versus Iterations At A Normal Speed



(a) X - Y 축 (on the X-Y Plane)

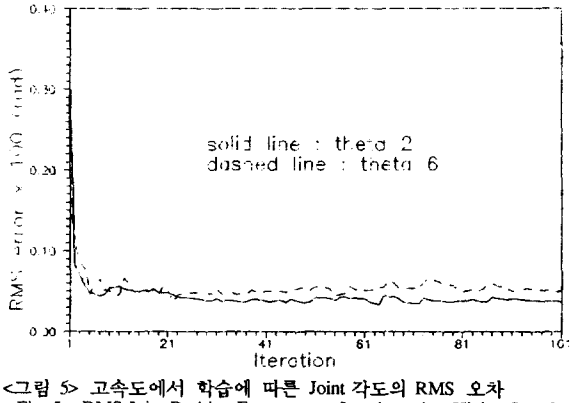


(b) Z축, 시간 축 (on the Z Plane with Time)

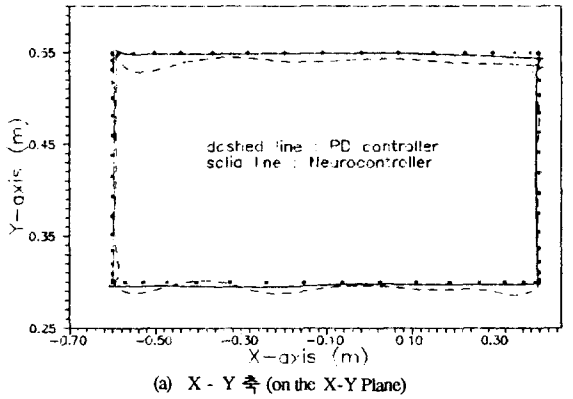
<그림 4> 신경제어와 PD제어 방식의 성능 비교
<Fig. 4> Performance Comparison of the PD and the Neurocontrol

4.2 고속 움직임에서 제어

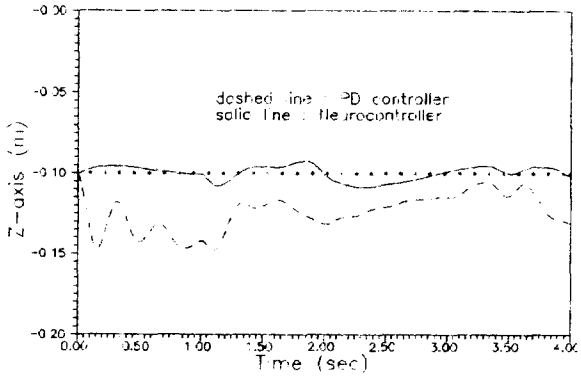
다음은 위에서 구한 PD Gain을 사용하여 최대 작업 영역내에서 고속도 움직임을 보인 것이다. <그림 5>는 학습 모드에서 RMS 오차를 나타낸다. 위의 실험 보다 완만하게 오차가 감소됨을 보여준다. <그림 6>은 PD Control 방법과 신경제어 방법을 비교한 것이다. PD Control방법이 오차가 증가함을 볼 수 있다. 그러나, 신경제어 방법은 100번 학습 이후 제어 모드에서 모든 부분의 오차가 충분히 감소됨을 보여준다.



<그림 5> 고속도에서 학습에 따른 Joint 각도의 RMS 오차
<Fig. 5> RMS Joint Position Error versus Iterations At a Higher Speed



(a) X - Y 축 (on the X-Y Plane)

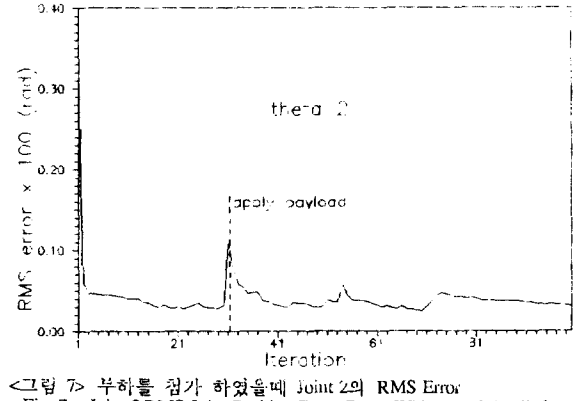


(b) Z축, 시간 축 (on the Z Plane with Time)

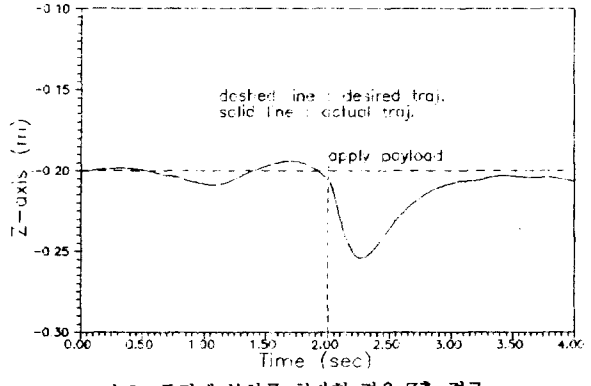
<그림 6> 고속도에서 신경제어와 PD제어 방식의 성능 비교
<Fig. 6> Comparison of the PD and the Neurocontrol At A Higher Speed

4.3 부하에 대한 적응 능력

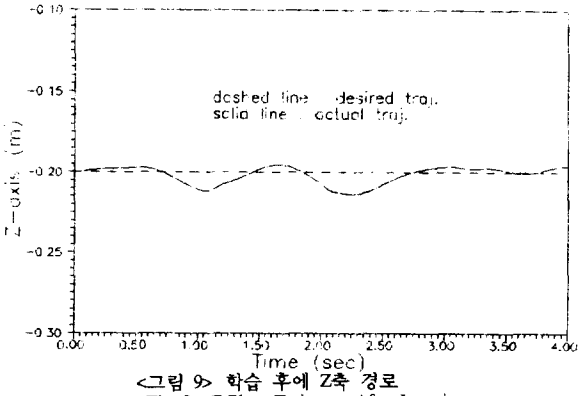
신경제어기를 30번째 학습중인 로보트에 중간 시점인 2초에 3Kg의 부하를 첨가 하였다<그림 7>(점선 표시). 따라서, 갑작스런 부하로 오차가 증가한다. 중력의 영향을 가장 많이 받는 Joint 2의 오차는 <그림 8>에서 같이 일시 증가 하지만 다시 학습을 하게 되어 감소하게 된다. <그림 9>은 100번 학습 이후에 제어 모드로 동작 시킨 경로이다. 부하가 보상된 것을 알 수 있다.



<그림 7> 부하를 첨가 하였을때 Joint 2의 RMS Error
<Fig. 7> Joint 2 RMS Joint Position Error Curve With a Load Applied



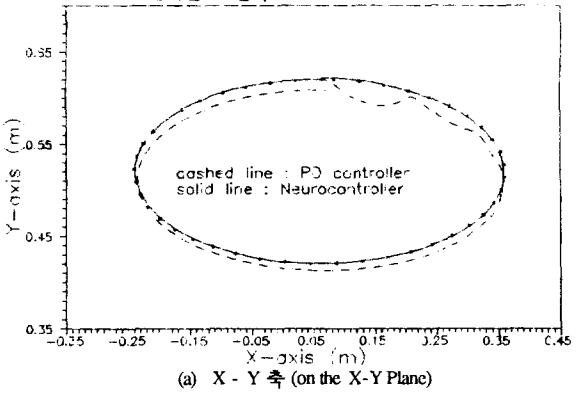
<그림 8> 중간에 부하를 첨가한 경우 Z축 경로
<Fig. 8> Z Position Trajectory With a Load Applied



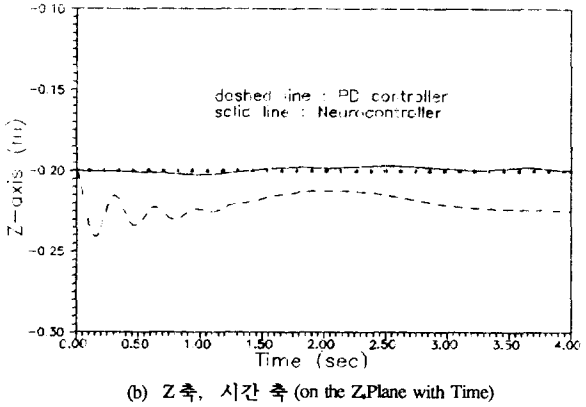
<그림 9> 학습 후에 Z축 경로
<Fig. 9> Z Plane Trajectory After Learning

4.4 일반화 (Generalization) 특성

신경제어기의 학습능력은 학습한 것만 기억하는 것이 아니라, System의 특성 자체를 학습으로 배우기 때문에 비슷한 일이 주어져도 일반화가 가능하다. 실험 4.1의 학습 모드에서 생긴 weight를 가지고 <그림 10>에서 보는것 같이 타원 모양의 경로를 주어졌을때 신경제어기의 제어 모드로 동작 시킨것이다. 한번도 배운적이 없는 경로이지만, weight에 로보트의 특성을 학습하였으므로 좋은 결과를 보인다.



(a) X - Y 축 (on the X-Y Plane)

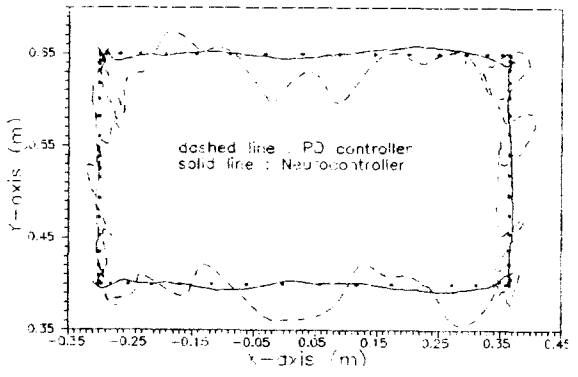


(b) Z축, 시간 축 (on the Z Plane with Time)

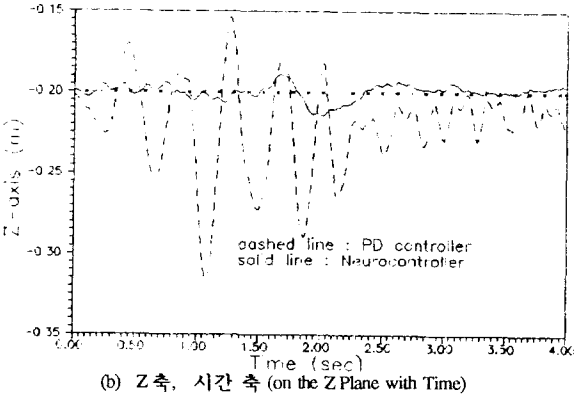
<그림 10> 신경제어와 PD제어의 일반화 특성 비교
<Fig. 10> Comparison of Generalization Characteristics

4.5 잡음 흡수 능력

신경제어기는 여러개의 PE가 병렬로 동작하기 때문에 잡음의 강건성이 생기게 된다. PD Control의 경우 약간의 잡음이 System을 크게 불안정하게 만들지만 신경제어의 경우 크게 영향을 받지 않는다. <그림 11>은 학습 모드에서 얻어진 weight를 가지고 잡음이 있는 상황에서 제어 모드로 동작 시킨것이다. 신경제어가 잡음에 대해여 강건함을 보여 준다. 여기서 잡음은 (실제값-0.1rad)에서 (실제값+0.1rad)까지의 pseudo random값을 의미한다.



(a) X - Y 축 (on the X-Y Plane)



(b) Z 축, 시간 축 (on the Z Plane with Time)

<그림 11> 신경제어와 PD제어의 잡음에 대한 강건성 비교
<Fig. 11> Comparison of Noise Robustness

5. 결론

위에서 신경회로망을 이용한 6축 로보트의 Dynamic 제어를 보였다. 신경제어는 자체가 가지고 있는 장점으로 학습과 병렬 처리가 가능하다. 따라서, 어떠한 작업도 학습으로 배우게 되고, 부하의 변화도 학습으로 제어가 가능하며, 학습으로 계속적인 성능 향상을 가져올 수 있다. 또한 간단한 알고리듬으로 구현하기가 쉽고, 잡음에 강건함을 가진다. 현재 개발 발표된 신경회로 Chip을 사용한다면, 위에서 보인 특성을 가진 경제적인 제어기의 개발도 머지 않아 이루어질 것이다.

참고 문헌

[1] 오세영, "신경회로의 로보틱스 및 산업 자동화 응용", *전자공학회지*, vol. 17, no. 3, 1990, 6월, pp. 28- 36.

[2] W. T. Miller, R. P. Hewes, F. H. Glanz, and L. G. Kraft, "Real time dynamic control of an industrial manipulator using a neural-network-based learning controller," *IEEE Trans. Robotics Automat.*, vol. 6, no. 1, Feb. 1990, pp 1-8.

[3] M. Miyamoto, M. Kawato, T. Setoyama, and R. Suzuki, "Feedback-error learning neural network for trajectory control of a robotic manipulation," *Neural Networks*, vol. 1, 1988, pp. 251-265.

[4] A. Guez and J. Selinsky,"Neurocontroller design via supervised and unsupervised learning," *J. Intelligent and Robotics Systems*,1989,pp. 307-335.

[5] D. E. Rumelhart, J. L. McClelland, and the PDP Reserch Group, *Parallel Distributed Processing*. MIT Press, vol. 1, 1986.

[6] M. Kawato, Y. Maeda, Y. Uno, and R. Suzuki, "Trajectory formation of arm movement by cascade neural network network model based on minimum torque-change criterion," *Biological Cybernetics*, vol. 62,1990, pp. 275-288.

[7] R. G. Hoptroff, T. J. Hall, and R. E. Burge, " Experiments with a neural controller", *Int. Joint Conf. on Neural Networks*, vol. 2, June, 1990, pp. 735-740.

[8] Yeon-Sik Ryu and Se-Young Oh, " A neural network architecture for dynamic control of robot manipulators", *Korean Auto. Control Conf.*, vol. 2, 1989, pp. 1113-1119.