

신경회로를 이용한 6축 Robot의 Dynamic Control

조 문 증^o 오 세 영
포항공대 전자전기공학과

Dynamic Control of A Six-link Robot Using Neural Networks

Moon-Jeung Joe and Se-Young Oh
Dept. of Electrical Engineering
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 unforeseen load changes. The results are compared with the conventional PD control scheme.

1. 서론

기존의 Robot Dynamic 제어에 있어서는 적당한 Dynamic Model에 대한 수치적 계산을 통한 제어를 하기 때문에 계속적으로 Dynamic 특성이 변화 하는 Robot의 경우 정확한 제어가 어려워 지고, 따라서 Robot의 제작도 이러한 문제를 생각하여 구조를 정밀하게 제작하여야 한다. 그러나 Robot의 정확한 Parameter를 구하는 것도 불가능 하지만 부품의 노후에서 오는 변화는 예측할수 없다. 특히, Kinematics만 고려한 현재의 Robot 제어 방식은 부하를 첨가하거나 고속동작의 경우 Dynamics를 고려한 제어보다 비효율적이라는 것은 잘 알려진 사실이다.

따라서 이러한 문제를 해결하기 위하여 신경회로를 응용하여 생물체에서 이루어지는 Sensory Motor Control를 Robot Control에 적용하고자 한다. 신경회로를 이용한 제어(신경제어)는 종래의 제어방식에서 볼 수 없는 다음과 같은 특징이 있다.

1) 신경제어는 모든계산이 각 뉴론(Neuron) 또는 PE(Processing Element)에서 독립적으로 병렬처리 된다. 따라서 실시간 제어가 가능하고 몇개의 PE의 고장으로 전체의 출력

에 별로 영향이 없다.

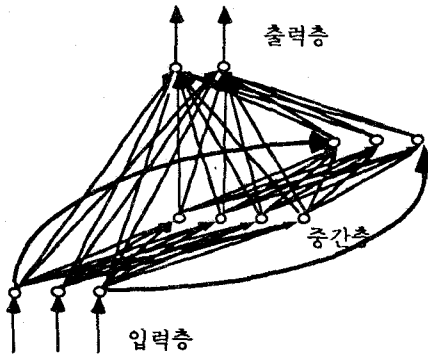
2) 신경제어는 단지 학습으로 제어방법을 배우기 때문에 System Model이 필요 없고 System Parameter가 변화 하여도 적응 능력이 있다. 따라서, 복잡한 System의 제어도 가능하다. 계속적인 학습은 Robot의 성능을 개선 시킨다.

현재까지 발표된 신경회로를 이용한 Robot의 Dynamic 제어는 아직 많지 않다. 그중 Miller가 Table Lookup 방식의 CMAC을 이용한 5축 로봇의 Dynamic Control이 있고[1], Miyamoto가 모의실험으로 3축 로봇의 Dynamics에 관계하는 Newton Euler Term을 입력으로 신경 회로가 제어하고 PD Controller로 학습하는 Feedback-Error-Learning이 있다[2]. Guez는 2축 Robot의 Dynamics Model을 Backpropagation Network이 학습을 통해 알아내는 방식으로 제어하는것을 보였다[3]. CMAC 방식은 많은 기억 용량이 필요한 것이 단점이고 Feedback-Error-Learning과 Guez는 어느 정도의 Robot Dynamics에 대한 Term을 알아야 한다는 단점이 있다. 그리고 산업용 로봇의 적용하기 위해서는 6축 제어를 필요로 한다.

본 논문에서는 처음으로 6축 로봇의 Dynamic 제어를 시도하였다. 로봇을 완전한 Black Box로 가정하고 BP network (Backpropagation Network 또는 Multilayer Perceptron) 과 PD Controller를 병렬로 사용하여 Feedback Error를 줄이는 방향으로 신경회로를 학습 시킨다. 신경회로는 원하는 Joint 각도-각속도와 실제 각도-각속도만을 받아 필요한 Joint Torque를 발생한다. 모의 실험결과 고속제어와 부하 적용에서 좋은 결과를 얻었다.

2. 다층 신경회로

지금까지 발표된 많은 신경회로 모델이 있지만 제어 문제에 적용할 수 있도록 Nonlinear, Continuous Mapping에 적합한 모델은 Multilayer Perceptron이다[4, 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 신경회로를 사용한다(그림. 1).



<그림 1> Full connection된 다층 신경회로

w_j 를 PE_i (하위층 i 번째 PE)에서 PE_j (상위층 j 번째 PE)로의 weight라 할 때 PE_j 의 출력은 다음과 같이 표시된다.

$$net_j = \sum_{i=1}^n (w_{ji} \cdot o_i) \quad (2-1)$$

$$o_j = f(net_j) \quad (2-2)$$

여기서 함수 $f(\cdot)$ 는 활성화함수(Activation Function)라 하고, 단조 증가함수이면서 미분가능한 함수이어야 한다. 보통의 경우 Sigmoid 함수를 사용한다.

식 (2-2) 의 출력은 다음층으로 이동하게 되고 중간단 PE에서의 비선형 변환에 의한 특성으로 임의의 입력력에 대한 비선형 Mapping이 가능하게 된다. 제어 문제와 같이 적당한 출력의 범위를 알수 없을 경우 출력층의 활성화함수는 적당한 기울기를 가지는 선형함수를 사용할수 있다.

다층 신경회로의 학습은 Backpropagation rule(BP rule)에 의해 실행된다[4]. BP rule은 여러 입력 샘플에 대한 원하는 출력을 가르쳐주는 Supervised 학습 방법에 의하여 원하는 출력으로 변환되도록 한다. 학습은 원하는 출력과 실제 출력의 차이를 줄이는 방향으로 한다. 따라서, Error가 감소하도록 weight를 바꾸는 것은 Gradient Descent Rule을 사용한다. 따라서, 임의의 PE_k 와 PE_j 사이의 weight의 변화는

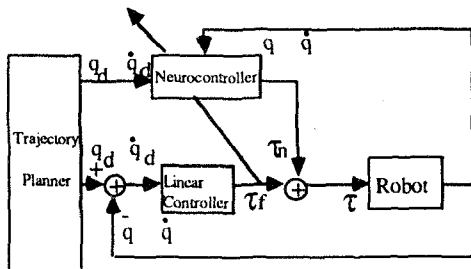
$$\Delta w_{kj} = \eta \cdot f'(net_k) \cdot o_j \cdot (d_k - o_k) \quad (2-3)$$

이다. 또한 출력단 아래의 하위단의 weight의 변화는

$$\Delta w_j = \eta \cdot f'(net_j) \cdot o_j \cdot \sum_k (d_k - o_k) \cdot w_{kj} \quad (2-4)$$

가 된다.

3. 신경제어 구조



<그림 2> 신경제어 구조

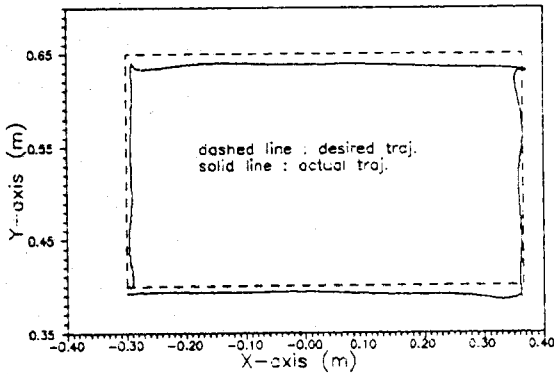
이 신경회로를 로봇트 제어에 응용하기 위해 <그림 2>와 같은 제어 구조를 제안한다. 우리가 원하는 경로가 주어지면 역기구학해를 풀어 각도와 각속도가 주어지고 현재 로봇트의 각도와 각속도의 차이에 의해 신경제어기 즉 PD Controller가 동작하여 τ_n 가 발생하고 신경제어기는 원하는 각도와 각속도, 현재 각도와 각속도를 입력으로 하는 Backpropagation Network를 통과하여 τ_n 를 발생한다. 로봇트는 두 Torque의 합으로 움직이게 된다. 이것은 순차적인 변화에서는 PD Controller가 주로 동작하게 하게 되고 평상시에는 신경제어기가 제어를 주로 담당하게 된다. 즉 급격한 변화에서는 신경회로의 학습할 시간이 없기 때문에 PD Controller를 병렬로 사용하게 된다. 그러나 이때에도 신경제어기가 동작하고 있어 급격한 상황에 변화에서도 System이 불안정하게 되는것을 방지한다. 신경제어기의 학습은 오차에 비례하도록 학습 시켜야 하므로 PD의 출력이 곧 신경회로의 학습신호가 된다. 이 구조의 장점은 Unsupervised Learning에 의해 학습에 필요한 Torque 샘플을 제시할 필요가 없고, 또한 로봇트의 Jacobian을 계산할 필요도 없다. Adaptive Algorithm이 간단하여 구현이 간단하고 계산속도가 빠르게 된다. N축의 경우 첫번째 중간층에 20개 PE, 두번째 중간층에 10개 PE의 경우 출력을 계산하기 위해 $4N^2 + 150N + 200$ 번의 덧셈과 곱셈이 필요하다(6축 경우 1244번). 또한 하나의 축의 오차가 다른 축에 영향을 끼치게 되어 모든 축이 협동하여 오차를 줄이게 된다. 따라서 부하에 대한 적응 능력이 있고 오차의 수렴속도가 빠르다. 학습에 의한 능력으로 Robot의 Dynamic Model 없이 단지 오차만으로 제어가 가능하다. 이러한 특성을 보이기 위해 PUMA 560의 6축 Dynamic 제어를 모의 실험으로 보인다.

4. 실험 결과

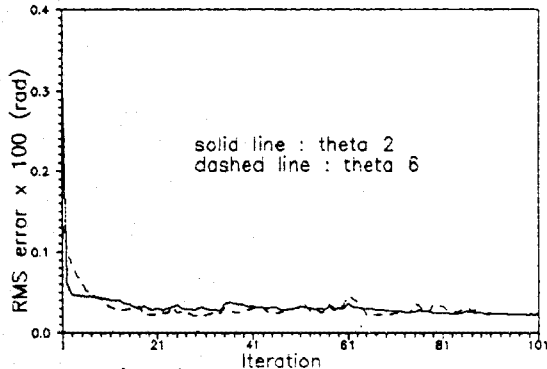
모의 실험은 Z축을 일정하게 유지하면서 XY 평면에 주어진 사각형 경로를 움직여 신경제어와 일반적인 제어방식인 PD Control를 비교하는 것으로 하였다.

4.1 일반적인 제어

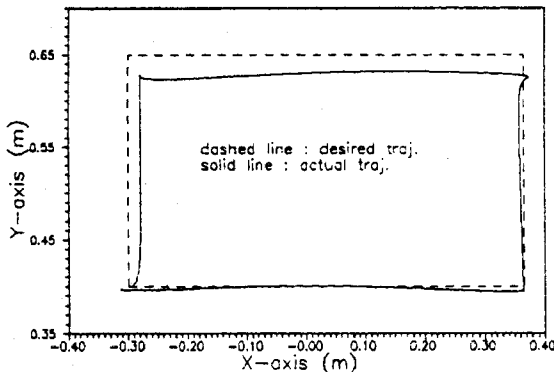
먼저 보통 속도로 움직일때 가장 좋은 결과가 나온 PD Gain을 구하였다. <그림 3>은 X-Y축에서 볼때 PD Controller만 동작할 경우이다. 점선은 원하는 경로이고 실선은 로봇트가 움직인 경로를 나타낸다. 각 변을 움직이는 시간은 1초가 되고, 따라서 전체를 움직이는 시간은 4초가 된다. Sampling Time은 10msec로 하였다. PD Controller의 경우 학습이 없으므로 여러번 시도해도 결과는 같다. 반면에 신경제어의 경우 계속적인 시도로 성능이 향상됨을 볼 수 있다. <그림 4>는 시도에 따른 Joint 각도의 Root Mean Square Error를 나타낸다. 학습이 계속될수록 오차가 계속 감소됨을 보여준다. <그림 5>는 1번째 시도한 경우이고 <그림 6>은 101번째 시도한 경우이다. 학습 후 성능이 향상됨을 알 수 있다.



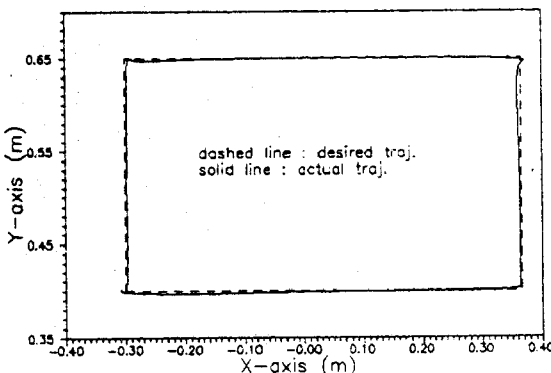
<그림 3> PD Controller만 동작할 경우(X-Y축)



<그림 4> 시도에 따른 Joint 각도의 RMS Error



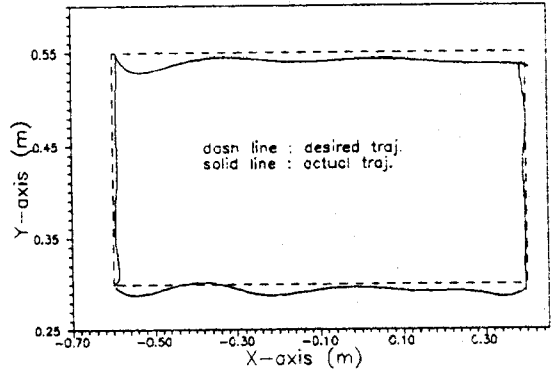
<그림 5> 1번째 시도한 경우



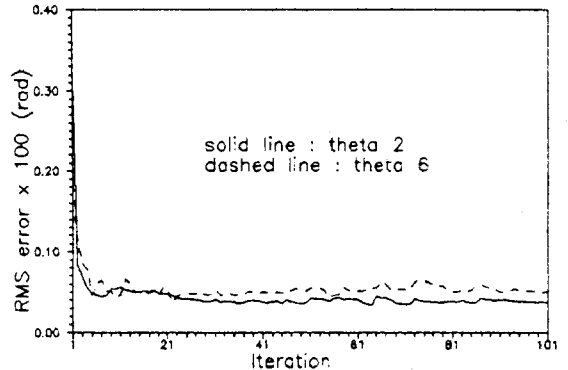
<그림 6> 101번째 시도한 경우

4.2 고속 움직임에서 제어

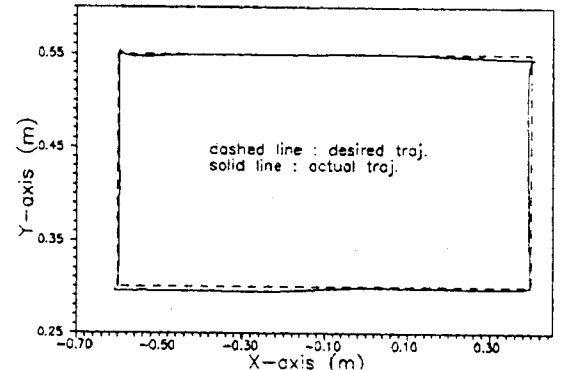
다음은 위에서 구한 Gain을 사용하여 최대 작업 영역내에서 고속도(평균속도 1m/sec) 움직임을 보인 것이다. <그림 7>은 PD Control방식이다. 오차가 증가함을 볼 수 있다. <그림 8>은 Joint 각도의 Error를 나타낸다. 위의 실험 보다 완만하게 오차가 감소됨을 보여준다. <그림 9>은 101번째 시도한 경우이다. 모든 부분에서 오차가 충분히 감소됨을 보여준다.



<그림 7> PD Controller만 동작할 경우(X-Y축)



<그림 8> 시도에 따른 Joint 각도의 RMS Error

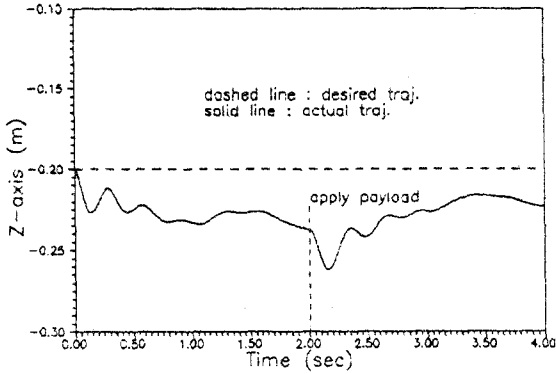


<그림 9> 101번째 시도한 경우

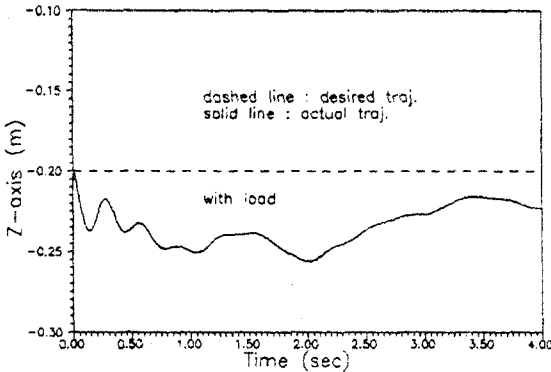
4.3 부하에 대한 적응 능력

신경제어기가 갑자기 생긴 부하에 대한 적응을 시험하여 본다. 먼저 PD control방식의 성능을 보면 로봇트가 움직이는 중간 시점인 2초에 3Kg의 부하를 첨가 하였다. <그림 10>는 Z축과 시간축으로 본 경우이다. 점선 표시한 곳이 부하를 첨가한 순간이다. 그리고, 부하를 단 상태에서 다시 움직였다 <그림 11>. 결과에서 처럼 PD control방식은 부하에 대한 적

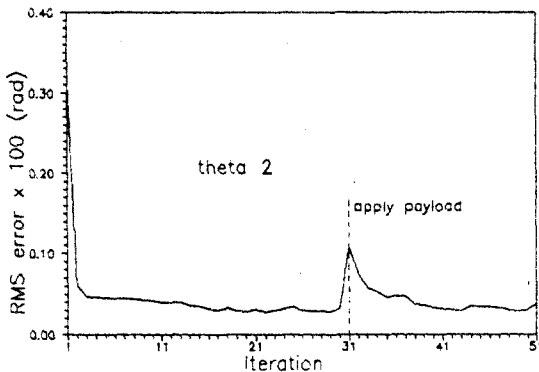
은 능력이 거의 없다. 이번에는 신경제어기를 31번의 학습 후에 부하를 첨가 하였다. 중력의 영향을 가장 많이 받는 Joint 2의 오차는 <그림 12>에서 같이 일시 증가 하지만 다시 학습을 하게 되어 감소하게 된다. <그림 13>는 부하를 첨가한(점선 표시) 31번째 이고 <그림 14>은 51번째 경로이다. 부하가 보상된 것을 알 수 있다.



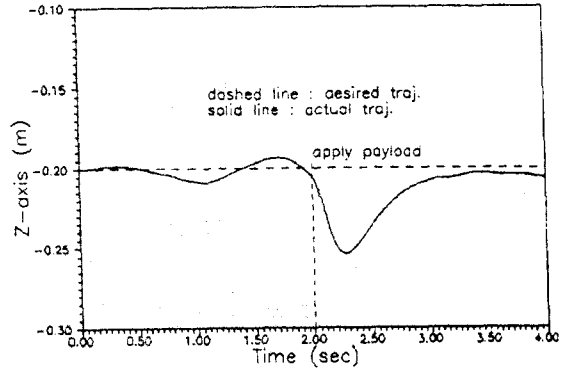
<그림 10> PD 방식 중간에 부하를 첨가 (Z축, 시간축)



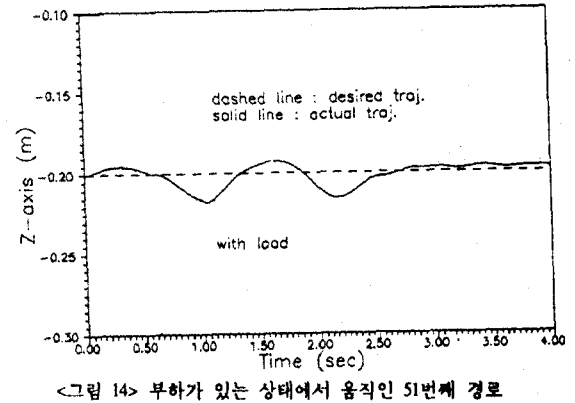
<그림 11> PD 방식에서 부하가 있는 상태에서 움직인 경로



<그림 12> 부하를 첨가 하였을때 Joint 2의 RMS Error



<그림 13> 중간에 부하를 첨가한 31번째 경로



<그림 14> 부하가 있는 상태에서 움직인 51번째 경로

5. 결론

위에서 신경회로망을 이용한 6축 로봇 제어기를 보였다. 신경회로망은 자체가 가지고 있는 장점으로 병렬 처리와 학습이 가능하다. 또한 간단한 알고리즘으로 실시간 제어가 가능하다. 또한 부하에 관계 없이 학습으로 제어가 가능하며 학습으로 계속적인 성능 향상을 가져올 수 있다.

참고 문헌

- [1] 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
- [2] 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.
- [3] A. Guez and J. Selinsky, "Neurocontroller design via supervised and unsupervised learning," *J. Intelligent and Robotics Systems*, 1989, pp. 307-335.
- [4] D. E. Rumelhart, J. L. McClelland, and the PDP Reserch Group, *Parallel Distributed Processing*, MIT Press, vol. 1, 1986.
- [5] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison Wesley, 1989.