

신경 회로망에 의한 로봇 매니플레이터의 PTP 운동에 관한 연구

경계현, 고명삼, 이발희
서울대학교 제어계측공학과 로봇릭스 및 지능시스템 연구실

(A Study on the PTP Motion of Robot Manipulators by Neural Networks)

Kye Hyun Kyung, Myoung Sam Ko, Bum Hee Lee
Robotics and Intelligent Systems Lab.
Dept. of Control and Instrumentation Engineering
Seoul National University

Abstract

In this paper, we describe the PTP motion of robot manipulators by neural networks. The PTP motion requires the inverse kinematic routine and the joint trajectory generation algorithm. We use the multi-layered Perceptron neural networks and the Error Back Propagation(EBP) learning rule for inverse kinematic problems. Varying the number of hidden layers and the neurons of each hidden layer, we investigate the performance of the neural networks. Increasing the number of learning sweeps, we also discuss the performance of the neural networks. We propose a method for solving the inverse kinematic problems by adding the error compensation neural networks(ECNN). And, we implement the neural networks proposed by Grossberg et al. for automatic trajectory generation and discuss the problems in detail. Applying the neural networks to the current trajectory generation problems, we can reduce the computation time for trajectory generation.

1. 서론

로봇 매니플레이터의 PTP 운동은 작업 공간상에서 이동할 목표 위치가 주어지면 중간 경로에 구애받지 않고 현재 로봇의 위치로부터 목표위치까지를 관절 궤적에 의해서 운동하는 동작을 말한다. 그림1은 로봇 매니플레이터의 PTP 운동 방식을 나타낸다. 즉 공간상의 로봇의 이동 목표점의 좌표값이 주어지면 이 값은 역기구학을 통하여 목표 관절값으로 변환되고 관절 궤적 발생기는 이 값과 현재의 관절값을 이용하여 로봇이 목표 지점에 도달하도록 하는 관절 궤적을 여러 제한(Constraints) 하에서 발생한다.

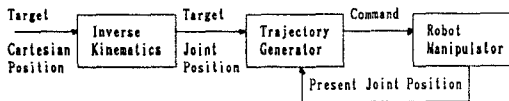


그림1. 로봇 매니플레이터의 PTP 운동

현재 사용되고 있는 로봇의 PTP 운동 방식은 로봇의 기구학적 구조에 따라 미리 정해진 역기구학 문제의 해결과 여러제한 조건 속에서 미리 계산된 궤적 형성을 필요로 한다.[1] 이와 같은 기존의 방식들은 다음의 문제점들을 가진다.

- 1) 로봇에 대한 정확한 기구학적 특성을 알아야 한다.
- 2) 역기구학 문제나 관절 궤적 발생 문제를 해결하는데 사용되는 수식의 수가 로봇 매니플레이터의 자유도(DOF)에 따라 증가한다.
- 3) 역기구학 해가 closed form으로 존재하지 않고 iterative form으로 존재하는 경우에 많은 계산 시간이 소요된다.
- 4) 각 로봇의 역기구학 해는 단지 그 로봇 모델 하나에 대해서만 유효하다.

한편, 생형 시스템은 그들의 구조와 기능들을 적응적으로 학습할

수 있는 유연성(flexibility)과 성형성(plasticity)을 갖는데, 이것은 생명체의 여러 조직(신경, 근육, 뼈)들 간의 상호관계와 상호작용이 스스로 조직하는(self-organizing) 현상을 나타내기 때문이다. 신경 회로망(neural networks)은 생명 시스템에서와 같이 스스로 조직하는 방법으로 정보를 처리하도록 인공적으로 개발된 시스템이다[2]. 이와 같은 신경 회로망은 다음의 특징들을 갖는다.[3]

- 1) 신경 회로망은 표본 데이터에 의해 학습된다. 따라서 프로그램의 작업이 제거될 수 있다.
- 2) 신경 회로망은 보간(interpolation) 기능을 갖는다.
- 3) 신경 회로망의 학습이 끝난 뒤 주어진 문제에 대한 해를 구하는데 요구되는 시간이 로봇의 자유도에 무관하다.
- 4) 신경 회로망의 self organizing 성질은 시스템 자체나 외부 환경에서 발생하는 변화에 적응 능력을 갖게 한다.
- 5) 신경 회로망은 fault tolerance 특징을 갖는다.

이러한 신경 회로망의 특징들은 신경 회로망으로 역기구학 문제와 관절 궤적 형성을 수행하면 앞에서 열거한 기존 방식의 문제점들이 해결될 수 있음을 보인다.

본 논문에서는 신경 회로망을 사용한 로봇 매니플레이터의 PTP 운동을 수행하기 위하여 서로 다른 구조의 신경 회로망을 사용한 역기구학 문제와 관절 궤적 발생기에 관한 연구 결과를 서술한다.

신경 회로망을 이용한 로봇의 역기구학 문제의 해결에 관한 연구로는 다중구조의 Perceptron 신경 회로망을 사용하고 Error Back Propagation(EBP) 학습 방법[7]으로 학습을 수행한 경우와[2,3], CMAC[5]를 사용한 경우[4]등이 있다. 위의 두 종류의 신경 회로망이 역기구학 문제의 해결과 같은 문제에서 사용될 수 있는 이유는 이들이 보간 능력을 갖기 때문이다. Josin[2]은 주로 학습 점의 수에 대한 역기구학 문제 해결의 성능 분석을 수행하였으며, Guez et al.[3]은 2DOF의 로봇과 3DOF의 로봇에 대한 역기구학 문제의 해결을 같은 구조의 신경 회로망으로 수행하여 로봇의 기구학적 구조의 복잡성에 관계없이 비슷한 성능을 수행함을 보였으며 또한 현재 사용되고 있는 역기구학 해답이 갖는 문제점을 제시하고 신경 회로망을 사용한 역기구학 문제 해결의 장점에 대해 서술하였다. Albus[5]는 CMAC를 이용한 로봇의 실시간 시각 추적에 관하여 연구하였다. 본 논문에서는 로봇의 역기구학 문제의 해결을 다중구조의 Perceptron 신경 회로망에 EBP 학습 방법을 통하여 수행하였으며, 신경 회로망의 중간층의 수와 각 중간층 신경 세포(neuron)의 수에 따른 성능 평가와 학습 회수에 따른 성능 평가를 수행하였다. 또한 Josin et al.[6]이 제안한 것과 같은 구조 오차 보정 신경 회로망(Error Compensation Neural Net: ECNN)을 추가하여 역기구학 문제 해결의 성능 향상을 보였다.

Grossberg등은 목표점만 주어지면 자동적으로 관절 궤적을 형성하는 신경 회로망을 제안하였다[8,9]. 이 신경 회로망에 의한 관절 궤적 형성은 다관절이 서로 다른 거리를 이동하여도 각 관절의 이동 목표점만 주어지면 동시에 운동을 마치도록 이루어지며, 미리 계산할 아무 것도 필요로 하지 않는다. 본 논문에서는 이 구조의 신경 회로망을 분석하고, 현재 로봇에서 사용하고 있는 관절 궤적 형성에 이 구조의 신경 회로망의 응용을 통한 이점에 대해 설명한다.

2장에서는 EBP 신경 회로망을 사용한 공간 좌표계상의 목표점을 목표 관절값으로 바꾸어주는 역기구학 문제 해결에 대해 서술하고, 3장에서는 역기구학 신경 회로망의 출력인 목표 관절값과 로봇의 현재 관절값을 입력받아 로봇의 관절 궤적을 출력하는 신경 회로

망에 대하여 서술한다. 4장에서는 결론은 제시한다.

2. 신경 회로망에 의한 역기구학 문제의 해결

2.1. 신경 회로망

본 논문에서 로봇의 역기구학 문제의 해결에 사용한 신경 회로망은 다층구조의 Perceptron으로 EBP 학습 방법으로 학습을 수행하였다. 각 신경 세포의 Activation function은 Sigmoid 함수를 갖도록 하였다. 시뮬레이션에서 학습율(learning ratio)은 모두 0.2로 하였으며, 모우멘텀은 0.9로 하였다.

2.2. 학습 방법

EBP 학습 방법은 지도 학습이다. 따라서 이 방법에 의한 역기구학 문제의 학습을 위해서는 공간 좌표계의 한 점의 좌표값과 그에 대응하는 관절 좌표계의 좌표값의 쌍들이 필요하다. 이는 로봇에 대한 기구학적 또는 역기구학적 구조를 미리 알아야 함을 의미한다. 그러나 그림2와 같이 로봇의 공간 좌표계의 좌표값을 감지할 수 있는 센서를 사용하면 EBP 학습 방법을 자율 학습 방법과 같이 사용할 수 있다. 즉 먼저 RMG(Random Movement Generation)에서 로봇이 이동할 관절값을 발생하면 로봇은 그 값에 따라 이동하게 되고 이때 로봇의 공간 위치는 센서를 통해 감지되어 역기구학 신경 회로망의 입력이 된다. 학습 법칙 블록은 RMG의 출력값과 신경 회로망의 출력값을 입력 받아 그 오차에 따라 역기구학 신경 회로망을 학습시킨다.

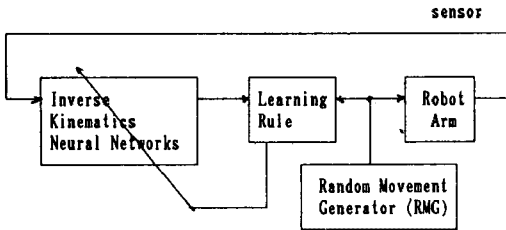


그림2. 역기구학 신경 회로망의 스스로 조직하는 학습 방법

한편 그림2의 방법에 의한 학습은 실제 로봇들을 구동하여야 하며, 로봇의 공간 좌표값을 감지하는 센서가 필요하다. 본 논문에서는 작업을 간단히 하기 위하여 로봇의 역기구학적 구조를 안다고 가정하고 그림3과 같이 학습을 수행하였다. 즉 로봇의 공간 위치값이 역기구학 신경 회로망과 IKS(Inverse Kinematics Solution) 시스템에 동시에 입력되면, 학습 법칙 블록은 각 블록의 출력값 사이의 오차에 따라 신경 회로망을 학습시킨다.

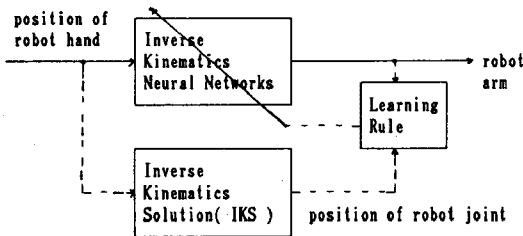


그림3. 역기구학 신경 회로망의 학습 방법

신경 회로망에 의한 역기구학 문제의 해결은 첫째 학습 점의 제한, 둘째 수렴의 한계등의 이유로 오차를 포함하게 된다. 본 논문에서는 그림4와 같이 오차 보정 신경 회로망(ECNN)을 추가하여 그 오차를 줄이도록 하였다. ECNN은 로봇의 공간 위치값과 역기구학 신경 회로망의 출력값을 입력으로 하며 오차 보정값을 출력한다. ECNN의 학습은 그림4와 같이 IKS의 출력과 학습이 끝난 역기구학 신경 회로망의 출력사이의 오차와 ECNN의 출력사이의 오차에 따라 수행한다.

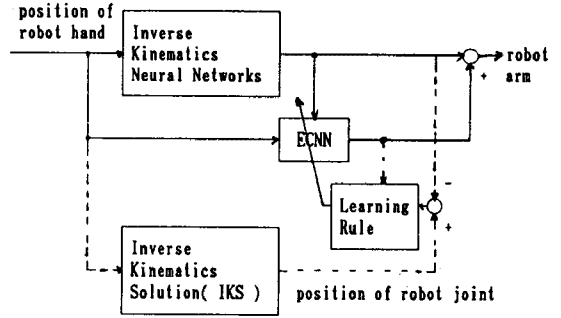


그림4. ECNN의 역할과 학습 방법

그림3과 그림4에서 점선으로 표시된 부분은 역기구학의 학습 기간에만 필요한 부분으로 학습이 끝난 뒤에는 제거된다.

2.3. 시뮬레이션 결과

본 논문에서 신경 회로망에 의한 역기구학 문제 해결의 시뮬레이션에 사용한 대상 로봇은 2DOF를 가지며 기구학적 구성은 그림5와 같다. 이 로봇의 기구학과 역기구학 해답은 식(1)과 (2)와 같다.

$$\begin{aligned}
 x &= r_1 \cos(\theta_1) + r_2 \cos(\theta_1 + \theta_2) & (1-1) \\
 y &= r_1 \sin(\theta_1) + r_2 \sin(\theta_1 + \theta_2) & (1-2) \\
 \theta_1 &= \text{atan2}(y/x) - \arccos(k_1/k_2) & (2-1) \\
 \theta_2 &= \pi - \arccos(k_3/k_4) & (2-2) \\
 k_1 &= x^2 + y^2 - (r_2^2 + r_1^2) & (2-3) \\
 k_2 &= 2 r_1 \sqrt{x^2 + y^2} & (2-4) \\
 k_3 &= r_1^2 + r_2^2 - x^2 - y^2 & (2-5) \\
 k_4 &= 2 r_1 r_2 & (2-6)
 \end{aligned}$$

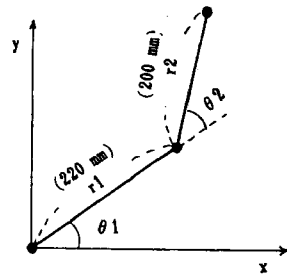


그림5. 시뮬레이션에 사용된 대상 로봇

역기구학 문제 해결을 위한 작업 공간은 그림6과 같이 전체 로봇의 작업 공간의 일부를 선택하였으며, 학습을 수행하기 위한 학습점은 그림6의 주어진 작업 공간에서 그림7에서와 같이 28점과 81 점을 선택하였다.

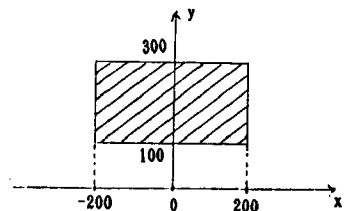


그림6. 역기구학 문제 해결에 사용된 작업 공간

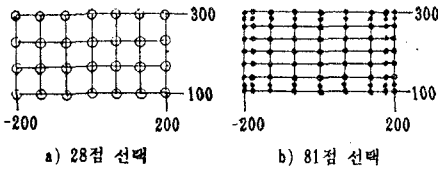


그림7. 학습점의 선택

신경 회로망에 의한 역기구학 문제 해결을 위한 시뮬레이션은 신경 회로망의 중간층(hidden layer)의 수를 한 층과 두 층으로 변화시키며 또 각 층의 신경 세포의 수를 다양하게 변화시키며 각 회로망 구조에 대한 오차조사에 대해 수행 하였으며, 또한 학습 회수에 따른 오차의 변화에 대해서도 조사하였다.

그림8은 28점의 학습점에 대한 다양한 회로망 구조에 따른 학습 결과이다. 이때 학습 회수는 각 회로망 구조에 대해 100000번씩 수행하였으며, 오차는 작업 공간에서 14641점을 선택하여 구하였다. 그림8의 a)는 중간층이 하나이고 중간층의 신경세포가 75개인 회로망에 대한 결과를 나타내며, 그림8의 f)는 중간층이 두개인 회로망에 대한 결과를 나타내며, 각각 4개인 경우를 나타낸다. Mean Square Error(MSE)는 식(3)과 같이 정의하였으며, 최대 오차(MAX_ERR)는 식(4)와 같이 정의하였다.

$$MSE = \sum \text{sqr}((x_d - x_n)^2 + (y_d - y_n)^2) / N \quad (3)$$

$$MAX_ERR = \max \{ \text{sqr}((x_d - x_n)^2 + (y_d - y_n)^2) \} \quad (4)$$

위의 식에서 (x_d, y_d)는 목표 공간 좌표점을 의미하며 (x_n, y_n)은 신경 회로망의 출력에 의한 로봇의 이동점을 의미한다. N은 시뮬레이션 점의 수이다. 그림8의 결과를 통해 중간층이 2개이고 각 중간층의 신경 세포의 수가 각각 5개인 신경 회로망이 역기구학 문제 해결에 가장 좋은 성능을 보임을 알 수 있다. 81점의 학습점에 대한 시뮬레이션도 같은 결과를 보였다.

INVERSE KINEMATIC USING BP

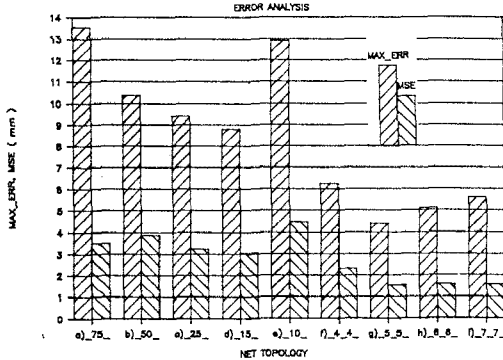
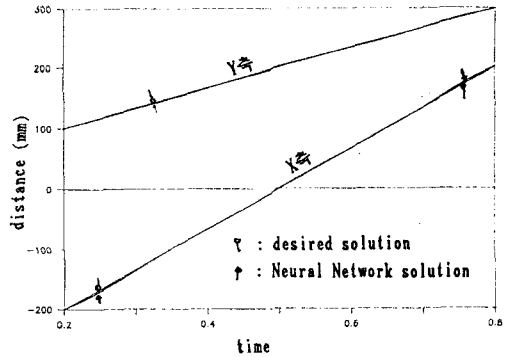


그림8. 신경 회로망 구조에 따른 오차 분석

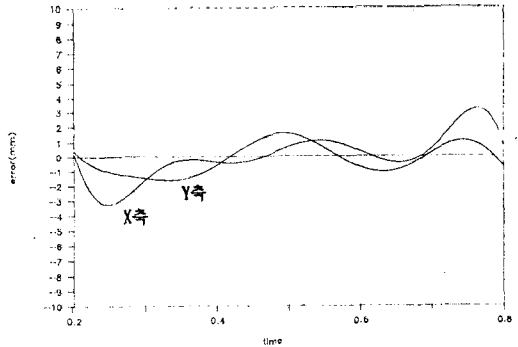
그림9는 28학습점으로 학습시킨 5.5 구조(중간층이 두개이고 각 중간층의 신경 세포의 수가 5개인 경우)의 신경 회로망을 사용하여 작업 공간에서 X-Y의 직선에 대해 역기구학 문제를 해결한 결과를 나타내며, 그림10은 81학습점으로 학습시킨 경우에 대한 같은 실험 결과를 나타낸다. 이들로부터는 학습점이 증가할수록 신경 회로망의 성능이 향상됨을 알 수 있다.

그림11은 81학습점으로 학습시킨 경우 학습 회수에 따른 작업 공간상에서 역기구학 문제 해결의 오차의 변화를 나타낸다. 그림11로부터 MSE는 학습 회수에 따라 감소함을 볼 수 있으나, 최대 오차는 학습에 따라 감소하다 학습이 진행될수록 다시 증가함을 볼 수 있다. 이것은 학습 회수가 증가함에 따라 학습점 주변의 점들에 대해서는 신경 회로망의 출력이 원하는 값으로 수렴하므로 학습 점들 사이의 점들에 대해서는 부드러운 보간 효과를 나타내기 위해 상대적으로 큰 오차를 나타내기 때문이다.

그림12는 5.5 구조의 신경 회로망에 대해 81학습점으로 100000 번 학습시킨 역기구학 신경 회로망에 그림4와 같이 ECNN을 추가한 경우 학습 회수에 대한 오차의 변화를 나타낸다. 그림11에서와 마찬가지로 MSE는 학습 회수에 따라 감소함을 볼 수 있으며 최대



a) 로봇 핸드의 운동



b) 경로에 대한 오차

그림9. 5.5 신경 회로망을 사용한 역기구학 시뮬레이션 결과 (학습점 : 28)

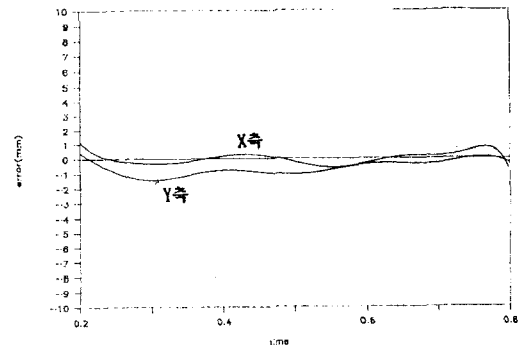


그림10. 5.5 신경 회로망을 사용한 역기구학 시뮬레이션 결과 (학습점 : 81)

오차는 학습에 따라 감소 하다. 30000번을 넘어서면서 다시 증가함을 볼 수 있다. ECNN의 학습점은 그림7의 b)와 같은 81점으로 하였다. ECNN을 추가하여 30000번 학습시킨 경우의 오차는 ECNN이 없는 경우에 비해 MSE는 약 1/2로 최대 오차는 약 1/3로 감소함을 볼 수 있다. 그림13은 ECNN을 추가하여 30000번 학습시킨 경우와 ECNN이 없는 경우 각각에 대해 작업 공간에서 X-Y의 직선에 대한 역기구학 문제 해결의 결과를 나타내며 ECNN을 추가한 경우 오차가 상당히 줄어들음을 볼 수 있다.

3. 신경 회로망에 의한 관절 제척 형성

Grossberg 등[8,9]은 그림14와 같은 구조의 자동 제척 형성을 위한 신경 회로망을 제안하였다. 이 구조의 신경 회로망은 목표 명령(Target Position Command: TCP)만 주어지면 운동의 전 속도 프로파일을 관찰하는 제어함수 GO 명령에 의하여 자동적으로 제척을 형성하게 되는데, 다만절이 각각 서로 다른 거리를 이동하여도 주어진 GO 명령에 따라 같은 형태의 운동 프로파일을 가지며 동시에 운동을 마치는 제척을 형성하도록 한다.

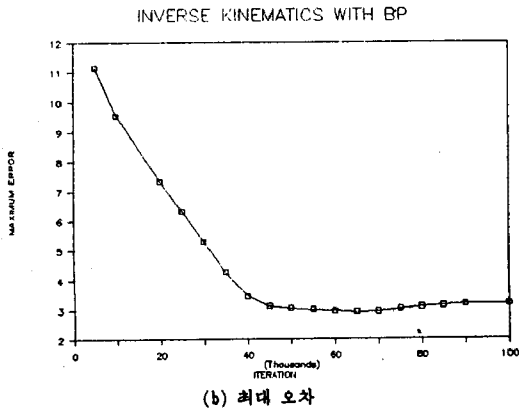
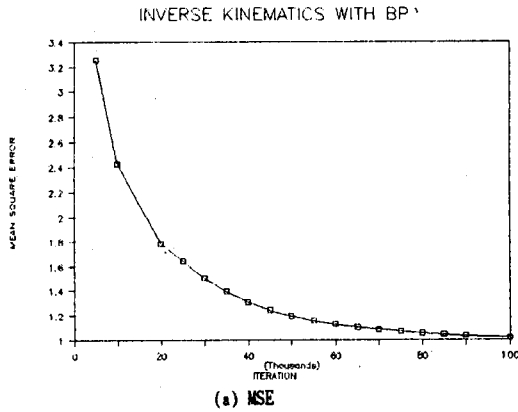


그림11. 학습 회수에 따른 오차의 변화

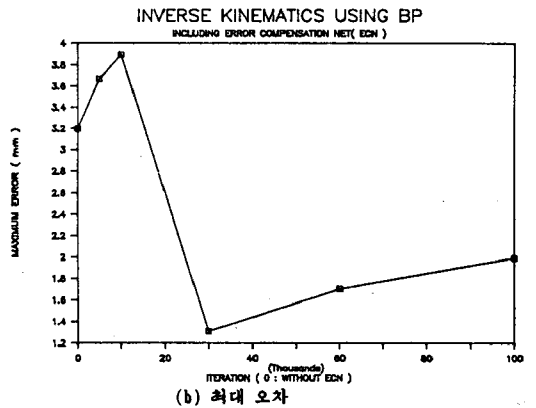
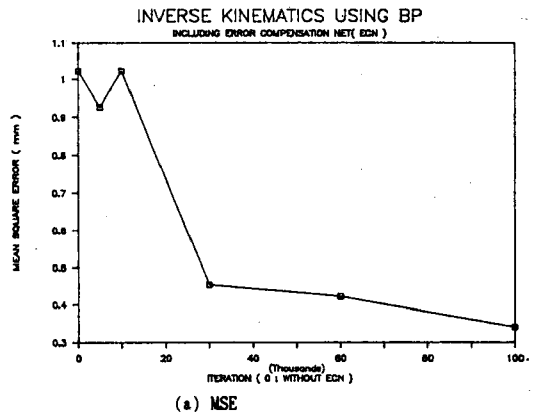


그림12. ECNN을 추가한 경우 학습 회수에 따른 오차의 변화

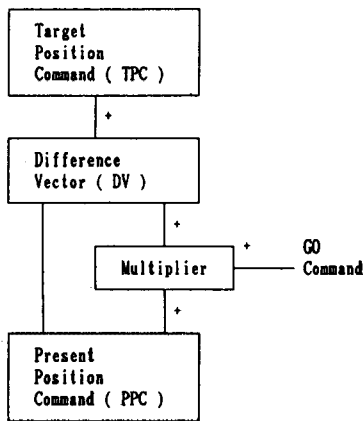


그림14. 자동 추적 신경 회로망 구조

이 구조의 신경 회로망을 그림15와 같이 반복적인 알고리즘으로 표현해 보기로 한다.

그림15에서 P0는 초기 위치를 나타내며, PD는 매 샘플링 시간마다 증분될 위치의 양(Position Difference: PD)을 의미한다. 이 알고리즘으로부터 매 샘플링 시간에서의 PPC 값을 구하면 다음과 같다.

$$PPC(0) = P_0$$

$$PPC(1) = PPC(0) + (TPC - P_0) * GO(1)$$

```

PPC = P0;
while ( PPC != TPC ) {
    DV = TPC - PPC;
    PD = DV * GO;
    PPC = PPC + PD;
    MOVE( PPC );
}
    
```

그림15. 자동 추적 형성 신경 회로망의 알고리즘 구현

$$PPC(2) = PPC(1) + (TPC - P_0) * (1 - GO(1)) * GO(2)$$

$$PPC(3) = PPC(2) + (TPC - P_0) * (1 - GO(1)) * (1 - GO(2)) * GO(3) \quad (5)$$

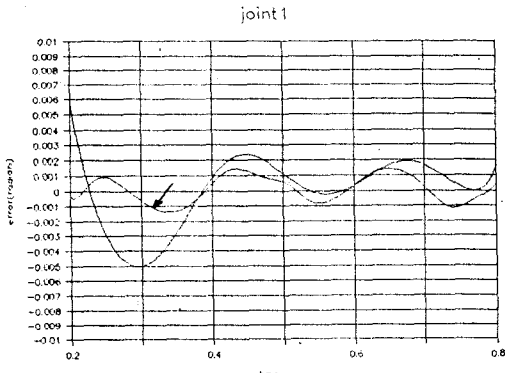
$$PPC(i) = P(i-1) + (TPC - P_0) * \prod_{k=0}^{i-1} (1 - GO(k)) * GO(i)$$

$$\text{where } GO(0) = 0.$$

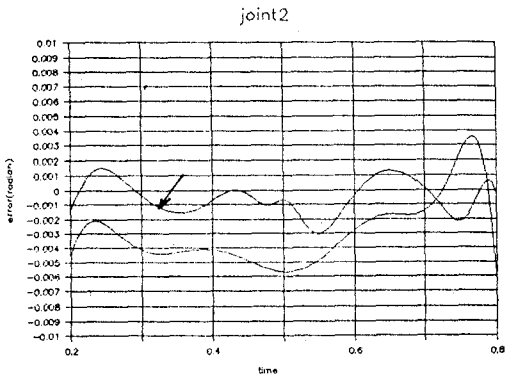
식(5)로부터 매 샘플링 시간마다 더해지는 위치 명령값 PD를 구하면 (6)식이 되는데, i번째 샘플링 시간에서의 PD(i) 값은 (TPC - P0)가 고정상수이므로 GO(i)에 의해서만 그 때만이 결정됨을 알 수 있다.

$$PD(i) = (TPC - P_0) * \prod_{k=0}^{i-1} (1 - GO(k)) * GO(i) \quad (6-1)$$

$$GO(i) = \frac{PD(i)}{(TPC - P_0) * \prod_{k=0}^{i-1} (1 - GO(k))} \quad (6-2)$$



(a) 1축 (→: ECNN을 추가한 경우)



(a) 2축 (→: ECNN을 추가한 경우)

그림13. ECNN을 추가한 역기구학 시뮬레이션 결과

즉,

$$\sum PD(i) = (TPC - P_0) \quad (7)$$

이므로

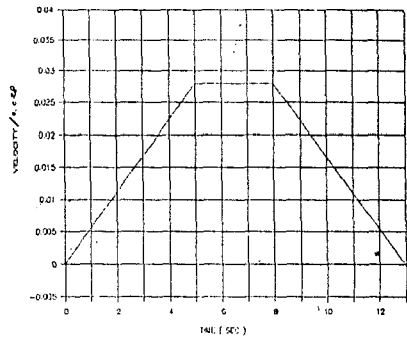
$$\sum_{i=0}^{i-1} \{ \pi(1 - GO(k)) * GO(i) \} = 1 \quad (8)$$

이 되어 각 관절의 이동 거리의 차이가 존재하여도 이 구조에 의하여 동시에 운동을 정지시킬 수 있는 제약을 형성할 수 있음을 보여 주며, 또 (6-1)식에서와 같이 매 샘플링 시간에서의 PD(i) 값이 이동 거리 (TPC - P₀)를 비례상수로 하는 형태이므로 이 구조에 의해 생성된 제력은 각 관절의 이동 거리의 차이가 존재하여도 같은 형태의 운동 프로파일을 갖게된다. 즉 GO 명령이 결정되면 이 구조를 통하여, 이동할 거리가 서로 다른 여러 관절들이 같은 형태의 운동 프로파일을 가지며 동시에 동작을 끝낼 수 있는 제약을 형성할 수 있다.

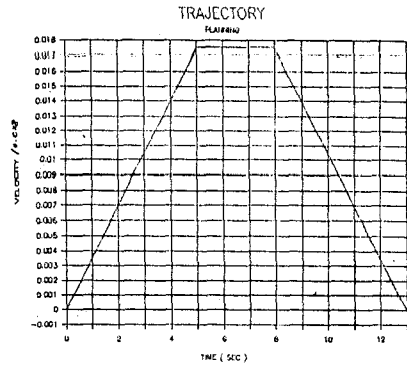
그림16의 b)는, 그림16의 a)와 같은 속도 및 위치 프로파일을 갖는 제약을 형성하고, 이 제약으로부터 (6-2)식을 사용하여 GO 함수를 구한 다음 이 GO함수를 이용하여 5의 거리를 이동할 관절의 제력을 형성시킨 결과이다. 그림16의 a)와 b)로부터 두 경우가 모두 같은 형태의 속도 및 위치 프로파일을 가지며, 13초 경과 후 목표 지점에 동시에 도달함을 볼 수 있다. 이것은 다관절 로봇의 PTP 운동의 경우에 가장 이동 시간이 오래 소요되는 축에 대하여 계산에 의해 제력을 형성하고 이 제약으로부터 GO함수를 계산하여 나머지 축들의 제력 형성에 사용할 수 있음을 보인다.

위에서 설명한 자동 제력 형성 신경 회로망의 문제는 어떻게 원하는 프로파일을 갖도록 GO 함수를 발생시키는가 이다. 그림17은 사다리꼴의 속도 프로파일을 갖도록 제력을 형성하고 식(6-2)를 이용하여 역으로 구한 GO 함수를 나타내는데, 가속도, 음적일 거리, 최대 속도 등의 값에 따라서 GO함수가 변하는 모양을 나타낸다. 그림17에서 GO 함수가 부드러운 곡선으로 표현되지 않음을 볼 수 있는데 이것은 가속도값이 변하는 경계점에서 발생한다.

TRAJECTORY PLANNING



a) 계산에 의한 제력



b) a)로 구한 GO로부터 구한 제력

그림16. 자동 제력 발생 시뮬레이션 결과

4. 결론

이상에서는 신경 회로망을 이용한 로봇 매니퓰레이터의 PTP 운동을 위하여 각기 다른 구조의 신경 회로망을 사용한 역기구학 문제의 해결과 관절 제력 형성에 관해 서술하였다.

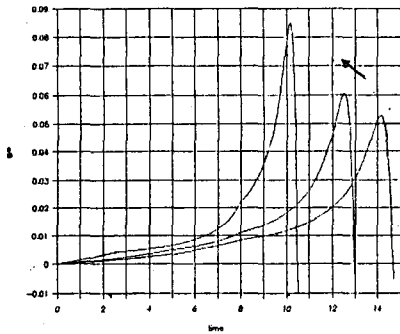
본 논문에서는 로봇의 역기구학 문제 해결을 다층 구조의 Perceptron 신경 회로망에 EBP 학습 방법을 통하여 수행하였으며, 신경 회로망의 중간층의 수와 각 중간층 신경 세포(neuron)의 수에 따른 성능 평가와 학습 회수에 따른 성능 평가를 수행하였다. 또한 오차보정 신경 회로망(Error Compensation Neural Net: ECNN)을 추가하여 역기구학 문제 해결의 성능 향상을 보였다.

또한 본 논문에서는 Grossberg등이 제안한 자동 관절 제력 형성 신경 회로망을 분석하여 이 신경 회로망에 의한 관절 제력 형성은 다관절이 서로 다른 거리를 이동하여도 각 관절의 이동 목표점만 주어지면 같은 운동 프로파일을 가지며 동시에 운동을 마치도록 이루어짐을 보였다. 그리고 이 구조의 신경 회로망에 의한 관절 형성의 문제점을 제시하는 한편, 다관절 로봇의 PTP 운동의 경우에 가장 이동 시간이 오래 소요되는 축에 대하여 계산에 의해 제력을 형성하고 이 제약으로부터 GO함수를 계산하여 나머지 축들의 제력 형성에 사용할 수 있음을 보여, 로봇의 기존의 관절 제력 형성에 이 구조의 신경 회로망을 응용하여 계산 속도의 이득을 얻을 수 있음을 보였다.

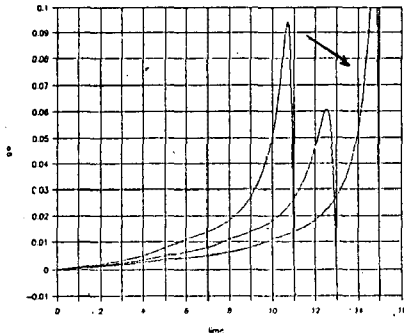
참고 문헌

1. K.S.Fu, R.C.Gonzalez, and C.S.G.Lee, ROBOTICS : Control, Sensing, Vision, and Intelligence, McGraw-Hill Book Company, 1987.
2. Gary Josin, "Integrating Neural Networks with Robots," AI Expert, pp.50-58, August, 1988.
3. Allon Guez, and Ziauddin Ahmad, "Solution to the inverse kinematics problem in Robotics by Neural Networks," IEEE ICNN'88, vol.11, pp.617-624, July 1988.
4. W.Thomas Miller, III, "Sensor-Based Control of Robotic Manipulators Using a General Learning Algorithm," IEEE Journal of Robotics and Automation, vol.RA-3, No.2, pp.157-165, April 1987.

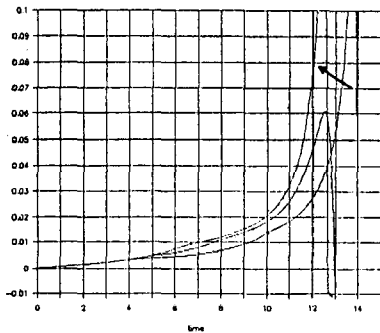
5. J.S.Albus, " A New Approach to Manipulator Control : The Cerebellar Model Articulation Controller (CMAC), " Transactions of the ASME, pp.220-227, September 1975.
6. G.Josin, D.Charney, and D.White, " Robot Control Using Neural Networks," IEEE ICNN'88, vol.II, pp.625-631, July 1988.
7. D.E.Rumelhart, J.L.McClelland, and the PDP Research Group, Parallel Distributed Processing: Explorations in the Microstructures of Cognition, vol.1, pp.318-362, 1986.
8. Daniel Bullock, and Stephen Grossberg, " A Neural Network Architecture for Automatic Trajectory Formation and Coordination of Multiple Effectors During Variable-Speed Arm Movements," IEEE ICON'87, vol.IV, pp.559-566, June 1987.
9. Stephen Grossberg, " Nonlinear Neural Networks: Principles, Mechanisms, and Architectures," Neural Networks, vol.1, pp. 17-61, 1988.



a) 가속도 : 가변, 움직일 거리, 최대 속도 : 고정



b) 움직일 거리 : 가변, 가속도, 최대 속도 : 고정



c) 최대 속도 : 가변, 움직일 거리, 가속도 : 고정

그림17. 사다리꼴 속도 프로파일을 갖는 제적으로부터 가속도, 움직일 거리, 최대 속도의 변화에 따라 구한 GO 함수