

# 홉필드 신경 회로망을 이용한 로봇 매니플레이터의 최적 시간 경로 계획

김영관\*, 조현찬\*, 이홍기\*\*, 진홍태\*

\* 중앙대학교 전자공학과 \*\* 중앙대학교 제어계측공학과

## Planning a minimum time path for robot manipulator using Hopfield neural network

Young Kwan Kim\*, Hyun Chan Cho\*, Hong Gi Lee\*\*, and Hong Tae Jeon\*

\* Dept. of Electronics Engineering, Chungang Univ.

\*\* Dept. of Control and Instrumentation Engineering,

Chungang Univ.

### ABSTRACT

We propose a minimum-time path planning scheme for the robot manipulator using Hopfield neural network. The minimum-time path planning, which can allow the robot system to perform the demanded tasks with a minimum execution time, may be of consequence to improve the productivity. But most of the methods proposed till now suffers from a significant computational burden and thus limits the on-line application. One way to avoid such a difficulty is to apply the neural network technique, which can allow the parallel computation, to the minimum-time problem. This paper propose an approach for solving the minimum-time path planning by using Hopfield neural network. The effectiveness of the proposed method is demonstrated using the PUMA 560 manipulator.

### 1. 서론

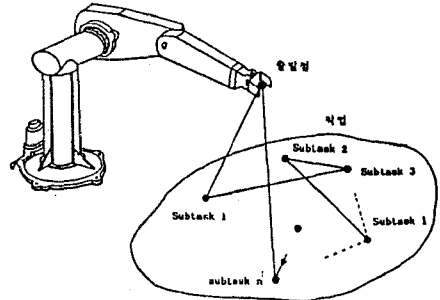
산업용 로봇은 다수의 링크들과 조인트들로 구성되어 있으며 공장 자동화에서 중추적 역할을 담당하는 컴퓨터로 제어되는 기계적인 매니플레이터 시스템이다. 이러한 매니플레이터는 프로그램의 변환을 통해 다양한 작업을 할 수 있기 때문에 FMS의 주요 work cell로 인식되고 있다.

일반적으로 산업용 로봇이 수행하는 작업(drilling, welding, painting, assembling, 그리고 운반 작업등)들은 여러개의 subtask (이는 로봇의 소작업을 의미함)들과 그 subtask 사이에서의 로봇의 주행 경로(travelling path)들로 구성되어 있으며, 다음 2 종류로 대분할 수 있다.

첫째, 각 subtask들 사이에 수행 순위 (order of execution)가 미리 규정되어 있는 작업(그림 1 참조).

둘째, 수행 순위가 규정되어 있지 않은 subtask들로 구성된 작업.

산업용 로봇이 상기 2 부류 작업들을 수행하는데 소요 되는 작업 시간은 공히 각 subtask의 수행 시간들과 subtask 사이에서의 주행 시간들의 합이다. 따라서 이 작업 시간의 최소화 (즉, subtask 수행 시간과 주행 시간의 최소화)는 로봇의 생산성 극대화에 크게 기여 할 것이다. 그러나 많은 예에서 각 subtask의 수행 시간은 일정하다고 간주할 수 있기 때문에 작업 시간의 최소화는 주행 시간의 최소화와 일치한다고 생각할 수 있다.



수행순서 : 출발점 → subtask 1 → subtask 3 →  
subtask 2 → subtask 1 → ... subtask  
→ subtask n → 출발점

그림 1. 수행 순위가 규정된 n개의 subtask들로 구성된 작업.

Fig. 1. A Task consisting of n subtasks with the prespecified execution order.

작업 시간의 최소화 (즉 주행 시간의 최소화)는 앞서 본류한 두 종류의 작업에 따라 각기 다르게 진행되어야 한다. 전자의 경우, 즉 작업 순서가 미리 규정된 경우에는 그 주행 경로들이 정해져 있기 때문에 기존의 연구 결과 ([1], [11])들에 의해 해결될 수 있다.

그러나 후자의 경우에는 각 subtask들 사이에서의 주행 시간의 최소화 뿐만 아니라 주행 경로의 선정 (혹은 각 subtask의 수행 순위 결정)이 전체 작업 시간의 최소화에 큰 영향을 미친다. 따라서 후자의 작업인 경우에는 각 subtask 사이에서의 주행 시간의 최소화 뿐만 아니라 최소 시간 측면에서의 경로 선정 (selection of path)에 관한 연구가 동시에 이루어져야 한다.

본 논문에서는 각 subtask에서 로봇의 주행 경로 (즉 수명 순위)가 정해져 있지 않는 경우, 그 작업 시간의 최소화를 얻기 위한 경로 선정에 관한 방법이 제시 된다. 이를 위해 기존의 계산 구조와는 다른 parallel processing의 잇점등을 가져 on-line 계산을 가능케하는 Hopfield 신경 회로망 (neural network) [7], 기법을 도입한다. 또한 각 subtask 사이에서의 최소 주행 시간을 결정하는데 있어 복잡하고 과다한 계산량의 문제점을 갖는 기존의 연구 결과들보다 계산 측면에서 좀더 효율적인 방법을 이용 실제 최적 시간 제어 시스템의 on-line implementation을 가능케 하고자 한다.

최근에 많은 관심을 불러 일으키고 있는 신경 회로망 (neural network) 기법은 인간 두뇌가 대량의 정보를 효율적으로 병렬 처리할 수 있을뿐 아니라 학습 능력을 갖고 있다는 점에 착안하여 제안되었다. 신경 회로망은 기존의 구조와는 전혀 다른 parallel processing의 구조를 갖고 있기 때문에 복잡한 계산을 필로로 하는 문제 해결에 널리 응용되고 있으며 특히 최적화 문제, pattern recognition, AI, optical 기술 등에 효율적으로 이용되고 있다.

최적화 문제 해결에 있어 널리 응용되는 신경 회로망의 정보 처리 방식은 크게 Back propagation에 의한 회로망 학습법, 확률적 동작을 도입한 Boltzman machine 에 의한 학습법, 그리고 Hopfield 회로망에 의한 학습법등 세가지로 분류할 수 있다.

이중 Hopfield 신경 회로망은 대칭적인 회로망에서 에너지가 부르는 속성치를 설정하여 그 에너지를 감소 시키는 방향으로 회로망을 동작시키는 학습법이다. Hopfield 신경 회로망은 TSP(Travelling Salesman Problem)문제 ([13],[14],[15]) 등과 같은 고전적인 최적화 문제등에 널리 응용되고 있다.

한편 논문의 구성은 다음과 같다. II 장에서는 홉필드 신경망 회로 구성에 관한 설명이 이루어지며, III 장에서는 주어진 작업 공간상에서의 두 TASK 사이의 최소 주행 시간의 결정과 전체 주행 시간의 최적화를 위한 홉필드 신경 회로망 구성이 설명된다. VI 장에서는 제안된 방법의 컴퓨터 모의 실험과 결과가 설명되며, 마지막으로 V 장에서는 결론 및 앞으로의 연구 방향을 제시된다.

## II. TSP의 홉필드 신경 회로망

TSP(traveling salesman problem)란 한 판매원이 n개의 도시를 모두 방문해야 할 경우, 어떤 순서로 도시들을 방문하면 방문 거리가 최소가 되는가를 결정하는 고전적인 최적화 문제로 다음과 같은 가정을 갖는다.

- 1) 처음 출발한 도시는 최종 도시와 일치 한다.
- 2) 판매원은 모든 도시를 정확히 한번만 방문 하여야 한다.

기존의 탐색 방법에 의해 상기 TSP를 해결할 경우(즉, 최적값을 구할 경우) 그 계산량은 도시의 수에 비례하여 폭발적으로 비례 한다. 이러한 어려움을 극복할 수 있는 한 방법이 최근에 각광을 받고 있는 홉필드 신경 회로망을 TSP 문제에 응용하는 것이다 [6,7,13,14,15].

n개의 도시들로 구성되어 있는 TSP문제에 홉필드 회로망을 적용하기 위해서는 n x n 순열 행렬(permutation matrix)의 (참조 그림 2) element 갯수에 해당하는 n<sup>2</sup>개의 뉴론들이 필요하다.

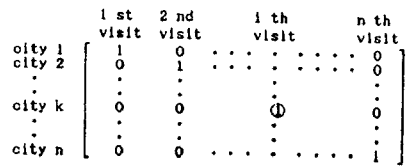


그림 2. n x n 순환 행렬. 여기에서 ①은 k 번째 도시를 i 번째로 방문함을 의미 한다.

Fig.2. n x n permutation matrix. Here a "Ⓚ" means that a salesman visits kth city at the i-th visit.

이때 각 뉴론들의 입-출력 관계는 다음과 같은 비선형 시그모이드 (nonlinear sigmoid) 함수 g(u<sub>ki</sub>)로 표현 된다.

$$V_{ki} = g(u_{ki}) = 1/2(1 + \tanh(u_{ki}/u_0)) \quad (1)$$

여기에서 u<sub>ki</sub>는 k 번째 도시의 i 번째 방문에 해당하는 뉴론의 입력을 의미하며 V<sub>ki</sub>(∈[0,1])은 그 출력을 뜻하고 u<sub>0</sub>는 상수이다. 그리고 홉필드 신경 회로망은 다음과 같은 에너지 함수를 갖는다.

$$E = A/2 \sum_k \sum_i \sum_{i \neq k} V_{ki} V_{kj} + B/2 \sum_k \sum_i \sum_{k \neq i} V_{ki} V_{ii} + C/2 \left( \sum_k \sum_i V_{ki} - N \right)^2 + D/2 \sum_k \sum_i d_{ki} V_{ki} (V_{i,i+1} + V_{i,i-1}) \quad (2)$$

여기에서 A,B,C,D는 양의 정수이며 d<sub>ki</sub>는 k 번째와 i 번째 도시 사이의 거리를 나타낸다. 상기 에너지 함수에서 우연의 첫번째항은 판매원이 각 도시를 한번만 방문하도록 하기 위한 penalty 함수이며 두번째항은 판매원이 i 번째 순회시에는 한 도시만 방문하도록 하기 위한 penalty 함수이다. 또한 세번째항은 모든 도시를 한번씩 방문하였는가를 확인하는 penalty 함수이고 마지막항은 각 도시의 방문 거리를 나타낸다. 특히 마지막항은 출발 도시와 최종 도착 도시가 동일하도록 하기 위한 penalty 함수를 동시에 포함하고 있다.

상기 에너지 함수 E를 감소 시키기 위한, 즉 가정 1)과 2)를 만족하고 아울러 방문 거리를 최소화 시키기 위한, 회로망의 입력 u<sub>ki</sub> 및 V<sub>ki</sub>를 결정하기 위해 Platt와 Barr [13,15]는 다음과 같은 탐색 알고리즘을 제안 하였다.

$$du_{ki}/dt = -u_{ki}/\tau - A \sum_{j \neq i} V_{kj} - B \sum_{i \neq k} V_{ii} - C \left( \sum_k \sum_i V_{ki} - n \right)^2 - D \sum_j d_{ki} (V_{j,i+1} + V_{j,i-1}) \quad (3)$$

$$V_{ki} = g(u_{ki}) = 1/2(1 + \tanh(u_{ki}/u_0)) \quad (4)$$

여기에서 τ는 각 뉴론의 시상수이다. 상기식들 (식(2), 식(3)과(4))의 연속적인 iteration후 최적값 u<sub>ki</sub> 및 V<sub>ki</sub>가 결정된다. 이때 u<sub>ki</sub>는 주어진 에너지 함수 E(즉, 방문 거리)를 최적화하기 위한 각 뉴론의 상태를 나타내며 V<sub>ki</sub>는 판매원이 방문거리를 최소화하는 도시들의 최적 방문 순서를 나타낸다.

### III. 휴필드 모델을 이용한 최적 시간 경로 설정

매니퓰레이터의 작업 공간에 다음과 같은 성질을 갖는 n 개의 sub-task들이 주어 진다고 가정 한다(그림 3 참조).

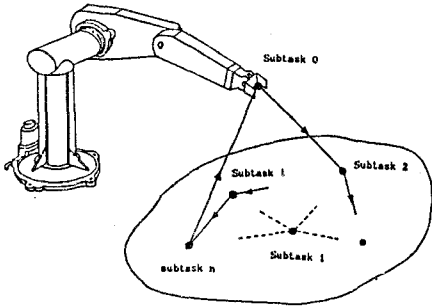


그림 3. 수행 순위가 규정되어 있지 않은 subtask들로 구성된 작업.

Fig. 3. A Task consisting of n subtasks without prespecifying the execution order.

- 1) 각 subtask의 수행 priority 와 시간은 동일 하다.
- 2) 임의의 두 subtask 사이에서의 수행 경로는 직선 경로이다.
- 3) 모든 subtask를 수행한후 매니퓰레이터는 처음 출발점 subtask 0 로 귀환 한다.

상기 작업을 수행하기 위해 매니퓰레이터가 선택할 수 있는 주행 경로는 많이 있지만 최소 시간 측면에서는 그 경로들은 한정 되어 있다. 이 한정된 경로, 즉 최적 시간 측면에서의 경로를 결정하기 위해서는 계산상의 잇점을 포함한 많은 장점을 갖는 휴필드 신경망 기법을 도입 한다.

일반적으로 매니퓰레이터가 주어진 작업을 최소 시간내에 완료 하기 위해서는 모든 subtask 들의 수행 시간과 subtask들 사이에서의 전체 주행 시간의 최소화가 필요하다. 그러나 각 subtask의 수행 시간과 임의의 두 subtask들 사이에서의 최소 주행 시간을 규정할 수 있다면, 주어진 최적 시간 경로 설정 문제는 subtask 들의 수행 순서(order of execution)를 결정하는 문제로 변환 된다. 더구나 두 subtask 사이에서의 최소 주행 시간을 거리 변수로 치환 한다면 상기 문제는 기존의 TSP 문제로 간주할 수 있으며 문제 해결에 휴필드 신경망 기법의 도입을 가능케 한다.

휴필드 신경망에 의한 문제 해결을 위해 두 subtask 사이에서의 최소 주행 시간 결정 과정이 다음에 설명된다.

#### III - 1 각 subtask 사이에서의 최소 주행 시간 결정

작업 공간내의 위치한 1번째 subtask ( i=1,2,...,n )의 위치(position) 와 방위(orientation)는 다음과 같은 4 x 4 Homogeneous transformation matrix 에 의해 표현될 수 있다.

$$H_i \in R^{4 \times 4} = \begin{bmatrix} n_i & s_i & a_i & p_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

여기에서  $n_i = [n_{xi}, n_{yi}, n_{zi}]^T$ ,  $s_i = [s_{xi}, s_{yi}, s_{zi}]^T$ , 그리고  $a_i = [a_{xi}, a_{yi}, a_{zi}]^T$  는 방위를 나타내는 단위 normal, slide, approach 벡터들이며  $p_i = [p_{xi}, p_{yi}, p_{zi}]^T$  는 위치 벡터이다.

1 번째 subtask를 수행하는 동안, end-effector의 카르테시안 주행 속도 (Cartesian travelling velocity) 성분은 0 으로 간주한다. 따라서 1 번째 subtask 수행후 매니퓰레이터가 일정한 위치 및 방위 속도를 갖는 다음 직선 경로를 수행 하기 위해서는 일정한 가속 구간이 필요하다. 이 1 번째 subtask를 수행 하기 직전에는 직선 주행 속도의 감속을 위한 감속 구간이 요구된다. 이러한 가속 및 감속 구간을 설정하기 위해 중간 경로점들을 다음과 같이 설정 한다(그림 4 참조).

$$H_{k+1} = H_i + k (H_{i+1} - H_i) \\ = [ n_{k+1}, s_{k+1}, a_{k+1}, p_{k+1} ] \quad (6)$$

$$H_{k+1} = H_{i+1} - k (H_{i+1} - H_i) \\ = [ n_{k+1}, s_{k+1}, a_{k+1}, p_{k+1} ] \quad (7)$$

여기에서  $k(0 \leq k \leq 1)$ 은 각/가속 구간의 길이를 위한 스칼라 값이며 구간  $H_i - H_{k+1}$  는 가속 구간 그리고 구간  $H_{k+1} - H_{i+1}$  는 감속 구간이 된다.

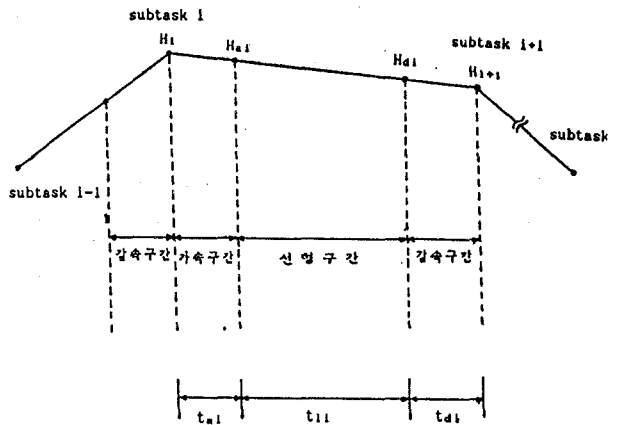


그림 4. 가속 및 감속 구간의 설정.

Fig. 4. Establishment of the acceleration and deceleration sections.

한편 각 주행 구간에서 매니퓰레이터 end-effector의 주행 시간들을  $t_{a1}$ ,  $t_{l1}$ , 그리고  $t_{d1}$  라 하면, 각 구간에서 end-effector의 카르테시안 선형 속도와 가속도 성분들을 다음과 같이 얻어 진다.

$$V_{a1} = \frac{V_{i1}}{t_{a1}} t, \quad \dot{V}_{a1} = \frac{V_{i1}}{t_{a1}} \quad (\text{가속 구간}) \quad (8)$$

$$V_{l1} = \frac{p_{d1} - p_{a1}}{t_{l1}}, \quad \dot{V}_{l1} = 0 \quad (\text{선형 구간}) \quad (9)$$

$$V_{d1} = -\frac{V_{i1}}{t_{d1}} (t - t_{a1} - t_{d1}), \quad \dot{V}_{d1} = \frac{-V_{i1}}{t_{d1}} \quad (\text{감속 구간}) \quad (10)$$

여기에서  $V_{a1}, \dot{V}_{a1}, \ddot{V}_{a1}$  는 각 구간에서의 선형 속도 벡터( $\in R^3$ )를 나타내며  $\dot{V}_{a1}, \ddot{V}_{a1}$  그리고  $\ddot{V}_{a1}$ 는 선형 가속도 벡터( $\in R^3$ )이다. 마찬가지로 각 구간에서의 방위 속도와 가속도들은 다음과 같이 유도된다.

$$\omega_{a1} = \frac{\omega_{11}}{t_{a1}} t, \quad \dot{\omega}_{a1} = \frac{\omega_{11}}{t_{a1}} \quad (\text{가속구간}) \quad (11)$$

$$\omega_{11} = \frac{\theta_{11}}{t_{11}} \cdot \rho_{11}, \quad \dot{\omega}_{11} = 0 \quad (\text{선형구간}) \quad (12)$$

$$\omega_{d1} = -\frac{\omega_{11}(t-t_{a1}-t_{11})}{t_{d1}}, \quad \dot{\omega}_{d1} = -\frac{\omega_{11}}{t_{d1}} \quad (\text{감속구간}) \quad (13)$$

여기에서  $\omega_{a1}, \omega_{11}, \omega_{d1}$  는 각 구간에서의 방위 속도( $\in R^3$ )를 나타내며  $\dot{\omega}_{a1}, \dot{\omega}_{11}, \dot{\omega}_{d1}$ 는 방위 가속도 벡터( $\in R^3$ )를 나타낸다. 또한 각  $\theta_{11}$  와 회전 기준 벡터  $\rho_{11}(\in R^3)$  는 다음식에 의해 정의된다 [11].

$$\theta_{11} = \{[(n_{a1})^T n_{a1} + (s_{a1})^T s_{a1} + (a_{d1})^T n_{a1} - 1]\}, \quad (14)$$

$$\rho_{11} = \frac{1}{2\sin\theta_{11}} \begin{bmatrix} (a_{d1})^T n_{a1} - (s_{d1})^T s_{a1} \\ (n_{d1})^T n_{a1} - (a_{d1})^T n_{a1} \\ (s_{d1})^T n_{a1} - (n_{d1})^T s_{a1} \end{bmatrix} \quad (15)$$

한편 매니플레이터는 제한된 동동력을 갖기 때문에 그 end-effector 의 선형 속도  $V(\in R^3)$  와 가속도  $\dot{V}$  그리고 방위 속도  $\omega(\in R^3)$  와 가속도  $\dot{\omega}$  는 다음과 같은 제한치를 갖는다고 가정할 수 있다.

$$\|v(t)\| \leq kv \quad (16a)$$

$$\|\dot{v}(t)\| \leq ka \quad (16b)$$

$$\|\omega(t)\| \leq kw \quad (16c)$$

$$\|\dot{\omega}(t)\| \leq kw_a \quad (16d)$$

상기 제한값과 식 (8) - (13)을 이용해  $i$  번째 subtask 와  $(i+1)$  번째 subtask 사이에서의 최소 주행 시간  $t_{ii}$  는 다음과 같이 구할 수 있다.

$$t_{ii} = \max(\|p_{a1}-p_{a1}\| / kv, |\theta_{11}| / kw) \quad (17)$$

$$t_{a1} = t_{d1} = (\|v_{11}\| / ka, \|\omega_{11}\| / kw_a) \quad (18)$$

$$t_i = t_{a1} + t_{ii} + t_{d1} \quad (19)$$

여기에서 ( ) 은 최소 시간을 의미하며  $v_{11} = (p_{a1} - p_{a1}) / t_{11}$ ,  $\omega_{11} = |\theta_{11}| / t_{11}$  이다.

### III - 2. 홉필드 신경망에 의한 최소 시간 경로 결정

앞서 언급하였듯이 식(19)에 의해 결정된 임의의 두 subtask 사이에서의 최소 주행 시간을 거의 변수로 바꾸어 생각하면, 주어진 문제는 기존의 TSP 문제와 일치하게 된다. 따라서 본 절에서는 TSP 해결에 적용된 홉필드 신경망 기법을 이용 주어진 작업 공간에서 최적 시간 경로를 결정하고자 한다(그림 3 참조).

이를 위해  $n$ 개의 subtask 들에 대한 하나의  $n \times n$  순환 행렬이 구성되고 그 element 갯수에 해당하는  $n^2$  개의 뉴론들을 설정한다. 이때 각 뉴론의 입력력 관계는 시그모이드 함수로 표현 된다 [3,7,15].

$$V_{ki} = \text{sigmoid}(u_{ki}) \quad (20)$$

여기에서  $u_{ki}$ 는  $k$  번째 subtask의  $i$  번째 수행에 해당하는 뉴론의 입력을 의미하며  $V_{ki}(\in [0,1])$ 은 그 출력력을 나타낸다.

또한  $n^2$  개의 뉴론으로 구성된 홉필드 신경망의 에너지 함수  $E$ 를 다음과 같이 규정 한다.

$$E_1 = D/2 \sum_i \sum_k \sum_l t_{ki} V_{ki} (V_{i,i+1} + V_{i,i-1}) \quad (21a)$$

$$E_2 = A/2 \sum_k \sum_l \sum_j V_{ki} V_{kj} + \sum_i \sum_k \sum_l V_{ki} V_{il} + C/2 (\sum_k \sum_l V_{kl} - n)^2 \quad (21b)$$

$$E = E_1 + E_2, \quad (21c)$$

여기에서  $E_1$  = 모든 subtask를 수행하는데 소요되는 전체 주행 시간,

$t_{ki}$  = 식(19)에 얻어진  $k$  번째 subtask 와  $i$  번째 subtask 사이의 최소 주행 시간,

$E_2$  = 각 subtask를 한번만 수행해야 하는 penalty 함수,  $A, B, C, D$  = 임의의 양의 경우.

식 (21)의 에너지 함수  $E$ 를 최소화하는, 즉 매니플레이터의 주행 시간을 최소화 하기 위한 뉴론의 입력  $u_{ki}$  와 출력  $V_{ki}$ 은 다음의 탐색 알고리즘에 의해 결정된다.

$$du_{ki}/dt = -\frac{\partial E}{\partial V_{ki}} \quad (22a)$$

$$= -A \sum_{j \neq i} V_{kj} - B \sum_{k \neq i} V_{il} - C (\sum_k \sum_l V_{kl} - n)^2 - D \sum_l t_{ki} (V_{i,i+1} + V_{i,i-1}) \quad (22b)$$

$$V_{ki} = \text{sigmoid}(u_{ki}) = 1/2(1+\tanh(u_{ki}/u_0)) \quad (22c)$$

상기 식(22)의 연속적인 iteration 후 최적값  $u_{ki}$  및  $V_{ki}$ 가 결정된다. 이때  $u_{ki}$ 는 매니플레이터의 주행 시간을 최소화하기 위한 뉴론의 입력을 나타내며  $V_{ki}$ 는 subtask 들의 최적 수행 순서를 나타낸다.

## IV. 시뮬레이션 및 결과

본 논문에서 사용한 매니플레이터는 6 자유도의 최전속을 갖는 PUMA 560 로봇 매니플레이터를 사용 하였으며, 작업 공간에 5개 subtask 들을 선정 하였다.

$$H_1 = \begin{bmatrix} 0.4330 & 0.1250 & 0.6995 & 0.2 \\ 0.2500 & 0.9665 & -0.058 & 0.6 \\ -0.866 & 0.25 & 0.4330 & 0.2 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (23a)$$

$$H_2 = \begin{bmatrix} 0.3368 & 0.5316 & 0.7772 & 0.4 \\ 0.059 & 0.8117 & -0.5610 & 0.8 \\ -0.9397 & 0.2418 & 0.2418 & 0.2 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (23b)$$

$$H_3 = \begin{bmatrix} 0.3214 & -0.5295 & 0.7851 & 0.8 \\ 0.3830 & 0.0194 & 0.4036 & 0.3 \\ -0.8660 & 0.1710 & 0.4698 & 0.2 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (23c)$$

$$H_4 = \begin{bmatrix} 0.8365 & 0.2241 & 0.50 & 1.0 \\ 0.4829 & 0.1294 & -0.2241 & 0.6 \\ -0.2588 & 0.9650 & 0.0 & 0.2 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (23d)$$

$$H_5 = \begin{bmatrix} 0.3830 & -0.7971 & 0.4669 & 0.4 \\ 0.6634 & 0.5891 & 0.4614 & 0.1 \\ -0.6428 & 0.1330 & 0.7544 & 0.2 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (23e)$$

상기 5개 subtask들 사이에서 미니플레이어의 주행 경로는 카르테시안 직선 경로로 하였으며 가, 가속 구간의 설정을 위해  $k = 0.1$ 로 가정한다. 또한 모의 실험에서 사용한 초기 조건은  $V_{k1}(-1) = 1.0/N + \theta_{k1}$ 로 (여기서, N은 모의 실험 과정에서의 subtask의 갯수이며  $\theta_{k1}$ 은 임의의 상수( $<0.1$  rad)이다) 하였으며 선형 및 방위속도의 가속도의 상한치들을 다음 값들로 선정하였다.

$$K_v = 0.6 \text{ m/s} \quad K_a = 0.3 \text{ m/s}^2$$

$$K_w = 1.0 \text{ rad/s} \quad K_{\omega} = 0.5 \text{ rad/s}^2$$

또한 가중치 변화의 시상수는 0.4로 하였으며, 각 subtask사이의 최소 주행 시간은 식 (17) - (18)로부터 구하였다. 그 결과들이 표 1에 정리 되었다. 상기 주어진 값들과 각 subtask 사이의 최소 주행 시간을 이용 Hopfield 망의 구성이 이루어졌으며 최소 주행 시간을 위한 주행 경로의 결정에 관한 결과들을 그림 5 - 그림 9 그리고 표 2를 통해 보여 주고 있다. 그림 5는 반복 학습 횟수에 따른 첫번째 뉴런의 출력 상태를 보여 주고 있고 그림 6, 그림 7, 그림 8, 그리고 그림 9는 두번째 뉴런, 세번째, 네번째, 다섯번째의 뉴런들의 출력 상태를 보여주고 있다. 이 결과들로부터 반복 학습에 따른 뉴런 출력들이 요구되는 상태로 훌륭히 수렴됨을 알 수 있다.

인편 표 2는 iterations이 끝난후 얻어진 결과들로부터 최소 주행 시간을 갖는 경로 선정 및 적임 순위를 보여주고 있다. 이 정보들에 의해 모든 subtask의 작업을 수행 할 경우, 소요되는 시간은 4.9094 (sec.)로 이는 최소 주행 시간이다. 이 결과들은 표 1에 의해 쉽게 입증될 수 있다.

k \ i	subtask 1	subtask 2	subtask 3	subtask 4	subtask 5
subtask 1	0.0	1.5012	1.0348	1.0097	0.9972
subtask 2	1.5012	0.0	1.0672	0.9983	0.8728
subtask 3	1.0348	1.0672	0.0	1.2020	1.0181
subtask 4	1.0097	0.9983	1.2020	0.0	0.9249
subtask 5	0.9972	0.8728	1.0181	0.9249	0.0

표 1. 두개의 이웃 subtask들 사이의 최소 주행 시간(sec.).

Table 1. Minimum traveling time between two adjacent subtasks.

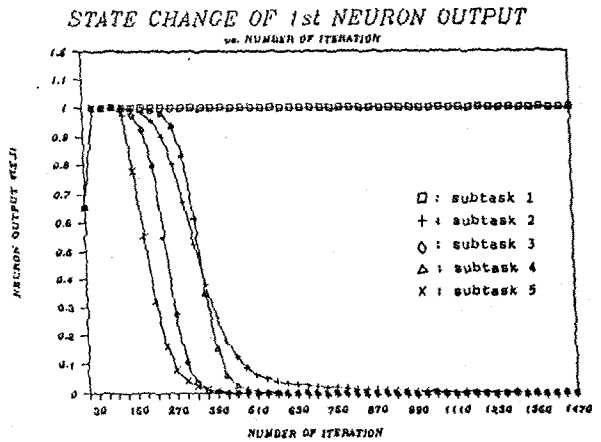


그림 5. 반복 학습에 따른 첫번째 뉴런의 출력 상태.

Fig. 5. Output state of the first neuron at each iteration.

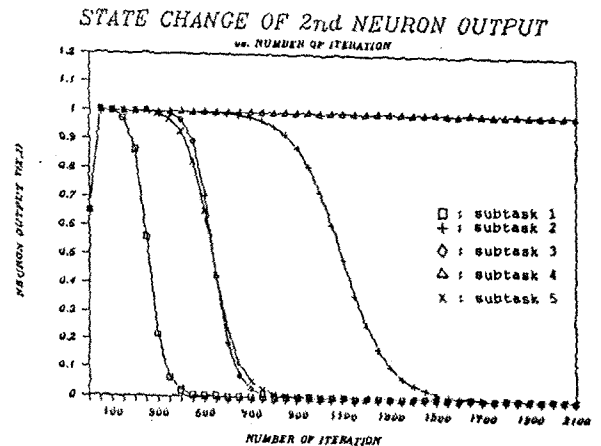


그림 6. 반복 학습에 따른 두번째 뉴런의 출력 상태.

Fig. 6. Output state of the 2nd neuron at each iteration.

STATE CHANGE OF 3rd NEURON OUTPUT  
vs. NUMBER OF ITERATION

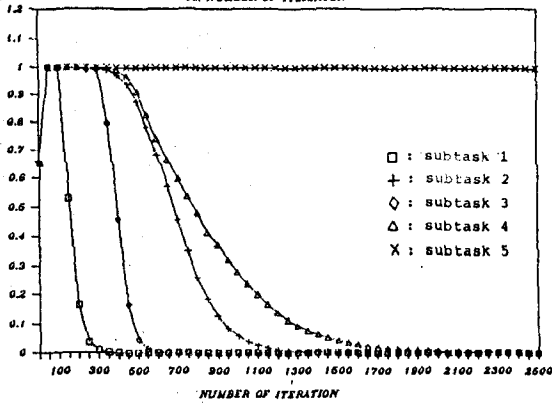


그림 7. 반복 학습에 따른 세번째 뉴런의 출력 상태.

Fig. 7. Output state of the 3rd neuron at each iteration.

STATE CHANGE OF 4th NEURON OUTPUT  
vs. NUMBER OF ITERATION

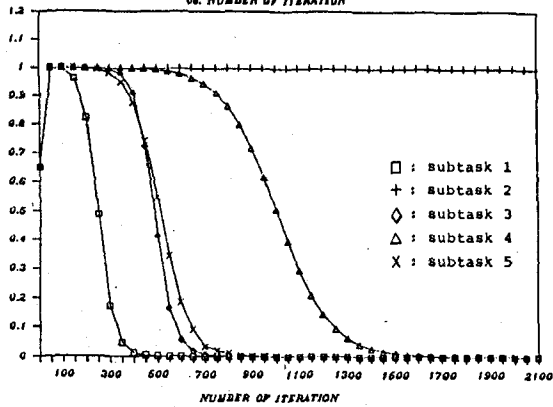


그림 8. 반복 학습에 따른 네번째 뉴런의 출력 상태.

Fig. 8. Output state of the 4th neuron at each iteration.

STATE CHANGE OF 5th NEURON OUTPUT  
vs. NUMBER OF ITERATION

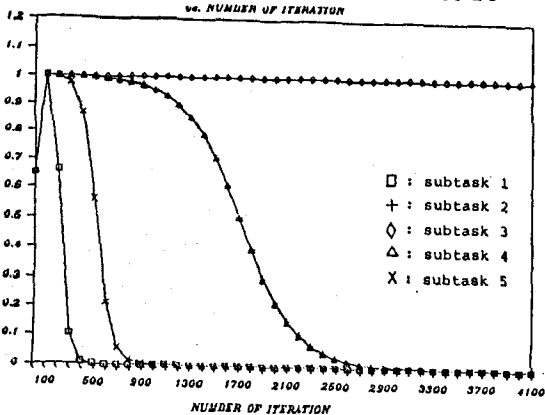


그림 9. 반복 학습에 따른 다섯번째 뉴런의 출력 상태.

Fig. 9. Output state of the 5th neuron at each iteration.

k \ position	1	2	3	4	5
subtask 1	1.0	0.0	0.0	0.0	0.0
subtask 2	0.0	0.0	0.0	1.0	0.0
subtask 3	0.0	0.0	0.0	0.0	1.0
subtask 4	0.0	1.0	0.0	0.0	0.0
subtask 5	0.0	0.0	1.0	0.0	0.0

Minimum Traveling Time = 4.9094 sec.

작업 순위 ; subtask 1 - subtask 4 - subtask 5 -  
subtask 2 - subtask 3 - subtask 1

표 2. 최소 시간 경로를 갖는 뉴런의 최종 상태 및 작업 순위.

Table 2. Final output state of each neuron yielding minimum-time path and execution order of subtask.

V. 결론

본 논문에서는 로봇 미니플레이터가 TASK 상에서의 작업 수행시, 최적 시간 경로 설정 문제를 홉필드 신경 회로망을 이용해서 해결했다.

기존의 수리 계획적인 방법에 의해서 작업을 수행할 경우 TASK의 수가 증가함에 따라서 그 작업 수행 경로도 폭발적으로 증가한다. 이 경우 본 논문에서 제안한 신경망 회로를 이용하면 계산량의 감소로 그 해를 용이하게 구할 수 있다.

본 논문의 효율성은 컴퓨터 모의 실험을 통하여 입증하였으며, wire의 연결, 나사 조이기 등 실질 조립 과정에서 그 효율성이 실질적으로 증대되리라 본다.

앞으로의 연구방향은 각 TASK의 수행 시간이 동일하지 않은 경우로의 확장과 동적 특성을 고려한 최적 경로의 결정등을 들수있다.

REFERENCES

[1] J.E. Bobrow, S. Dubowsky and J.S. Gibson, "Time-optimal control of robotics manipulators along specified paths," The International Journal of Robotics Research, Vol.4, no. 3, pp.3-17, 1985

[2] James.H. Cervantes and Richard.R. Hildebrant, "Comparison of three neuron-based computation schemes," IJCNN, Vol.3, pp 657-671 1987.

- [3] Gerald.W.Davis, "Sensitivity analysis of Hopfield neural network," IJCNN, Vol. 3, pp 325-328 1987.
- [4] R.K. Eleley, "A learning architecture for control based on Back-propagation neural network," Proc. of the IEEE Conf. on Neural Network, Vol.2, pp 587-594, 1988.
- [5] A.Guez and Z.ahmad, "Solution to the inverse kinematic problem in robotics by neural network," Proc. of the IEEE Conf. on Neural Networks, " 1988.
- [6] Shullesh.U.Hegde, Jeffery.L. Sweet and William.B. Levy, "Determination of parameters in a Hopfield/Tank computational network," Proc. of the IEEE Conf. on Neural Network, Vol.2, 1988.
- [7] J.J. Hopfield and D.W. Tank, "Neural computation of Decision in Optimization problem," Bio-Cybernetics, Vol.52, pp 141-152, 1985.
- [8] J.J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," Proc.National Acad.Sci. USA 81, pp 3088-3092, 1984.
- [9] J.J.Hopfield, "Neural network and physics systems with emergent collective computational ability," Proc.National Acad.Sci. USA 79, pp 2254-2558, 1982.
- [10] G.Josin, D.Charney, and D.White, "Robot control using neural networks," Proc. of the IEEE Int. Conf. on Neural Networks, 1988.
- [11] J.Y.S. Luh and C.S. Lin, "Optimum path Planning for Mechanical Manipulators," Trans. of the ASME, Vol.102, June 1981, pp142-151.
- [12] Paul.P. "The mathematics of computer controlled Manipulator," Proc.of the 1977 Joint Automatic Control Conference, 77CH-1220-3cs, Vol.1, pp 124-131.
- [13] Harold. Szu, "Fast TSP algorithm based on binary neuron output and analog neuron input using the zero-diagonal interconnect matrix and necessary and sufficient of the permutation matrix," IEEE.Inte.Conf. on Neural Network, Vol.2, pp 259-266, 1988.
- [14] David.E. Vandenbout and T.K. Miller, "A Traveling Salesman Problem objective function that work," IEEE.Inte.Conf. on Neural Network, Vol.2, pp 299-303, 1988.
- [15] E. Wacholder, J. Han and R.C. Mann, "An extension of the Hopfield-Tank model for the solution of the multiple Traveling Salesman Problem," IEEE.Inte.Conf. on Neural Network, Vol.2, pp 305-324, 1988.