

홉필드 신경 회로망을 이용한 로봇트 매니플레이터의 최적 시간 경로 계획

(Planning a Time-optimal Path for Robot Manipulator Using Hopfield Neural Network)

趙 鉉 讚*, 金 榮 寬**, 全 洪 兌*, 李 鴻 奇***

(Hyun Chan Cho, Young Kwan Kim, Hong Tae Jeon, and Hong Gi Lee)

要 約

본 논문에서는 홉필드 신경 회로망을 이용하여 로봇트 매니플레이터의 최적 시간 경로 계획(time-optimal path planning)을 제시한다. 로봇트 시스템이 최소 수행 시간에 주어진 임무를 수행하도록 하는 최적 시간 경로 계획은 생산성을 향상시키는 결과를 가져온다. 그러나 지금까지 제안된 대부분의 방법은 엄청나게 많은 계산량으로 인해 On-line 응용에는 그 한계가 있다. 이런 어려움을 극복하는 한 가지 방법으로서 병렬 계산이 가능한 신경 회로망 기법을 최적 시간 문제에 이용할 수 있다.

본 논문은 홉필드 신경 회로망을 이용하여 최적 시간 경로 계획을 해결하는 방법을 제안한다. 제안된 방법의 효율성은 PUMA 560매니플레이터를 이용하여 입증하였다.

Abstract

We propose a time-optimal path planning scheme for the robot manipulator using Hopfield neural network. The time-optimal 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 proposes an approach for solving the time-optimal path planning by using Hopfield neural network. The effectiveness of the proposed method is demonstrated using a PUMA 560 manipulator.

*正會員, 中央大學校 電子工學科
(Dept. of Elec. Eng., Chungang Univ.)

**正會員, (株)大韓電線 技術研究所
(The Korean Elec. Wire Co. Ltd. Technical
Reaserch Center)

***正會員, 中央大學校 制御計測工學科
(Dept. of Cont. & Inst. Eng., Chungang Univ.)

接受日字: 1990年 5月 1日

I. 서 론

산업용 로봇트는 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 2부류 작업들을 수행하는데 소요되는 작업시간은 공히 각 subtask의 수행 시간들과 subtask 사이에서의 주행 시간들의 합이다. 따라서 이 작업 시간의 최소화(즉, subtask 수행시간과 주행시간의 최소화)는 로봇 생산성 극대화에 크게 기여 할 것이다. 그러나 많은 예에서 각 subtask의 수행시간은 일정하다고 간주할 수 있기 때문에 작업 시간의 최소화는 주행시간의 최소화와 일치한다고 생각할 수 있다.

작업시간의 최소화(즉 주행시간의 최소화)는 앞서 분류한 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,^{11,12}들에 의해 해결될 수 있다.

그러나 후자의 경우에는 각 subtask들 사이에서의 주행시간의 최소화 뿐만 아니라 주행 경로의 선정(혹은 각 subtask의 수행 순위 결정)이 전체 작업시간의 최소화에 큰 영향을 미친다. 따라서 후자의 작

업인 경우에는 각 subtask 사이에서의 주행 시간의 최소화 뿐만 아니라 최소 시간 측면에서의 경로선정(selection of path)에 관한 연구가 동시에 이루어져야 한다.

최근에 많은 관심을 불러 일으키고 있는 신경 회로망(neural network) 기법은 인간 두뇌가 대량의 정보를 효율적으로 병렬처리할 수 있을뿐만 아니라 학습 능력을 갖고 있다는 점에 착안하여 제안 되었다. 신경 회로망은 기존의 구조와는 전혀 다른 parallel processing의 구조를 갖고 있기 때문에 복잡한 계산을 필요로 하는 문제 해결에 널리 응용되고 있다. 특히 Hopfield 신경 회로망은 대칭적인 회로망에서 에너지라 부르는 측정치를 설정하여 그 에너지를 감소시키는 방향으로 회로망을 동작시키는 학습법으로서 TSP(travelling salesman problem)문제,^{13,14,15}등과 같은 고전적인 최적화 문제에 널리 응용되고 있다.

본 논문에서는 상기의 hopfield신경 회로망 기법을 도입해 각 subtask에서 로봇의 주행 경로(즉 수행 순위)가 정해져 있지 않은 경우 그 작업시간의 최소화를 얻기 위한 경로 선정에 관한 방법이 제시된다. 또한 복잡하고 과다한 계산량의 문제점을 갖는 기존의 연구결과들 보다 계산 측면에서 좀더 효율적인 방법을 이용 실제 최적 시간제어 시스템의 on-line implementation을 가능케 하고자 한다.

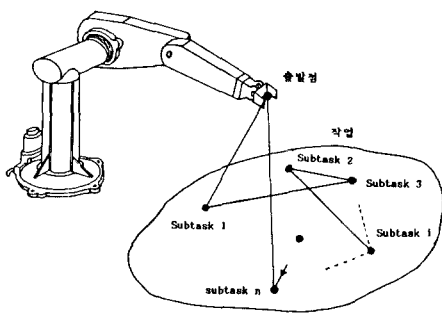
한편 논문의 구성은 다음과 같다. II 장에서는 홉필드 신경 회로망 구성에 관한 설명이 이루어지며, III 장에서는 주어진 작업 공간상에서의 두 subtask사이의 최소 주행 시간의 결정과 전체 주행 시간의 최적화를 위한 홉필드 신경 회로망 구성이 설명된다, IV 장에서는 제안된 방법의 컴퓨터 모의실험과 결과가 설명되며, 마지막으로 V 장에서는 결론 및 앞으로의 연구 방향을 제시한다.

II. TSP의 홉필드 신경회로망

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

- 1) 처음 출발한 도시는 최종 도시와 일치한다.
- 2) 판매원은 모든 도시를 정확히 한번만 방문하여야 한다.

기존의 탐색방법에 의해 상기 TSP를 해결할 경우(즉, 최적값을 구할 경우)그 계산량은 도시의 수에 비례하여 폭발적으로 비례한다. 이러한 어려움을 극



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

복할 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 신경 회로망을 적용하기 위해서는 n×n순열 행렬(permutation matrix)의(참조그림2) element 갯수에 해당하는 n²개의 뉴론들이 필요하다.

	1st visit	2nd visit	...	i th visit	...	n th visit
city1	1	0	...	∴	...	0
city2	0	1	...	∴	...	0
∴	∴	∴	∴	∴	∴	∴
city k	0	0	...	Ⓛ	...	0
∴	∴	∴	∴	∴	∴	∴
city n	0	0	...	∴	...	1

그림 2. n×n 순환 행렬.(여기에서 Ⓛ은 K번째 도시를 i번째로 방문함을 의미한다.)

Fig. 2. n×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_{ki})로 표현된다.

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

여기에서 u_{ki}는 k번째 도시의 i번째 방문에 해당하는 뉴론의 입력을 의미하며 V_{ki}(∈[0, 1])은 그 출력을 뜻하고 u₀는 상수이다. 그리고 홉필드 신경회로망은 다음과 같은 에너지 함수를 갖는다.

$$E = A/2 \sum_k \sum_l \sum_{i \neq k} V_{ki} V_{lj} + B/2 \sum_l \sum_k \sum_{k \neq l} V_{ki} V_{li} + C/2 (\sum_k \sum_l V_{kl} - N)^2 + D/2 \sum_k \sum_l \sum_i d_{kl} V_{ki} (V_{li+1} + V_{li-1}) \quad (2)$$

여기에서 A, B, C, D는 양의 정수이며 d_{kl}는 k번째와 l번째 도시 사이의 거리를 나타낸다. 상기 에너지 함수에서 우변의 첫번째항은 판매원이 각 도시를 한번만 방문하도록 하기 위한 penalty 함수이며 두번째항은 판매원이 i번째 순회시에 한 도시만 방문하도록 하기위한 penalty 함수이다. 또한 세번째항은 모든 도시를 한번씩 방문하였는가를 확인하는 penalty 함수이고 마지막항은 각 도시의 방문거리를 나타

낸다. 특히 마지막항은 출발 도시와 최종 도착 도시가 동일하도록 하기위한 penalty 함수를 동시에 포함하고 있다.

상기 에너지 함수 E를 감소시키기 위한, 즉 가정 1)과 2)를 만족하고 아울러 방문거리를 최소화 하기위한, 회로망의 입력 u_{ki} 및 V_{ki}를 결정하기 위해 platt와 barr^[13,15]는 다음과 같은 탐색 알고리즘을 제안하였다.

$$du_{ki}/dt = -u_{ki}/\tau - A \sum_{j \neq i} V_{kj} - B \sum_{i \neq k} V_{li} - C (\sum_k \sum_j V_{kj} - N)^2 - D \sum_l d_{kl} (V_{li+1} + V_{li-1}) \quad (3)$$

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

여기에서 τ는 각 뉴론의 시상수이다. 상기식들(식(2), 식(3)과(4))의 연속적인 iteration후 최적값 u_{ki}* 및 V_{ki}*가 결정된다. 이때 u_{ki}*는 주어진 에너지 함수 E(즉, 방문거리)를 최적화하기 위한 각 뉴론의 상태를 나타내며 V_{ki}*는 판매원이 방문거리를 최소화하는 도시들의 최적 방문 순서를 나타낸다.

III. 홉필드 모델을 이용한 최적 시간 경로 설정

매니플레이터의 작업공간에 다음과 같은 성질을 갖는 n개의 subtask들이 주어진다고 가정한다(그림3 참조).

- 1) 각 subtask의 수행 priority와 시간은 동일하다.
- 2) 임의의 두 subtask 사이에서의 수행경로는 직선 경로이다.
- 3) 모든 subtask를 수행한 후 매니플레이터는 처음 출발점 subtask 0로 귀환한다.

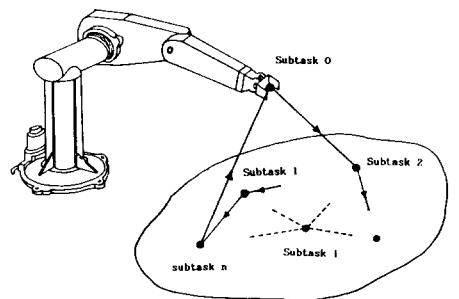


그림 3. 수행 순위가 규정되어 있지 않은 subtask들로 구성된 작업

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

상기 작업을 수행하기 위해 매니플레이터가 선택할 수 있는 주행 경로들은 많이 있지만 최소시간 측면에서는 그 경로들은 한정되어 있다. 이 한정된 경로, 즉 최적 시간 측면에서의 경로를 결정하기 위해 본 절에서는 계산상의 잇점을 포함한 많은 장점들을 갖는 홉필드 신경망 기법을 도입한다.

일반적으로 매니플레이터가 주어진 작업을 최소 시간내에 완료하기 위해서는 모든 subtask 들의 수행시간과 subtask들 사이에서의 전체 주행 시간의 최소화가 필요하다. 그러나 각 subtask의 수행시간과 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 변환된다. 더구나 두 subtask 사이에서의 최소 주행 시간을 거리 변수로 치환 한다면 상기 문제는 기존의 TSP 문제로 간주할 수 있으며 문제 해결에 홉필드 신경 회로망 기법의 도입을 가능케 한다.

홉필드 신경 회로망에 의한 문제 해결을 위해 두 subtask 사이에서의 최소 주행 시간 결정 과정이 다음에 설명된다.

1. 각 subtask 사이에서의 최소 주행시간 결정

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

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

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 각각 방위를 나타내는 단위 normal, slide, approach 벡터들이며 $p_i = [p_{xi}, p_{yi}, p_{zi}]^T$ 는 위치 벡터이다.

i번째 subtask를 수행하는 동안, end-effector의 카르테시안 주행속도(Cartesian travelling velocity) 성분은 0으로 간주한다. 따라서 i번째 subtask 수행후 매니플레이터가 일정한 위치및 방위 속도를 갖는 다음 직선 경로를 수행하기 위해서는 일정한 가속구간이 필요하며 i+1번째 subtask를 수행하기 직전에는 직선 주행 속도의 감속을 위한 감속구간이 요구된다. 이러한 가속및 감속 구간을 설정하기 위해 중간 경로점들을 다음과 같이 설정한다(그림 4 참조).

$$H_{ai} = [[n_i \ s_i \ a_i] \cdot Rot(K_i, k\theta_i), P_i + k(P_{i+1} - P_i)] \\ = \begin{bmatrix} n_{ai} & s_{ai} & a_{ai} & p_{ai} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$$H_{ai} = [[n_{i+1} \ s_{i+1} \ a_{i+1}] \cdot Rot(K_{i+1}, k\theta_{i+1}), P_{i+1} - k(P_{i+1} - P_i)] \\ = \begin{bmatrix} n_{ai} & s_{ai} & a_{ai} & p_{ai} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

여기서, 벡터 K_i 와 각도 θ_i 그리고 벡터 K_{i+1} 과 각도 θ_{i+1} 은 각각 $Rot(K_i, \theta_i) = [n_i \ s_i \ a_i]^{-1} [n_{i+1} \ s_{i+1} \ a_{i+1}]$ 과 $Rot(K_{i+1}, \theta_{i+1}) = [n_{i+1} \ s_{i+1} \ a_{i+1}]^{-1} [n_i \ s_i \ a_i]$ 로 부터 결정된다. 또한 $k(0 \leq k \leq 1)$ 은 감/가속 구간의 결정을 위한 스칼라값이며 구간 $H_i \sim H_{ai}$ 는 가속 구간 그리고 구간 $H_{i+1} \sim H_{ei}$ 는 감속 구간이 된다.

한편 각 주행 구간에서 매니플레이터 end-effector의 주행 시간들을 t_{ai} , t_{ii} , 그리고 t_{di} 라 하면, 각 구간에서 end-effector의 카르테시안 선형 속도와 가속도 성분들은 다음과 같이 얻어진다.

$$V_{ai} = \frac{V_{ii}}{t_{ai}} t, \dot{V}_{ai} = \frac{V_{ii}}{t_{ai}} \quad (\text{가속구간}) \quad (8)$$

$$V_{ii} = \frac{P_{di} - P_{ai}}{t_{ii}}, \dot{V}_{ii} = 0 \quad (\text{선형구간}) \quad (9)$$

$$V_{di} = -\frac{V_{ii}}{t_{di}} (t - t_{ai} - t_{di}), \dot{V}_{di} = -\frac{V_{ii}}{t_{di}} \quad (\text{감속구간}) \quad (10)$$

여기에서 V_{ai} , V_{ii} , V_{di} 는 각 구간에서의 선형 속도 벡터($\in R^3$)들이며 \dot{V}_{ai} , \dot{V}_{ii} 그리고 \dot{V}_{di} 는 선형 가속도 벡터($\in R^3$)이다.

마찬가지로 각 구간에서의 방위 속도와 가속도들은 다음과 같이 유도된다.

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

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

$$\omega_{di} = -\frac{\omega_{ii}}{t_{di}} (t - t_{ai} - t_{di}), \dot{\omega}_{di} = -\frac{\omega_{ii}}{t_{di}} \quad (\text{감속구간}) \quad (13)$$

여기에서 ω_{ai} , ω_{ii} , ω_{di} 는 각 구간에서의 방위 속도 ($\in R^3$)를 나타내며 $\dot{\omega}_{ai}$, $\dot{\omega}_{ii}$ 그리고 $\dot{\omega}_{di}$ 는 방위 가속도 벡터($\in R^3$)를 나타낸다

또한 각 θ_{ii} 와 회전 기준 벡터 $\rho_{ii} \in R^3$ 는 다음식에 의해 정의된다.^[11]

$$\theta_{ii} = \{ [(n_{di})^T n_{ai} + (s_{di})^T s_{ai} + (a_{di})^T a_{ai} - 1] \} \quad (14)$$

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

한편 매니플레이터는 제한된 구동력을 갖기 때문에 그 end-effector의 선형 속도 $V \in R^3$ 와 가속도

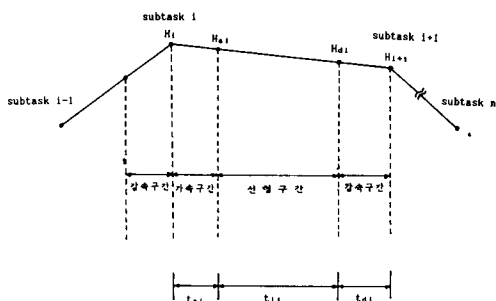


그림 4. 가속 및 감속 구간의 설정
 Fig. 4. Establishment of the acceleration and deceleration sections.

\dot{v} 그리고 방위 속도 $\omega (\in R^3)$ 와 가속도 $\dot{\omega}$ 는 다음과 같은 제한치를 갖는다고 가정할 수 있다.

$$\|v(t)\| \leq k_v \tag{16a}$$

$$\|\dot{v}(t)\| \leq k_a \tag{16b}$$

$$\|\omega(t)\| \leq k_w \tag{16c}$$

$$\|\dot{\omega}(t)\| \leq k_{wa} \tag{16d}$$

상기 제한값과 식(8)~(13)을 이용해 i 번째 subtask와 $(i+1)$ 번째 subtask 사이에서의 최소 주행시간 t_{ii}^* 는 다음과 같이 구할 수 있다.

$$t_{ii}^* = \max\{ \|p_{di} - p_{ai}\| / k_v, |\theta_i| / k_w \} \tag{17}$$

$$t_{ai}^* = t_{di}^* = \{ \|v_{ii}^*\| / k_a, \|\omega_{ii}^*\| / k_{wa} \} \tag{18}$$

$$t_{ii}^* = t_{ai}^* + t_{ii}^* + t_{di}^* \tag{19}$$

여기에서 $(\cdot)^*$ 은 최소 시간을 의미하며 $v_{ii} = (P_{di} - P_{ai}) / t_{ii}^*$, $\omega_{ii} = |\theta_i| / t_{ii}^*$ 이다.

2. 홉필드 신경 회로망에 의한 최소 시간 경로 결정 앞서 언급하였듯이 식 (19)에 의해 결정된 임의의 두 subtask 사이에서의 최소 주행 시간을 거리 변수로 바꾸어 생각하면, 주어진 문제는 기존의 TSP 문제와 일치하게 된다. 따라서 본 절에서는 TSP 해결에 적용된 홉필드 신경망 기법을 이용 주어진 작업 공간에서 최적시간 경로를 결정하고자 한다(그림3참조).

이를 위해 n 개의 subtask들에 대한 하나의 $n \times n$ 순환 행렬을 구성하고 그 element 개수에 해당하는 n^2 개의 뉴론들을 설정한다. 이때 각 뉴론의 입출력 관계는 시그모이드 함수로 표현된다.^[3,7,15]

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

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

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

$$E_1 = D/2 \sum_i \sum_k \sum_j t_{ki}^* V_{ki} (V_{i,i+1} + V_{i,i-1}) \tag{21a}$$

$$E_2 = A/2 \sum_k \sum_i \sum_{j \neq i} V_{ki} V_{kj} + \sum_i \sum_k \sum_{k \neq i} V_{kj} V_{ii} + C/2 (\sum_x \sum_i V_{ki} - n)^2 \tag{21b}$$

$$E = E_1 + E_2 \tag{21c}$$

여기에서 E_1 =모든 subtask를 수행하는데 소요되는 전체 주행시간,

t_{ki}^* =식(19)에서 얻어진 k 번째 subtask와 i 번째 subtask 사이의 최소 주행시간,

E_2 =각 subtask를 한번만 수행케하는 penalty함수, A, B, C, D =임의의 양의 정수.

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.

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

$$= -A \sum_{j \neq i} V_{kj} - B \sum_{k \neq i} V_{ii} - C (\sum_k \sum_j V_{kj} - n^2) - D \sum_i t_{ki}^* (V_{i,i+1} + V_{i,i-1}) \tag{22b}$$

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

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 subtask들의 최적 수행순서를 나타낸다.

IV. 시뮬레이션 및 결과

본 논문에서 사용한 매니올레이터는 6자유도의 회전축을 갖는 PUMA 560로봇 매니올레이터를 사용하였으며, 작업 공간에 5개 subtask 들을 선정하였다.

$$H_1 = \begin{bmatrix} 0.4330 & 0.1250 & 0.8995 & 0.2 \\ 0.2500 & 0.9665 & -0.058 & 0.6 \\ -0.866 & 0.25 & 0.4330 & 0.2 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \tag{23a}$$

$$H_2 = \begin{bmatrix} 0.3368 & 0.5316 & 0.7772 & 0.4 \\ 0.059 & 0.8117 & -0.5810 & 0.8 \\ -0.9397 & 0.2418 & 0.2418 & 0.2 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (23b)$$

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

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

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

상기 5개 subtask들 사이에서 매니플레이터의 주행 경로는 카르테시안 직선 경로로 하였으며 가, 감속 구간의 설정을 위해 $k=0.1$ 로 가정한다. 또한 모의 실험에서 사용한 초기 조건은 $V_{ki}(-1) = 1.0/N + \partial_{ki}$ 로(여기서, N은 모의실험 과정에서의 subtask의 갯수이며 ∂_{ki} 은 임의의 상수 ($< 0.1u_0$)이다.) 하였으며 선형 및 방위의 속도 및 가속도의 상한치들을 다음 값들로 선정하였다.

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

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

또한 가중치 변화의 시상수는 0.4로 하였으며, 각 subtask사이의 최소 주행 시간은 식 (17)~(18)로 부터 구하였다. 그 결과들이 표 1에 정리되었다. 상기 주어진 값들과 각 subtask 사이의 최소 주행시간을 이용 Hopfield 망의 구성이 이루어 졌으며 최소 주행시간을 위한 주행 경로의 결정에 관한 결과들을 그림5~그림9 그리고 표 2를 통해 보여주고 있다. 그림5는 반복학습횟수에 따른 첫번째 뉴런의 출력 상태를 보여주고 있고 그림6, 그림7, 그림8, 그리고 그림9는 각각 두번째 뉴런, 세번째, 네번째, 다섯번째의 뉴런들의 출력 상태를 보여주고 있다. 이 결과들로 부터 반복 학습에 따른 뉴런 출력들이 요구되는 상태로 훌륭히 수렴됨을 알 수 있다.

한편 표 2는 iteration이 끝난후 얻어진 결과들로 최소 주행 시간을 갖는 경로 선정 및 작업순위를 보여주고 있다. 이 경로들에 의해 모든 subtask의 작업을 수행할 경우, 소요되는 시간은 4.9094(sec.)로

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

Table 1. Minimum traveling time between two adjacent subtasks.

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

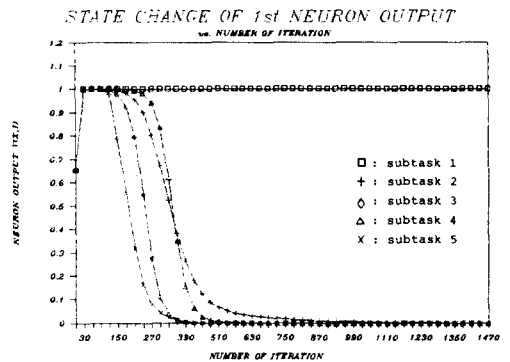


그림 5. 반복 학습에 따른 첫번째 뉴런의 출력 상태
Fig. 5. Output state of the first neuron at each iteration.

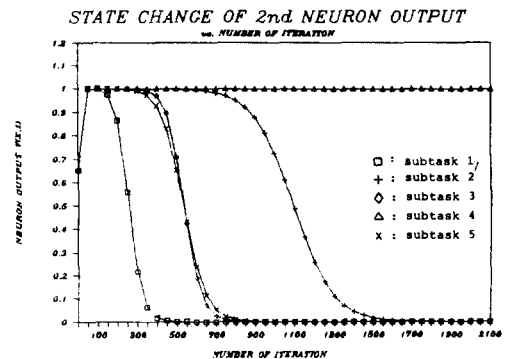


그림 6. 반복 학습에 따른 두번째 뉴런의 출력 상태
Fig. 6. Output state of the 2nd neuron at each iteration.

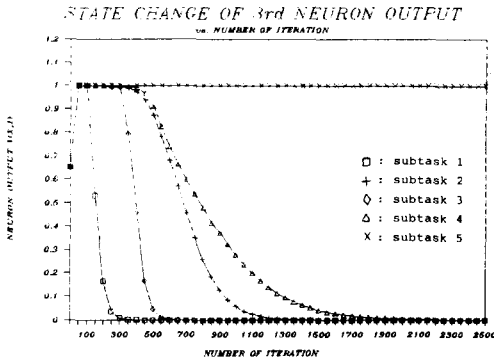


그림 7. 반복 학습에 따른 세번째 뉴런의 출력 상태
Fig. 7. Output state of the 3rd neuron at each iteration.

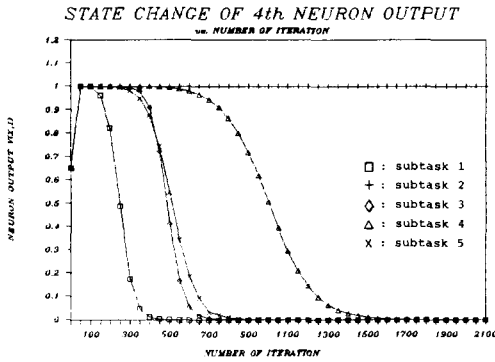


그림 8. 반복 학습에 따른 네번째 뉴런의 출력 상태
Fig. 8. Output state of the 4th neuron at each iteration.

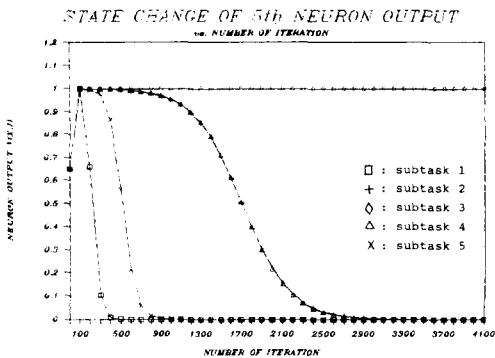


그림 9. 반복 학습에 따른 다섯번째 뉴런의 출력 상태
Fig. 9. Output state of the 5th neuron at each iteration.

표 2. 최소시간 경로를 갖는 뉴런의 최종 상태 및 작업 순위

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

k \ i	position	position	position	position	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.
작업순위 : subtask 1 → subtask 4 → subtask
subtask 2 → subtask 3 → subtask 1

이는 최소 주행 시간이다. 이 결과들은 표 1에 의해 쉽게 입증될 수 있다.

V. 결 론

본 논문에서는 로봇 매니퓰레이터가 TASK상에서의 작업 수행시, 최적시간 경로 설정 문제를 홉필드 신경 회로망을 이용하여 해결했다.

기존의 수리 계획적인 방법에 의해서 작업을 수행할 경우 TASK의 수가 증가함에 따라서 그 작업 수행 경로도 폭발적으로 증가한다. 이 경우 본 논문에서 제한한 신경 회로망을 이용하면 계산량의 감소로 그해를 용이하게 구할 수 있다.

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

앞으로의 연구방향은 각 TASK의 수행시간이 동일하지 않은 경우로의 확장과 동적 특성을 고려한 최적 경로의 결정등을 들 수 있다.

參 考 文 獻

[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. Hildebrandt, "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] Shuilesh. 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 responce 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 Mechancial Manipulators," *Trans. of the ASME*, vol. 102, pp.142-151, June 1981.
- [12] Paul P, "The mathmatics of computer controlled Manipulator," *Proc. of the 1977 joint Automatic Control Conference*, 77CH-1220-3cs, vol. 1, pp. 124-131, 1977.
- [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. Vandebout 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 extention 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.

著 者 紹 介



趙 鉉 讚 (正會員)
 1960年 11月 20日生. 1983年 광운대학교 전자공학과 졸업. 1985년 중앙대 대학원 전자공학과 석사. 1990년 중앙대 대학원 전자공학과 박사과정 수료. 주관심분야는 Robotics, Optimal Control,

Neural Net 등임.



全 洪 兌 (正會員)
 1955년 11월 27日生. 1976년 서울대학교 전자공학과 졸업. 1983년 뉴욕 주립대 전기공학과 석사 1986년 뉴욕 주립대 전기공학과 박사. 1986년~현재 중앙대학교 전자공학과 부교수. 주관심분야는

Robotics, Neural Net., Fuzzy Control 등임.



金 榮 寬 (正會員)
 1965년 7월 7日生. 1988년 한양대학교 전자공학과 졸업. 1990년 중앙대 대학원 전자공학과 석사. 1990년 8월~현재 (주) 대한전선 기술연구소 제 5 연구부 근무중. 주관심분야는 Robotics 및

Neural Net 등임.



李 鴻 奇 (正會員)
 1958년 11월 28日生. 1981년 서울대학교 전자공학과 졸업. 1983년 서울대학교 전자공학과 석사. 1986년 University of Texas at Austin 박사. 1986~1989년 Louisiana State University 조교수 1989년~현재 중앙대학교 제어계측 공학과 조교수. 주관심분야는 Nonlinear Control, Robotics, Neural Net, 등임.