'91 韓國自動制御學術會議論文集 1991. 10.22~24

홉필드 신경회로망을 이용한 다중 로보트의 최적 시간 제어

∘최영길*, 이홍기**, 전홍태***

*부천 전문대학 전자계산기과
**중앙대학교 제어계측공학과
***중앙대학교 전자공학과

Optimal Time Control of Multiple Robot Using Hopfield Neural Network

∘Young-Keel Choi*, Hong-Gi Lee**, Hong-Tae Jeon***

*Dept. of Computer Engineering, Bu-cheon Technical College
**Dept. of Control & Instrumemtation Engineering, Chung-Ang Univ.
***Dept. of Electronic Engineering, Chung-Ang Univ.

ABSTRACT

In this paper a time-optimal path planning scheme for the multiple robot manipulators will be proposed by using hopfield neural network. The time-optimal path planning, which can allow multiple robot system to perform the demamded tasks with a minimum execution time and collision avoidance, 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 rearrange the problem as MTSP(Multiple Travelling Salesmen Problem) and then apply the Hopfield network technique, which can allow the parallel computation, to the minimum time problem. This paper proposes an approch for solving the time-optimal path planning of the multiple robots by using Hopfield neural network. The effectiveness of the proposed method is demonstrated by computer simulation.

1. 서론

많은 소작업으로 구성된 주어진 작업을 산업용 로보트가 수행할 경우 최소 시간(minimum time)내에 완료하는 일은 생산성 향상에 크게 기여한다. 이때 단독 로보트(single robot) 보다는 다중 로보트(multiple robot)들에 의해 주어진 작업을 분담하여 수행할 경우 작업 시간은 크게 단축될것이다. 이 점에 착안 하여 최근 다중로보트를 위한 경로(path)나 궤적(trajectory)계획에 관한 연구가 서서히 시도 되고 있다[1][4].

일반적으로 다중 로보트들에게 주어진 작업들은 그 성격에 따라 다음과 같이 크게 구분 할 수 있다.

1) 소작업들 사이에 수행순서(order of excution)가 미리 규정된 확정작업과,

2) 수행순서가 정해지지않은 비 확정작업

상기 두 부류의 작업을 두대 이상의 다중 로보트에의해 수행할 경우 우선 먼저 고려해야 할 사항은 충돌 회피(collision avoidance)와 작업 시간의 최적화 문제이다. 확정 작업의 경우 주행 순서의 결정은 다중 로보트 주행 경로들의 확정을 의미하기때문에 기존의 연구 결과들을 이용해 어느정도 해결 할 수가있다 [6].

그러나 비 확정 작업의 경우에는 전자와는 다른 접근 방향을 가져야한다. 즉, 충돌 회피를 위한 문제를 먼저 해결하고 그후 경로의 최적화를 시도하거나 혹은 최적화 문제를 먼저 해결하고 그후 충돌 회피를 위한 문제를 해결하도록 해야 한다.

본 논문은 후자의 방향으로 문제의 효율적 해결을 위해 다음의 가정을 갖는다.

1) 각 로보트를 end-effector단에서 하나의 구(sphere)로 간주하고 다중 로보트의 충돌은 end-effector의 충돌로 제한한다.

2) 각 소작업 사이의 경로를 카르테시안 직선 경로(cartesian straight line path)로 한다.

위 가정하에 본논문에서는 다중로보트를 위한 최적시간및 충돌 회피를 위한 하나의 방안을 제안한다. 제안하는 방식은 다음 두 단계로 나누어 이루어진다. 첫째 다중 로보트의 최적시간 문제를 MTSP(Multiple Travelling Salesmen Problem)으로 간주하고 이 해결을 위해 홉필드 신경회로망을 도입한다 [2][5]. 그후 두번째 단계에서는 결정된 최적화 경로에서 발생할 수 있는 다중 로보트의 충돌 영역을 탐색하여 그 회피를 위한 경로를 재구성한다.

한편 본 논문의 구성은 다음과 같다. 2장에서는 MTSP의 문제 정의와 홉필드 신경회로망에의한 해결 방안을 설명하고 3장에서는 홉필드 신경회로망을 이용한 다중 로보트의 최적 시간 경로 생성 방법 그리고 충돌 회피 방식을 제안한다. 그후 4장에서는 본 알고리즘의 효율성을 증명하기 위하여 컴퓨터 시뮬레이션 결과를 보여주고 5장에서는 결론 및 향후의 연구 방향에 대해 서술한다.

2. MTSP문제와 홉필드 신경회로망[5]

MTSP문제는 m명의 판매원들이 기준 도시(base city)를 동시에 출발하여 n개의 도시를 순회 방문할 경우, 각 판매원이 어떤 순서로 방문하면 판매원들에 의한 전체 방문거리가 최소화가 되는가를 결정하는 문제이다. 이때 MTSP문제는 다음과 같은 제한 조건을 갖는다.

1)판매원들은 기준 도시를 동시에 출발하여 n개도시를 정확히 한번만 방문하고 기준 도시로 돌아온다.(각 도시는 한 판매원에의해 한번만 방문되어야한다)

2)m명의 판매원 만큼 순환 경로가 있어야하며 기준도시는 공통이다.

이러한 MTSP문제를 해결할 수 있는 한 방안은 그 문제를 (n+m-1)도시의 단일 판매원에의한 TSP문제로 변환 하는것이다. 즉, 판매원 수만큼 가상도시들을 설정하고 그 가상도시들을 모두 기준 도시와 중복 시킨다. 단일 TSP문제로 변환한후 홉필드 신경회로망에의한 해결 방안은 기존의 방식과 동

일하다.

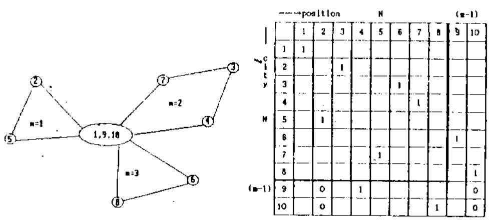
홉필드 신경회로망을 이용 상기 최적화 문제를 해결하기 위해서는 다음과 같이 순환 행렬((n+m-1)x(n+m-1))이 구성되어야한다(그림 1 참조).

1)"1"값이 순환행렬(1,1)위치에 설정되야한다.(이는 기준 도시가 한 판매원에의해서 첫번째 그리고 마지막에 반드시 방문되어야함을 의미함)

2)순환행렬 (i,2)위치와 (i,n+m-1)의 값은 반드시 "0"이어야 한다(i=n+1,...,n+m).

이 제한 조건은 가상도시가 기준 도시를 전후하여 방문되어서는 않된다는것을 의미함.

3)순환 행렬중 실제 도시에 해당하는 행렬에는 정확히 m-1개의 "1"이 있어야하며 마지막 m-1행에서는 없어야한다.(이 제한 조건은 판매원이 실제 도시를 적어도 한번 방문해야함을 의미한다)



(a)판매원의 주행경로 (b)순환 행렬

그림1.순환 행렬의 예(3명의 판매원에의한 8개도시의 경우)

위와 같이 순환 행렬을 구성한후 홉필드 신경회로망 기법에의해 최적해를 발견하기위해 순환 행렬의 각 요소는 하나의 뉴런으로 간주되고 이 뉴런은 다음과 같은 입출력 특성을 갖는다.

$$V_{ki} = \frac{1}{2}\left[1+\tanh(U_{ki}/U_{oo})\right] \tag{1}$$

여기에서 k는 도시를 나타내고 i는 방문 순서를 나타낸다. 그리고 U_{ki}는 입력, V_{ki}는 출력, U_{oo}는 상수값을 나타낸다.(V_{ki} = 1은 k번째 도시가i번째 방문되어야함을 나타낸다)

상기 뉴런으로 구성된 홉필드 신경회로망의 최적 상태를 구하기 위해서는 우선 다음과 같은 에너지 함수가 정의되어야한다.

$$E_p = \frac{1}{2}\sum_{k=1}^{n+m-1}\sum_{\substack{l=1\\l\neq k}}^{n+m-1}\sum_{i=1}^{n+m-1} d_{kl}V_{ki}(V_{l,i+1}+V_{l,i-1}) \tag{2}$$

여기에서 d_{kl}은 도시 l과 도시 k사이의 거리를 나타내며 윗식은 판매원들에 의한 전체 방문거리를 나타낸다.

식 (2)외에 앞에서 설명한 제한 조건들을 만족시키기위해 그 조건 들을 다음과 같이 수식화 한다.

$$E_1 = \frac{1}{2}\sum_{k=1}^{n+m-1}\sum_{i=1}^{n+m-1}\sum_{j=1}^{n+m-1} V_{ki}V_{kj} = 0 \tag{3}$$

$$E_2 = \frac{1}{2}\sum_{i=1}^{n+m-1}\sum_{k=1}^{n+m-1}\sum_{\substack{l=1\\l\neq k}}^{n+m-1} V_{ki}V_{li} = 0 \tag{4}$$

$$E_3 = \frac{1}{2}\left[\sum_{k=1}^{n+m-1}\sum_{i=1}^{n+m-1} V_{ki} - (n+m-1)\right]^2 = 0 \tag{5}$$

$$E_4 = \frac{1}{2}\sum_{k=n+1}^{n+m-1}\sum_{\substack{l=n+1\\l\neq k}}^{n+m-1}\sum_{i=1}^{n+m-1} V_{ki}(V_{l,i+1}+V_{l,i-1}) \tag{6}$$

$$E_5 = \frac{1}{2}\left[\sum_{k=n+1}^{n+m-1}\sum_{i=1}^{n+m-1} V_{ki} - (m-1)\right]^2 \tag{7}$$

$$V_{11} = 1,\ V_{k,n+m-1} = V_{k,2} = 0,\quad k=n+1,n+2,\ldots,n+m-1 \tag{8}$$

식 (2)와 식 (3)~(8)을 이용 홉필드 신경회로망의 최종 에너지 함수는 다음과같이 구할 수 있다.

$$E = E_p + \sum_{\alpha=1}^{5}\lambda_\alpha E_\alpha \tag{9}$$

윗식에서 λ_α는 Lagrange multiplier이며 윗식의 최적화 상태에서의 각 뉴런의 출력 "1"은 각 도시의 해당 최적 방문 순서를 나타낸다.

식 (9)를 최소화 하는 최적해는 BDMM[5]과 Hopfield 및 Tank[2]가 제시한 다음과 같은 탐색 알고리즘에 의해 최종적으로 얻어진다.

$$\begin{aligned}\frac{dU_{ki}}{dt} = \frac{U_{ki}}{\tau} - \Bigg\{ & \sum_{\substack{l=1\\l\neq k}}^{n+m-1} d_{kl}(V_{l,i+1}+V_{l,i-1}) + \lambda_1\sum_{\substack{j=1\\j\neq i}}^{n+m-1} V_{kj} \\ & + \lambda_2\sum_{\substack{l=1\\l\neq k}}^{n+m-1} V_{li} + \lambda_3\left[\sum_{l=1}^{n+m-1}\sum_{j=1}^{n+m-1} V_{lj} - (n+m-1)\right] \\ & + \left\{\lambda_4\sum_{\substack{l=n+1\\l\neq k}}^{n+m-1}(V_{l,i+1}+V_{l,i-1}) + \lambda_5\left[\sum_{l=n+1}^{n+m-1}\sum_{j=1}^{n+m-1} V_{lj} - (m-1)\right]\right\}\Bigg\} \\ & k=n+1,\ldots,n+m-1\end{aligned} \tag{10}$$

$$\forall k, i = 1,2,\ldots n+m-1$$

$$\frac{d\lambda_\alpha}{dt} = +E_\alpha \qquad \forall\alpha = 1,2,\ldots 5 \tag{11}$$

3.다중 로보트의 최적 경로 생성

3.1 최적 시간 경로

3.1.1 소작업 사이에서의 최소 주행시간 결정

작업 공간내의 위치한 i번째 소작업(i=1,2,...,n)의 위치와 방위는 다음과 같은 4x4 homogeneous transformation matrix에 의해 표현될 수 있다.

$$X_i(\in R^{4\times 4}) = \begin{bmatrix} n_i & s_i & a_i & p_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

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, 그리고 approach 벡터들이며 $p_i(=[p_{xi}, p_{yi}, p_{zi}]^T)$ 는 위치 벡터이다.

i 번째 소작업를 수행하는 동안, end-effector의 카르테시안 주행속도 (Cartesian travelling velocity) 성분은 0 으로 간주한다. 따라서 i 번째 소작업 수행후 매니퓰레이터가 일정한 위치및 방위 속도를 갖은 다음 직선 경로를 수행 하기 위해서는 일정한 가속 구간이 필요하며 i+1 번째 소작업를 수행 하기 직전에는 직선 주행 속도의 감속을 위한 감속 구간이 요구된다. 이러한 가속및 감속 구간을 설정하기 위해 중간 경로점들을 다음과 같이 설정 한다(참조 그림 2).

$$X_{ai} = [[n_i \;\; s_i \;\; a_i] \cdot Rot(K_i,\, k\theta_i),\; P_i + k(P_{i+1} - P_i)]$$

$$= \begin{bmatrix} n_{ai} & s_{ai} & a_{ai} & p_{ai} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (13)$$

$$X_{di} = [[n_{i+1} \;\; s_{i+1} \;\; a_{i+1}] \cdot Rot(K_{i+1},\, k\theta_{i+1}),\; P_{i+1} - k(P_{i+1} - P_i)]$$

$$= \begin{bmatrix} n_{di} & s_{di} & a_{di} & p_{di} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (14)$$

여기서, 벡터 K_i와 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 $Rot(K_{i+1}, \theta_{i+1}) = [n_{i+1} \;\; s_{i+1} \;\; a_{i+1}]^{-1}[n_i \;\; s_i \;\; a_i]$ 로부터 결정된다. 또한 $k(0 \le k \le 1)$은 감/가속 구간의 결정을 위한 스칼라 값이며, 구간 $X_i \sim X_{ai}$ 는 가속 구간 그리고 구간 $X_i \sim X_{di}$ 는 감속 구간이 된다.

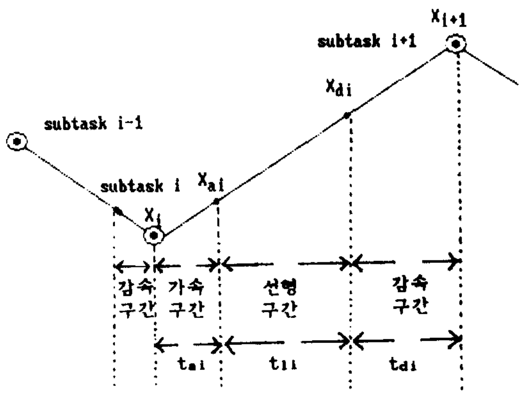


그림 2 가속 및 감속 구간의 설정

한편 각 주행 구간에서 매니퓰레이터 end-effector의 주행 시간들을 t_{ai}, t_{li}, 그리고 t_{di} 라 하면, 각 구간에서 end-effector의 카르테시안 선형 속도와 가속도 성분들은 다음과 같이 얻어 진다.

$$P_{ai} = \frac{P_{li}}{t_{ai}} t, \qquad \dot{P}_{ai} = \frac{P_{li}}{t_{ai}} \quad (\text{가속 구간}) \quad (15a)$$

$$P_{li} = \frac{P_{di} - P_{ai}}{t_{li}}, \qquad \dot{P}_{li} = 0 \quad (\text{선형 구간}) \quad (15b)$$

$$P_{di} = -\frac{P_{li}}{t_{di}}(t - t_{ai} - t_{di}), \qquad \dot{P}_{di} = \frac{-P_{li}}{t_{di}} \quad (\text{감속 구간}) \quad (15c)$$

여기에서 P_{ai}, P_{li}, P_{di}는 각 구간에서의 3x1 선형 속도 벡터들이며 $\dot{P}_{ai}$, $\dot{P}_{li}$ 그리고 $\dot{P}_{di}$는 3x1 선형 가속도 벡터들이다. 마찬가지로 각 구간에서의 각 속도와 가속도들은 다음과 같이 유도된다.

$$\omega_{ai} = \frac{\omega_{li}}{t_{ai}} t, \qquad \dot{\omega}_{ai} = \frac{\omega_{li}}{t_{ai}} \quad (\text{가속구간}) \quad (16a)$$

$$\omega_{li} = \frac{\theta_{li}}{t_{li}} \cdot \rho_{li}, \qquad \dot{\omega}_{li} = 0 \quad (\text{선형구간}) \quad (16b)$$

$$\omega_{di} = -\frac{\omega_{li}}{t_{di}}(t - t_{ai} - t_{li}), \qquad \dot{\omega}_{di} = -\frac{\omega_{li}}{t_{di}} \quad (\text{감속구간}) \quad (16c)$$

여기에서 ω_{ai}, ω_{li}, ω_{di} 는 각 구간에서의 3x1 각 속도를 나타내며 $\dot{\omega}_{ai}$, $\dot{\omega}_{li}$, 그리고 $\dot{\omega}_{di}$는 3x1 각 가속도 벡터이다. 또한 각 θ_{li}와 회전 기준 벡터 $\rho_{li}(\in R^3)$는 다음 식에 의해 정의된다.

$$\theta_{li} = \{[(n_{di})^T n_{ai} + (s_{di})^T s_{ai} + (a_{di})^T a_{ai} - 1]\} \qquad (17a)$$

$$\rho_{li} = \frac{1}{2\sin\theta_{li}} \begin{bmatrix} (a_{di})^T s_{ai} - (s_{di})^T a_{ai} \\ (n_{di})^T a_{ai} - (a_{di})^T n_{ai} \\ (s_{di})^T n_{ai} - (n_{di})^T s_{ai} \end{bmatrix} \qquad (17b)$$

한편 매니퓰레이터는 제한된 구동력을 가지므로 end-effector의 선형속도와 가속도 그리고 각속도와 각가속도는 다음과 같은 제한치를 갖는다고 가정할 수 있다[3].

$$\| P(t) \| \le K_v \qquad (18a)$$

$$\| \dot{P}(t) \| \le K_a \qquad (18b)$$

$$\| \omega(t) \| \le K_w \qquad (18c)$$

$$\| \dot{\omega}(t) \| \le K_{wa} \qquad (18d)$$

상기 제한값과 식 (15)와 (16)을 이용해 i번째 subtask 와 (i+1)번째 subtask 사이에서의 최소 주행 시간 t_i^*는 다음과 같이 구할 수 있다.

$$t_{li}^* = \max\{ \| P_{di} - P_{ai} \| / k_v,\; \| \theta_i \| / k_w \} \qquad (19a)$$

$$t_{ai}^* = t_{di}^* = \{ \| P_{li}^* \| / k_a,\; \| \omega_{li}^* \| / k_{wa} \} \qquad (19b)$$

$$t_i^* = t_{ai}^* + t_{li}^* + t_{di}^* \qquad (19c)$$

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

3.1.2 MTSP에 의한 최적 시간 경로 생성

MTSP문제에서 거리 변수를 식 (15)에 의해 결정된 임의의 두 소작업 사이의 최소 주행 시간들로 바꾸어 주면 로보트들

에 의한 최적 시간 주행 경로를 용이하게 구할 수 있다.

MTSP문제를 이용 다중 로보트의 최적 시간 경로를 구하기 위해 먼저 다중 로보트가 동시에 출발할 하나의 가상 출발점(혹은 가상 기준 소작업)을 설정한다. 이 점을 설정하는 방법은 무수히 많을 수 있지만 가급적 주행 경로의 교차를 최소화 할 수 있는 점으로 가상 출발점을 설정 한다. 이에 본 논문에서는 모든 소작업을 포함하는 작업공간(주로 원형)의 기하학적 중심점을 가상 출발점으로 정한다.(그림 3참조)

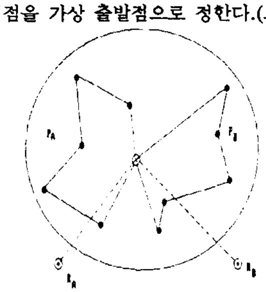


그림3. 가상 출발점의 결정(P_a와 P_b는 각각 로보트 A와 B의 최적 시간 경로)

가상 출발점을 결정한후 각 다중 로보트에 의한 최적 경로는 다음의 단계에 의해 구한다.

단계1)소작업 수(m)과 로보트 수(n)을 이용. (n+m-1) x (n+m-1)의 순환 행렬을 구성한다.(이때 순환 행렬의 각 element는 하나의 뉴론으로 간주함)

단계2)식 (2)~식 (8)의 모든 거리 함수를 앞절에서 구한 소작업 사이의 최소 주행 시간들로 대체함.

단계3)식 (10)과 (11)에서 거리 함수를 최소 주행 시간으로 대체한후 그 탐색 알고리즘에 의해 다중 로보트의 최적 시간 경로를 구함.

3.2 충돌 회피를 위한 경로의 재 구성

앞 절에서 홉필드 신경회로망에의해 최적시간 경로를 생성한후 다음에 고려해야할 사항은 충돌 문제이다. 즉, 다중로보트들이 주어진 경로를 각각 주행할 경우 충돌 가능성이 항상 존재한다. 비교적 충돌 가능성이 높은 지역은 주행 경로들의 교차 부분이라 할 수있다. 그러나 주행 경로가 교차한다해도 교차 부분에 도달하는 소요시간이 다를 경우 충돌은 없게된다.

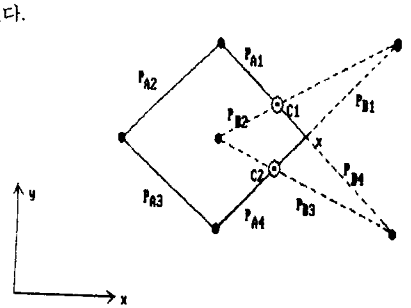


그림4. 주행 경로의 교차(P_{ai}는 로보트 a의 경로이며 P_{bi}는 로보트 b의 경로이다.)

본 절에서는 두 대의 로보트를 예를들어 충돌 가능성 결정과 충돌 회피를 위한 경로 재구성에 관해서 설명한다. 그림 4에서 처럼 두 로보트의 주행 경로들이 결정되었을 경우 일차적으로 충돌 가능성이있는 부분은 P_{a1}과 P_{b2}가 교차하는 교차점 C_1과 P_{a4}와 P_{b3}가 교차하는 교차점 C_2이다.

그러나 교차점 C_1과 C_2에서 충돌여부를 알기 위해서는 교차점의 위치와 기준 소작업에서 각 교차점까지 각 로보트가 주행하는데 걸리는 시간을 구해야한다. 교차점 C_i는 기하학적인 방법에의해서 쉽게 구할 수가있으며 교차점에 도달하는 시간은 소작업 사이의 주행하는 시간을 알면 구할 수가 있다. 두 로보트의 충돌은 교차점에 도달하는 소요시간이 같게되면 발생하게된다. 그렇지 않을경우에는 교차점이있다하더라도 충돌은 없게된다.

두 로보트가 충돌할 가능성이 있는경우 이를 회피할수 있는 방안은 다음 두가지가 있을 수 있다[7].

1)교차점 부근에서 두 로보트 중 한 로보트의 주행속도를 조절하는 방법

2)교차점 부근에서 한 로보트의 경로를 재 구성하는 방법

본 논문은 후자의 방법으로 교차점 부근에 새로운 중간점을 선정해 한 로보트의 경로를 재 구성한다. 이 과정은 그림 5에의해 설명된다.

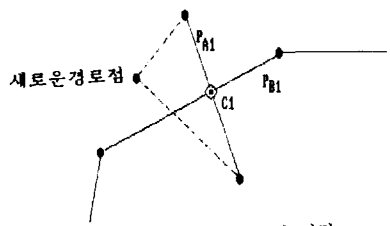


그림5. 교차점 부근에서의 경로점 결정

4.컴퓨터 시뮬레이션

본 논문에서 3대의 고정로보트가 서로 등간격으로 위치하였을 경우 각 로보트의 Work Space 의 반경을 max { r1,r2,r3 } = 500 Cm 으로 하고 이 때 K_v = 0.6 m/s, K_a = 0.3 m/s^2, K_w = 1.0 rad/s, K_{wa} = 0.5 rad/s^2 으로 하였다.

로보트의 위치는 각 그룹에서 도시 "0"으로 표시 하였으며 소작업의 총수는 18개로서 그 위치는 표 1과 같다.

이에 따른 최적 경로는 그림 6으로 나타난다.

그룹 1 CITIES			그룹 2 CITIES			그룹 3 CITIES		
CITY-0	100.00	100.00	CITY-0	900.00	100.00	CITY-0	500.00	900.00
CITY-1	175.00	284.30	CITY-1	627.35	150.21	CITY-1	255.27	724.59
CITY-2	285.22	330.25	CITY-2	576.43	275.00	CITY-2	398.18	627.08
CITY-3	241.87	462.51	CITY-3	578.26	365.32	CITY-3	501.43	597.27
CITY-4	423.18	376.84	CITY-4	723.49	347.20	CITY-4	498.27	478.54
CITY-5	441.56	247.14	CITY-5	762.12	451.35	CITY-5	695.39	623.66
CITY-6	305.23	162.72	CITY-6	856.39	224.95	CITY-6	553.07	771.40
max distance 425.54 min distance 119.41			max distance 417.03 min distance 90.30			max distance 451.54 min distance 107.47		

표 1. 소작업의 위치

5. 결론

본 논문에서는 다중 로보트의 최적 시간 제어및 충돌 회피를 위한 해결 방안을 제시하고 있다. 제안된 방식은 기존의 방식이 필요로 하는 막대한 계산량을 줄이기 위해 병렬 계산

의 가능성을 갖는 홉필드 신경회로망 기법을 도입하고 있다. 향후 홉필드 신경회로망의 하드웨어 구성이 이루어지면 본 알고리즘은 실제 로보트 현장에 도입할 수 있을것으로 판단된다.

앞으로의 과제는 본 논문의 서론에서 가정한 end-effector의 충돌 문제를 로보트의 기구학적 충돌 즉, 링크들에의한 충돌 문제로 확장 시켜 연구하는 일이다. 또한 로보트의 정확한 동특성을 고려하여 최적 궤적을 생성하는 것도 장차 연구해야 할 중요한 과제이다.

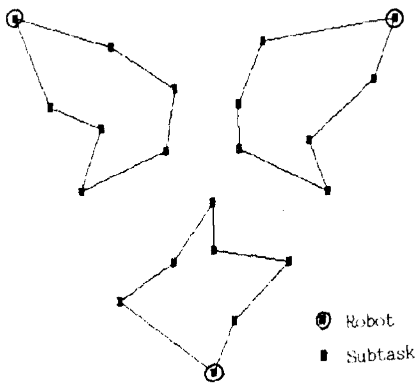


그림 6. 최적 경로

참고문헌

[1] Stephen Cameron, "Efficient intersection Tests for objects defined constructively," IEEE.Int.Conf. on Robotics and research, pp.3-25, 1989

[2] J.J. Hopfield and D.W. Tank, "Neural computation of Decision in Optimization problem," Bio- Cybernetics, Vol. 52, pp.141-152, 1985

[3] J.Y.S.Luh and C.S.Lin, "Optimum path planning for mechanical Manipulators," Trans. of the ASME, Vol.102,pp.142-151,June, 1981

[4] C.MacNish and F. Fallside, "Temporal Reasoning : A solution for multiple Agent collision Avoidance," IEEE. Int. Conf. on Robotics and research, pp.494-499 ,1990

[5] E. Wacholder,J.Han and R.C.Mann, "An extension of the Hopfield - Tank model for the solution of the multiple Travelling Salesmen Problem," IEEE.Int.Conf. on Neural Network, Vol. 2, pp.305-324,1988

[6] 변증남 "신경 최적화 회로망을 이용한 두대의 로보트를 위한 최소 시간 경로 계획," 대한 전자공학 논문지, 제 27권, pp.44-52, 1990

[7] 이범희 "두 매니퓰레이터의 충돌회피를 위한 동작 단계수 정법," 대한전자공학회 논문지,제 25권, pp.35-40, 1988

[8] 조현찬, 김영관, 전홍태, 이홍기, "홉필드 신경회로망을 이용한 로보트 매니퓰레이터의 최적 시간 경로계획," 대한전자공학회 논문지, 제 27권, 9호,pp.54-61, 1990