

골격지도에서 개별 목표를 갖는 두 이동로봇의 충돌 회피 알고리즘

A Collision Avoidance Algorithm for Two Mobile Robots with Independent Goals in Skeleton Map

양 동 훈, 홍 석 교*

(Dong-Hoon Yang and Suk-Kyo Hong)

Abstract : This paper proposes a collision avoidance algorithm for two mobile robots with independent goals in a same workspace. Using skeleton map, an environment is presented as a graph consisting of nodes and arcs. Robots generate the shortest set of paths using the combination of objective functions of the two robots. Path for collision avoidance of a robot can be selected among three class; the shortest path, detour, paths with a waiting time at safety points around crossing points. Simulation results are presented to verify the efficiency of the proposed algorithms.

Keywords : mobile robot, path planning, skeleton map, collision avoidance

I. 서론

이동로봇은 공장에서 선으로 표시된 경로를 따라 이동하는 자율이동로봇(AGV)과 최근 공공장소의 안내서비스, 가정의 청소로봇 등 정해진 경로를 따르지 않고 주행하는 로봇 등으로 그 기능과 활용분야가 다양해져 많은 연구가 진행되고 있다. 한 공간에 여러 로봇이 작업을 위해 이동하고 있는 경우 로봇 간의 충돌을 예상할 수 있다. 이 충돌을 막기 위한 회피 방법이 많이 연구되고 있으며 그 방법은 크게 중앙집중형(centralized) 제어와 분산형(decoupled) 제어의 두 가지 방법으로 나눌 수 있다[1].

중앙집중형 방식은 개별 로봇의 배치(configuration)공간을 직교곱(cartesian product)을 통해 구성된 결합 배치공간을 이용하여 충돌이 발생하지 않을 경로들을 생성하는 방법이며 분산형 방식은 각 로봇의 경로를 각각 생성한 후 그 경로들의 충돌여부를 판단하여 경로 수정 및 로봇의 속도조정 과정(coordination)을 통해 충돌을 회피하게 하는 방식이다. 이 두 방식의 성능 차이는 경로생성 과정의 연산복잡성과 경로를 정확히 생성할 수 있는가를 비교를 통해 알아 볼 수 있다. 전자는 보다 완전한 경로를 찾아낼 수 있는 장점이 있는 반면, 후자는 연산과정을 간략히 할 수 있다는 장점이 있다. 전자의 장점에도 불구하고 구현의 어려움과 큰 연산을 요구하는 문제로 현재 분산형 방식이 많이 연구되고 있다[2-4].

경로생성 방법과는 별도로 여러 로봇을 제어하는 방식을 다시 중앙제어 방식과 개별제어 방식의 두 가지로 나눌 수 있다. 중앙제어 방식은 FIRA 방식의 축구로봇에서와 같이 로봇들의 움직임을 관찰하고 동작을 지시하는 연산장치를 따로 두는 방식이며[5], 개별제어 방식은 각 로봇들이 각자의 경로를 생성하고 주행하며 다른 로봇을 감지했을 때, 서로

통신을 통하여 정해진 규칙에 의해 충돌을 회피하는 방법이다. 개별제어 방식에서는 로봇의 위치를 파악하는 것과 로봇 간의 통신문제, 장애물과 로봇의 구별 등의 분야에 많은 연구가 진행되고 있다[6,7]. 또한 여러 로봇간의 충돌회피에 관한 연구 이외에 군집주행제어, 협력제어에 관한 연구가 현재 진행되고 있다[8-10]. LaValle은 각 로봇의 개별 성능지수를 결합하는 방식으로 충돌문제를 해결하였으나, 이는 중앙집중 방식을 이용하여 로봇들이 같은 시간에 각자 위치에서 출발하여 충돌이 발생하지 않도록 각자의 경로 상에서 주행 유무를 결정하여 충돌을 회피한다[11,12]. 여러 이동 로봇의 충돌 회피에 관한 많은 연구에서 속도 조절, 우선권배정 문제 등을 이용하여 충돌을 회피하는 방법 등이 제시된 바 있으나, 좁은 복도에서 만난 로봇의 경로계획 문제는 위와 같은 방식으로 해결되지 않는다. 여러 이동로봇의 충돌회피 문제는 무인자동차의 연구에 응용될 수 있으며, 실험용 자동차 수준에서 실험이 수행되었다[13].

본 논문에서는 같은 작업 공간에서 각기 다른 출발점과 도착점을 갖는 이동 로봇이 주행 중에 가까운 거리에서 서로를 감지하였을 때 충돌하지 않고 안전한 경로를 생성하여 회피하게 하는 분산형 방식의 충돌회피 알고리즘을 제안하고자 한다. 기존의 분산형 제어방식에서는 로봇이 출발하기 전에 개별 생성된 경로를 조정하여 충돌을 회피하는데 반해, 본 논문에서 제안하는 방식은 개별 생성된 경로를 따라 주행하는 도중에 다른 로봇을 만나게 되었을 때, 통신을 통해 경로를 교환하고 충돌이 발생하지 않도록 하는 최적 경로를 찾아 주행하는 방식이다. 이 방법의 장점은 여러 로봇이 한 공간에서 이동해야 할 때, 로봇들이 같은 시간에 출발해야 한다는 가정이 필요하지 않다는 점이다. 또한 로봇들이 같은 작업 공간에 있다고 하여도 서로 충돌하는 경로만을 갖지는 않기 때문에 조정과정이 불필요한 경우도 있을 수 있다. 기존의 방법은 이 경우에도 충돌경로가 발생하는 경우와 같이 알고리즘이 수행되어야 하나, 이 논문에서 제안하는 알고리즘은 불필요한 충돌회피 연산이 수행되지 않으므로 보다 효율적이다.

* 책임저자(Corresponding Author)

논문접수 : 2005. 12. 15., 채택확정 : 2006. 4. 3.

양동훈, 홍석교 : 이주대학교 전자공학부

(yann@ajou.ac.kr/skhong@ajou.ac.kr)

※ 본 연구는 과학기술부 특정기초연구지원사업(R01-2006-000-11373-0)의 지원에 의하여 연구되었음.

이 논문은 다음과 같이 구성되어 있다. 2장에서는 경로생성을 위한 새로운 골격지도 생성방법을 제안하고, 3장에서는 이 논문에서 제안하는 다개체 이동로봇의 충돌회피 경로계획 알고리즘을 설명한다. 4장에서는 제안된 알고리즘의 유효성을 시뮬레이션을 통해 보이고, 5장에서 결론이 제시될 것이다.

II. 제안된 골격지도와 경로생성 방법

1. 기존의 골격지도

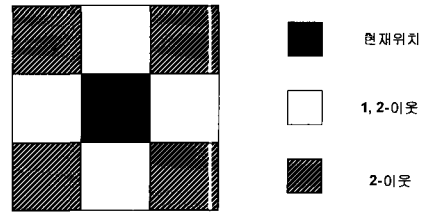
로봇의 작업공간에는 다각형 장애물들이 존재하여 각 로봇은 이를 피하면서 원하는 목표지점까지 이동하여야 한다. 각 로봇의 경로는 장애물과의 충돌을 방지하기 위해 안전한 경로 생성을 위한 로드맵인 골격지도(skeleton map)를 이용하여 장애물과 멀리 떨어진 곳에 생성된다. 우선 골격지도는 파면확장(wave-front expansion)방법을 이용하여 생성된다[1]. 작업공간은 R^2 공간이며, 그 지도는 격자로 나누어져 있고 q 는 한 격자의 (x, y) 좌표를 뜻한다. 이 방법을 이용하여 생성된 골격을 그림 2(a)에 나타내었다. 골격선을 이루는 격자들의 집합을 S 로 표현한다.

2. 제안된 골격지도 생성 알고리즘

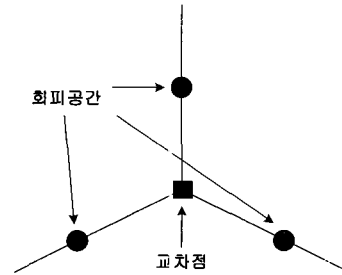
이산화된 골격지도는 많은 성분을 갖는 경로를 생성하는데, 이 정보의 양은 로봇들이 교환하여 경로비교 및 회피경로를 생성하기에 지나치게 클 수 있다. 따라서 골격지도의 정보를 줄이기 위해 교차점과 교차점 부근의 자(子)노드(Node)를 이용한 새로운 골격지도의 생성을 제안하고자 한다. 새로운 골격지도의 생성 방법은 다음과 같다.

- 1) 골격선 상의 모든 교차점 주변 자노드 검색
- 2) 서로 연결되는 이웃 교차점 쌍을 검색
- 3) 노드 간의 연결
 - 3-1. 교차점과 교차점의 자노드 간의 연결
 - 3-2. 서로 연결되는 이웃 교차점의 자노드 간의 연결
 - 3-3. 출발점과 도착점을 골격선에 연결

모든 교차점의 골격선 상의 자노드는 표 1의 알고리즘을 통해 검색된다. 먼저 골격선상에서 3갈래 이상의 길이 만나는 교차점을 검색한다. 이렇게 검색된 교차점들을 반복리스트 L_0 에 저장하고 이 리스트 상의 모든 교차점에서부터 골격선을 따라 퍼져나가며 검색을 시작한다. 여기서, 2-이웃의 의미는 다음과 같다. 2차원 격자로 표현된 지도상에서 한 지점에서 이동할 수 있는 지점은 8개가 된다. 이동이 좌, 우, 상, 하의 네 방향으로만 가능하다고 하면 8개의 지점 중에서 한 번에 이동할 수 있는 지점은 네 곳이 된다. 그리고 다시 두 번 안에 이동할 수 있는 지점은 네 곳이 더 늘어 8곳이 된다. 이를 2-이웃이라 한다(그림 1(a)). 단, 여기서 출발 지점과 점 또는 선으로 접하지 않은 지점은 두 번에 이동할 수 있어도 이웃이 아니다. 이를 일반화하여 m 차원 격자공간에서 한 지점의 이웃은 다음과 같이 분류한다. $2m$ 개의 1-이웃을 갖고, $2m^2$ 개의 2-이웃을, 그리고 $3^m - 1$ 개의 m -이웃을 갖는다. 반복리스트 L_0 에 저장된 교차점들의 2-이웃인 q 에 $U(q)$ 를 1을 부여하고 $O(q)$ 에 q 를 저장한다. 이 2-이웃들을 L_1 에 저장한다. 이를 β 번 반복하게 된다. 이 반복 후 도



(a) 격자에서의 이웃



(b) 교차점과 회피공간

그림 1. 알고리즘 설명.

Fig. 1. Descriptions of algorithm.

달하는 지점들이 회피공간으로 이용될 것이다. 이 지점들을 자노드로 정의하는데, 각 자노드가 어느 교차점으로부터 파생되었는지 알려주는 것이 $O(q)$ 이다. 그림 1(b)는 교차점과 그 자노드로 이루어진 회피공간을 보여준다. β 는 교차점과 회피공간 사이의 거리와 관계가 있으므로 이를 작게 설정할 경우 교차점을 지나는 로봇과 회피공간에 있는 로봇이 충돌할 수도 있으므로 충분한 크기로 설정하여야 한다. i 는 교차점에서 골격선을 따라 이동한 횟수를 나타낸다. $U(q)$ 는 임시적으로 사용되는 q 에서의 위치준위를 나타낸다. 이 과정에서 교차점으로부터 가까운 위치에 다른 교차점이 존재하는 경우 위 반복과정에서 $U(q) \neq M$ 인 q 가 발생하므로 이는 다음 반복리스트 L_i 에 저장되지 않는다. 따라서 이는 자노드로 간주되지 않는다. C_{cr} 은 교차점의 집합이다.

충돌회피에 이용되는 교차점의 자노드 뿐 아니라, 목표점까지 이동하기 위해 각 교차점을 잇는 경로를 생성하기 위해서 서로 연결이 가능한 교차점들을 검색하여야 한다. 표 2에 연결된 교차점의 검색 알고리즘을 제시하였다. $Q_{connect}$ 는 한 교차점에 연결된 교차점들의 집합을 나타낸다.

표 1과 2의 알고리즘을 통해 검색된 교차점과 교차점의 자노드, 그리고 주어진 로봇의 출발점과 도착점이 노드로 정의되어 경로생성에 이용된다. 표 3에서는 교차점과 검색된 교차점들의 자노드, 서로 연결될 수 있는 교차점들의 자노드를 연결하고, 출발점과 도착점을 골격에 연결하여 골격지도를 완성하는 방법을 제시하였다. 골격지도에서는 여러 경로가 만나는 교차점이 지도의 위상기하학적(topological) 특성을 결정한다. 위의 두 지도는 위상기하학적 특성이 달라지므로 위상동형(homeomorphism)은 아니지만 두 지도를 이용하여 생성하는 경로는 서로 호모토픽(homotopic)이다. 그림 2에 4개의 장애물을 갖는 사각형태의 환경에 기존의 골격지도와 위치정보의 양을 줄인 제안된 골격을 이용한 지도를 나타내었다.

표 1. 모든 교차점의 골격선상에 존재하는 자노드 검색 알고리즘.

Table 1. Search algorithm for child-nodes that exist on the skeleton of all the crossing points.

1	$\forall q \in S$ 에 대해
2	$U(q) \leftarrow M$ (M 은 매우 큰 수)
3	$\forall q \in S$ 에 대해
4	3개 이상의 2-이웃을 가지면
5	$O(q) \leftarrow q$
6	$U(q) \leftarrow 0$
7	q 를 Q_{cr} 에 추가
8	q 를 L_0 에 추가
10	$i = 0, 1, 2, \dots, \beta$ 에 대해 반복
11	$\forall q \in L_i$ 에 대해
12	q 의 모든 2-이웃 중 S 에 속하는 q' 에 대해
13	$U(q') = M$ 이면
14	$U(q') \leftarrow U(q') + 1$
15	$O(q') \leftarrow O(q)$
16	q' 를 L_{i+1} 에 추가
17	$\forall q \in S$ 에 대해
18	$U(q) = \beta$ 이면 q 를 $Q_{child}(O(q))$ 에 추가

표 2. 연결된 교차점의 검색 알고리즘.

Table 2. Search algorithm for the crossing points connected each other.

1	$\forall q \in S$ 에 대해
2	$U(q) \leftarrow M$ (M 은 매우 큰 수)
3	$q \in Q_{cr}$ 에 대해
4	$O(q) \leftarrow q$
5	$U(q) \leftarrow 0$
6	q 를 L_0 에 추가
7	$i = 0, 1, 2, \dots, L_i$ 가 비어질 때까지 반복
8	$\forall q \in L_i$ 에 대해
9	q 의 모든 2-이웃 중 S 에 속하는 q' 에 대해
10	$U(q') = M$ 이면
11	$U(q') \leftarrow U(q') + 1$
12	$O(q') \leftarrow O(q)$
13	$q' \notin Q_{cr}$ 이면
14	q' 를 L_{i+1} 에 추가
15	$q' \in Q_{cr}$ 이면
16	q' 를 $Q_{connect}(O(q))$ 에 추가

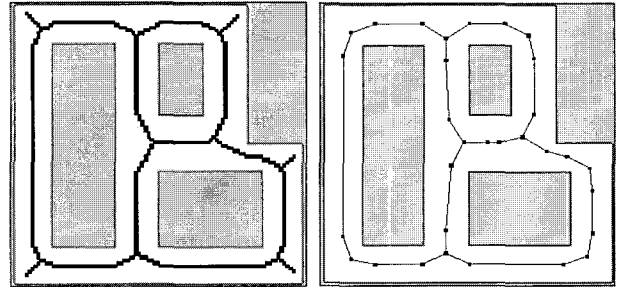
3. 경로생성

작업공간에서 로봇의 위치는 x 로 표현된다. 경로계획의 목적은 출발점부터 목표점까지 장애물과의 충돌로부터 안전한 경로 $\tau: [0, 1] \rightarrow C_{free}$ 를 생성하는 것이다. 여기서 C_{free} 는 충돌로부터 안전한 작업공간을 나타낸다. 이때, $\tau(0) = x_{init}$, $\tau(1) = x_{goal}$ 이다. 이동로봇의 비용은 다음과 같이 표현된다.

표 3. 노드 간의 연결 알고리즘.

Table 3. Connection algorithm between the nodes.

3-1	1 $\forall q \in Q_{cr}$ 에 대해
	2 q 와 $\forall q' \in Q_{child}(q)$ 를 연결
3-2	1 $\forall q \in Q_{cr}$ 에 대해
	2 $\forall q' \in Q_{child}(q)$ 에 대해
	3 $\forall q'' \in Q_{child}(q')$ 에 대해, ($q'' \in Q_{connect}(q)$)
	4 $d(q', q'')$ 계산
	5 최소 $d(q', q'')$ 를 갖는 (q', q'') 을 연결
3-3	1 출발점 q_s 과 도착점 q_g 이 주어졌을 때,
	2 $q_s' = \arg \min_{\forall q \in S} d(q_s, q)$
	3 $q_g' = \arg \min_{\forall q \in S} d(q_g, q)$
	4 $\forall q_{scr} \in Q_{connect}(q_s')$ 에 대해
	5 $\forall q_{nscr} \in Q_{child}(q_{scr})$ 에 대해
	6 $d(q_s, q_{nscr})$ 계산
	7 최소 $d(q_s, q_{nscr})$ 를 갖는 (q_s, q_{nscr}) 을 연결
	8 $\forall q_{ger} \in Q_{connect}(q_g')$ 에 대해
	9 $\forall q_{ngcr} \in Q_{child}(q_{ger})$ 에 대해
	10 $d(q_g, q_{ngcr})$ 계산
	11 최소 $d(q_g, q_{ngcr})$ 를 갖는 (q_g, q_{ngcr}) 을 연결



(a) 기존의 골격지도 (b) 제안된 골격지도

그림 2. 골격지도.

Fig. 2. Skeleton maps.

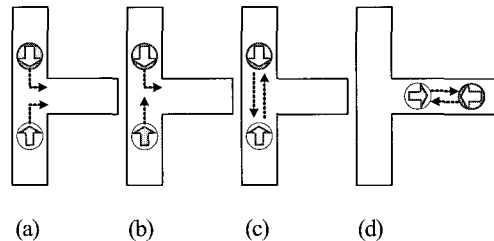


그림 3. 두 로봇이 만나는 여러 경우.

Fig. 3. Various cases where two robots meet each other.

$$L(x_1, \dots, x_{K+1}, u_1, \dots, u_K) = \sum_{k=1}^K l_k(x_k, u_k) + l_{K+1}(x_{K+1}) \quad (1)$$

여기서, l_k 는 한 구간의 비용, l_{K+1} 은 최종비용이다. 최적 경로 계획 문제는 비용을 다음과 같이 뚝으로써 간단히 표현할 수 있다.

$$l_k = d(x_k, x_{k+1}), \quad \forall k \in \{1, \dots, K\}, \quad (2)$$

$$\begin{cases} l_{K+1}(x_{K+1}) = 0, & \text{if } x_{K+1} = x_{goal} \\ l_{K+1}(x_{K+1}) = \infty, & \text{otherwise} \end{cases}$$

이 비용함수를 최소화하는 경로를 동적 계획법(dynamic programming)을 이용하여 생성한다[14]. 각 경로의 비용은 목표점까지의 비용함수(cost-to-go)를 누적시키는 방식으로 구한다.

III. 이동로봇의 충돌회피 알고리즘

1. 두 로봇이 만나는 여러 상황

두 로봇이 각 경로를 따라 이동하는 중에 서로 만날 수 있는 상황을 그림 3의 (a), (b), (c)와 같이 교차점을 사이에 두고 만나는 경우와 그림 3(d)와 같이 좁은 길에서 만나는 경우로 나눌 수 있다. 다시 교차점에서 만나는 경우를 세 가지로 나누었는데, 그림 3(a)는 서로 다른 길에서 오던 로봇이 교차점을 지나 같은 길로 가려 하는 경우이고, 그림 3(b)는 한 로봇이 다른 로봇이 온 길로 가려 하고 다른 로봇은 이 로봇이 오던 길이 아닌 다른 길로 가려는 경우이다. 그림 3(c)는 각 로봇이 상대 로봇이 오던 방향으로 가려 하는 경우이다.

2. 이동로봇의 충돌회피 알고리즘

N대의 로봇이 존재할 때, x^i 위치에 있는 i 번째 로봇을 $A_i(x^i)$ 로 나타내고, 이 로봇이 위치할 수 있는 공간을 X^i 라 한다. 두 로봇 ($N = 2$) A_i, A_j 의 위치를 다음과 같은 상대 공간으로 일반화할 수 있다.

$$X = X^i \times X^j \quad (3)$$

\times 기호는 직교곱을 뜻한다. 이때, 두 로봇 A_i, A_j 가 충돌하게 되는 공간을 다음과 같이 정의한다.

$$X_{coll}^{ij} = \{x \in X \mid A_i(x^i) \cap A_j(x^j) \neq \emptyset\} \quad (4)$$

이 논문에서는 이를 이용하여 다개체 이동로봇의 충돌회피 문제를 다음과 같이 정의한다.

문제: 주행 중 서로를 감지한 두 이동로봇이 $x \notin X_{coll}^{ij}$ 인 요소로만 구성된 최단 경로 쌍을 생성하는 것.

여기서, 두 로봇이 충돌회피를 위해 생성하는 경로 $\gamma \in \Gamma$

들은 그림 4와 같이 세 가지로 분류된다.

$\gamma_{shortest}$: 현재 위치에서 목표점까지 가장 짧은 경로(그림 4(a))

γ_{detour} : 상대로봇의 위치를 지나지 않는 가장 짧은 우회 경로(그림 4(b))

γ_{wait} : 상대로봇과 충돌하지 않을 수 있는 공간(S)에 회피하는 경로와 그 공간으로부터 목표점까지의 경로를 더한 경로(그림 4(c))

경로 $\gamma_{shortest}$ 와 γ_{detour} 는 로봇 당 한 개씩으로 결정된다. 그러나 γ_{wait} 의 경우에는, 로봇이 위치하는 골격선과 다른

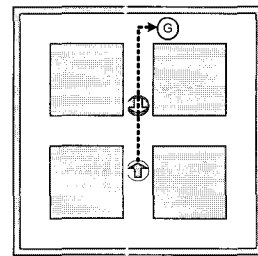
골격선과의 연결 상황에 따라 경로의 개수가 달라진다. 그림 4의 (c)에서 두 로봇 중 아래에 위치한 로봇을 살펴보면, 현재 로봇이 위치한 복도는 위쪽으로 세 개의 복도와 만나고 아래쪽으로 두 개의 다른 복도와 만난다. 즉 두 개의 교차점이 현재 로봇이 위치한 복도에 존재함을 알 수 있다. 따라서 이 로봇이 회피할 수 있는 총 자노드의 개수는 일곱 개가 된다. 위의 로봇 또한 같은 수의 자노드를 가지므로 회피할 수 있는 경로 쌍은 49개임을 알 수 있다. γ_{wait} 의 경로를 따를 때에는 로봇이 자노드에 도달한 후, 상대로봇이 이 교차점을 지나가는 경로를 따르는 것으로 가정하고 충돌을 피해 로봇이 교차점을 지나갈 동안 그 위치에서 대기하는 시간이 필요하다. 이 시간을 다음과 같이 둘 수 있다.

$$T_{wait} = \frac{2d(x^i, \text{교차점})}{\|v^i\|} \quad (5)$$

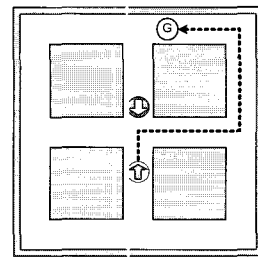
$d(x^i, x^j)$ 는 x^i 와 x^j 의 직선거리를 의미한다. 이 과정을 통해 생성된 회피경로들의 비용이 다음과 같이 정의된다.

$$L^i(\gamma) = \sum_{k=1}^K l_k^i(x_k^i, u_k^i) + l_{K+1}^i(x_{K+1}^i) + T_{wait} \quad (6)$$

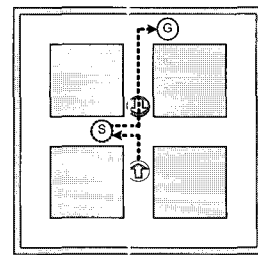
여기서, 각 항은 다음과 같다.



(a) 가장 짧은 경로



(b) 가장 짧은 우회로



(c) 안전지역을 거치는 경로

그림 4. 충돌회피를 위한 여러 경로.
Fig. 4. Various paths for collision avoidance.

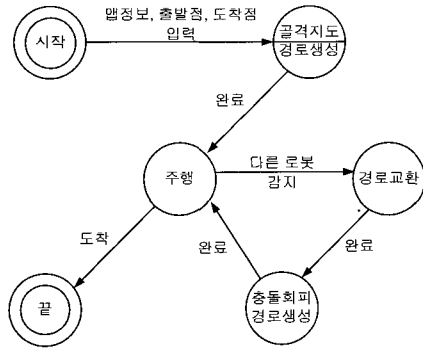


그림 5. 로봇의 주행 과정.

Fig. 5. Traveling process of a robot.

$$l_k^i = \begin{cases} d(x_k^i, x_{k+1}^i), & k \neq K+1 \\ 0, & k = K+1 \end{cases}$$

$$l_{K+1}^i(x_{K+1}^i) = \begin{cases} 0, & \text{if } x_{K+1}^i = x_{goal}^i \\ \infty, & \text{otherwise} \end{cases}$$

$$T_{wait} = \begin{cases} 0, & \gamma \neq \gamma_{wait} \\ (5), & \gamma = \gamma_{wait} \end{cases}$$

여기서 α 는 임의의 상수로 정의할 수 있다. 정의된 각 경로들의 비용을 이용하여 두 로봇의 충돌회피경로의 비용을 계산한다. 이때 두 로봇의 개별 경로의 최소 비용보다는 두 로봇의 회피과정에서 발생하는 비용을 합하여 그 비용이 최소가 되도록 한다. 여러 로봇의 협력 작업 및 동시 작업에서 그 비용을 고려할 때는 여러 로봇의 개별 비용을 다음과 같이 스칼라화한 비용을 이용한다[11].

$$H(\gamma^i, \gamma^j) = \beta_i L^i(\gamma^i) + \beta_j L^j(\gamma^j) + c^{ij} \quad (7)$$

$$c_k^{ij} = \begin{cases} 0, & \forall (x_k^i, x_k^j) \notin X_{coll} \\ \infty, & \exists (x_k^i, x_k^j) \in X_{coll} \end{cases}$$

β_i, β_j 는 상수이나 각 로봇의 중요도가 같다고 보면 같은 값으로 설정할 수 있다.

충돌상황을 회피해야 하는 로봇의 주행 과정을 그림 5에 나타내었다. 먼저 로봇에 작업공간에 대한 이산화된 맵정보가 주어지고, 출발점과 도착점이 입력되면 로봇은 [1]의 방법을 이용하여 골격지도를 생성한다. 이 골격지도를 제안된 골격지도 생성 알고리즘을 이용하여 새 골격지도를 만든다. 이 지도에 동적계획법을 이용하여 최단 경로를 생성한 후, 주행을 한다. 각 경로를 따라 주행하는 두 로봇이 만나게 되면, 각 로봇은 그림 4의 경로들을 생성한 후 교환한다. 두 로봇은 스칼라화를 통해 최단 경로 쌍을 찾고 그 경로들을 따라 주행을 한다.

IV. 시뮬레이션 결과

이 논문에서 제안한 충돌회피 알고리즘이 앞에서 정리한 여러 상황에 적용이 가능한가와 생성된 회피경로의 타당함을 살펴보기 위하여 Matlab을 이용한 시뮬레이션을 수행하였다. 이 시뮬레이션에 적용된 환경은 사각형태의 벽과 장애물

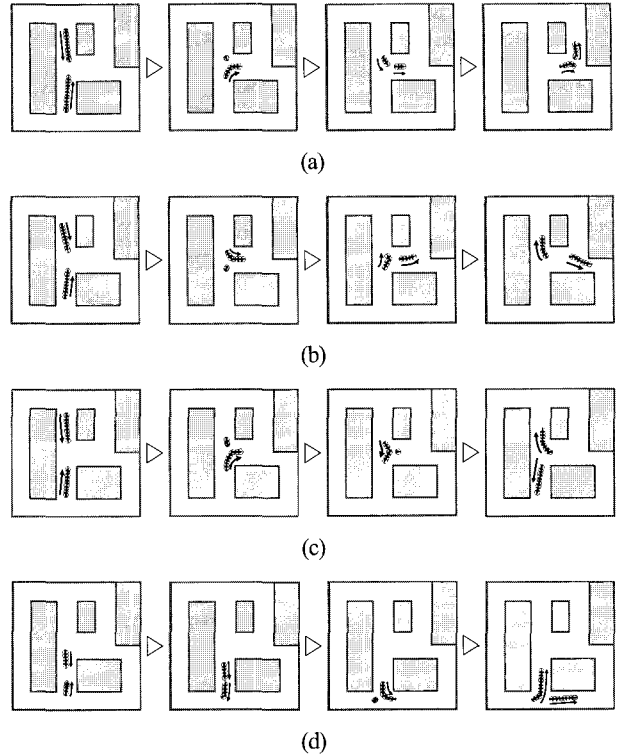


그림 6. 두 로봇의 충돌회피 과정.

Fig. 6. Sequences of collision avoidance of two robots.

로 이루어져 있고, 이 논문에서 제안한 방법으로 생성된 골격지도는 그림 2(b)에 나타내었다. 이 골격지도를 살펴보면 9개의 교차점이 존재함을 알 수 있으며, 이 교차점들의 자노드를 이용하여 두 로봇이 충돌을 회피하게 될 것이다.

그림 3에 정리한 충돌 가능 상황을 회피하는 시뮬레이션이 그림 6에 나타나 있다. 첫 번째는 서로 다른 길에서 이동하던 두 로봇이 교차점을 사이에 두고 만나 같은 길로 주행하려는 상황이다. 이 문제를 해결하기 위해서는 교차점에 가까운 로봇이 먼저 이동하고 다른 로봇이 뒤따라야 될 것이다. 그림 6(a)에 제안된 알고리즘을 사용한 시뮬레이션 결과가 나타나 있다. 그림 6(b)에서는 두 번째 경우로 아래의 로봇이 위의 로봇이 오던 길로 가려 하고 위의 로봇은 오른쪽의 길로 가려 하는 상황을 보여 준다. 위의 로봇이 길을 비워줘야 하는 경우이므로 먼저 오른쪽 길로 이동하게 된다. 만약 아래의 로봇이 오른쪽 길로 이동하면 위의 로봇은 위의 로봇은 길을 잃게 되므로 우회경로를 선택하게 되고 이는 보다 큰 비용을 필요로 하므로 이 경로는 택하지 않는다. 그림 6(c)는 세 번째 경우로 두 로봇이 서로 자리를 바꾸어야 하는 상황이므로 교차점에 가까운 로봇이 오른쪽 길로 이동하여 다른 로봇이 교차점을 지나길 기다린 후, 다시 자신의 경로를 따라 이동한다. 그림 6(d)에서는 네 번째 경우로 좁은 길에서 두 로봇이 만나 교차점에서 가까운 아래의 로봇이 후진하여 교차점의 자노드로 이동하여 대기한 후, 위의 로봇이 지나간 후 다시 자신의 경로를 따라 이동하는 상황을 보여주고 있다.

각 로봇이 충돌 회피를 위해 생성한 회피 경로가 위 시뮬레이션에서는 8개이고 이때 두 로봇의 회피를 위한 경로 쌍

은 64개가 된다. 선택된 모든 경로 쌍은 최단 비용을 필요로 하며, 이 경로를 따라 두 로봇은 충돌을 회피하는 것을 볼 수 있다.

V. 결론

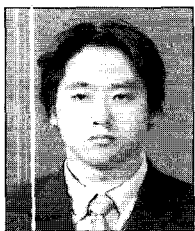
이 논문에서는 같은 작업공간에서 계획된 경로를 따라 이동하고 있는 두 대의 이동로봇이 서로를 감지하였을 때, 충돌하지 않는 최단거리의 경로들의 쌍을 찾아 주행하도록 하는 알고리즘을 제안하였다. 이를 위해 기존의 골격지도를 개선하여 적은 정보로 그 골격을 표현할 수 있는 새로운 골격지도를 제안하였다. 충돌 회피를 위해 제안된 알고리즘은 두 로봇이 만날 수 있는 다양한 경우에 대해 간단하게 최단경로의 쌍을 찾을 수 있음을 보여준다. 두 로봇이 만나는 상황에 따라 회피동작을 미리 설정하는 방식으로 연산량이 작은 회피 또한 가능하지만, 매우 많은 경우의 수를 고려해야 하므로 알고리즘이 매우 복잡해진다. 그러나 이 논문에서 제안하는 알고리즘은 경로 쌍의 충돌 여부와 최소 비용을 계산하는 과정에서 큰 연산을 필요로 하지만 알고리즘이 간단해 동작 오류를 줄일 수 있다. 시뮬레이션을 통해 제안한 알고리즘으로 위의 네 가지 경우의 회피 문제가 해결 됨을 확인하였다. 앞으로 실험을 통한 충돌회피 알고리즘의 검증이 이루어져야 할 것이며, 이를 위해 로봇 간의 통신 문제 및 로봇과 장애물과의 구별인식 문제 등이 연구되어야 할 것이다.

참고문헌

- [1] J. C. Latombe, Robot Motion Planning, Kluwer, 1991.
- [2] J. Barraquand and J. C. Latombe, "Robot motion planning: A distributed representation approach," *Int. J. Robot. Res.*, vol. 10, no. 6, pp. 628-649, December 1991.
- [3] E. Erdmann and T. Lozano-Perez, "On multiple moving objects," *Proc. of the IEEE Int. Conf. Robotics and Automation*, pp. 1419-1424, 1986.
- [4] P. A. O'Donnell and T. Lozano-Perez, "Deadlock-free and

collision-free coordination of two robot manipulators," *Proc. of the IEEE Int. Conf. Robotics and Automation*, pp. 484-489, 1989.

- [5] 김종환, 로봇축구시스템, 대영사, 2000.
- [6] R. Alami, "Multi-robot cooperation through incremental plan-merging," *Proc. of the IEEE Int. Conf. Robotics and Automation*, pp. 2573-2579, 1995.
- [7] K. Azarm and G. Schmidt, "Conflict-free motion of multiple mobile robots based on decentralized motion planning and negotiation," *Proc. of the IEEE Int. Conf. Robotics and Automation*, pp. 3526-3533, 1997.
- [8] H. Takahashi, H. Nishi, and K. Ohnishi, "Autonomous decentralized control for formation of multiple mobile robots considering ability of robot," *IEEE Trans. on Industrial Electronics*, vol. 51, no. 6, pp. 1272-1279, December, 2004.
- [9] A. Yamashita, T. Arai, J. Ota, and H. Asama, "Motion planning of multiple mobile robots for cooperative manipulation and transportation," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 2, pp. 223-237, April, 2003.
- [10] G. Sanchez and J. C. Latombe, "Using a PRM planner to compare centralized and decoupled planning for multi-robot systems," *Proc. of the IEEE Int. Conf. Robotics and Automation*, pp. 2112-2119, 2002.
- [11] S. M. LaValle and S. A. Hutchinson, "Optimal motion planning for multiple robots having independent goals," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 6, pp. 912-925, December, 1998.
- [12] R. Ghrist, J. M. O'Kane, and S. M. LaValle, "Pareto optimal coordination on roadmaps," *Proc. of Workshop on Algorithmic Foundations of Robotics*, pp. 185-200, 2004.
- [13] J. Baber, J. Kolodko, T. Noel, M. Parent, and L. Vlacic, "Cooperative autonomous driving: Intelligent vehicles sharing city roads," *IEEE Robotics and Automation Magazine*, pp. 44-49, March, 2005.
- [14] D. P. Bertsekas, "Dynamic programming and optimal control," *Athena Scientific*, 2001.



양 동 훈

1976년 7월 29일생. 1999년 아주 대학교 전자공학부(공학사). 2001년 아주대학교 대학원 전자공학과(공학석사). 현 아주대학교 대학원 전자공학과 박사과정 재학중. 관심분야는 이동로봇 경로계획 및 주행제어.

홍 석 교

제어·자동화·시스템공학 논문지 제 11 권 제 10 호 참조.