

< 논 문 >

장애물의 기하투영에 의한 일차매개곡선을 이용한 충돌회피 경로계획

남 궁 인*
(1996년 9월 14일 접수)

A Collision-Free Path Planning Using Linear Parametric Curve Based on Geometry Mapping of Obstacles

Ihn Namgung

Key Words : Robot Path Planning(로봇 경로계획), Geometry Mapping(기하투영), Obstacle Avoidance(장애물회피)

Abstract

A new algorithm for planning a collision-free path is developed based on linear parametric curve. In this paper robot is assumed to a point, and two linear parametric curve is used to construct a path connecting start and goal point, in which single intermediate connection point between start and goal point is considered. The intermediate connection point is set in polar coordinate(θ, ρ), and the interference between path and obstacle is mapped into CPS(connection point space), which is defined a CWS GM(circular work space geometry mapping). GM of all obstacles in workspace creates overlapping images of obstacle in CPS(Connection Point Space). The GM for all obstacles produces overlapping images of obstacle in CPS. The empty area of CPS that is not occupied by obstacle images represents collision-free paths in Euclidian Space. A GM based on connection point in elliptic coordinate(θ, δ) is also developed in that the total length of path is depend only on the variable δ . Hence in EWS GM(elliptic work space geometry mapping), increasing δ and finding the value of δ for collision-free path, the shortest path can be searched without carrying out whole GM. The GM of obstacles expresses all possible collision-free path as empty spaces in CPS. If there is no empty space available in CPS, it indicates that path planning is not possible with given number of connection points, i. e. path planning is failed, and it is necessary to increase the number of connection point. A general case collision-free path planning is possible by applying GM to configuration space obstacles. Simulation of GM of obstacles in Euclidian space is carried out to measure performance of algorithm and the resulting obstacle images are reported.

1. 서 론

로봇의 장애물회피 및 경로계획은 로봇제어를 위

*회원, 한국전력기술(주)

한 중요한 요소중의 하나이다. 장애물회피와 경로 계획은 특히 로봇프로그래밍의 많은 부분을 단순화 시켜서 단지 출발점과 종착점의 위치와 자세 등을 지정해주면 로봇이 지나야 할 경로를 구하고, 장애 물과 충돌을 회피한 작업을 가능하게 한다. 이러한

로봇의 장애물회피 및 경로계획은 로봇프로그래밍의 능력을 증대시키고, 작업장내에서 충돌등 사고를 줄여주고, 작업자와 컴퓨터간의 연계를 증진시켜준다. 일반적으로 로봇 경로계획은 주어진 두점사이를 연결하는 장애물과 교차하지 않는 경로를 구하는 문제이다. 경로계획에는 출발점과 종착점을 연결하는 장애물과 간섭이 없는 경로를 유클리드 공간 상에서 직접 구하는 방법이 있고, 로봇의 형상공간을 구하여 형상공간내에서 경로계획을 수행할 수도 있다. 로봇 형상공간 상의 경로계획의 잇점은 로봇의 모양, 크기, 로봇의 형태(이동형 또는 다관절형)에 관계없이 경로계획 알고리즘을 적용할 수 있는데 있다. 로봇의 형상공간은 로봇의 자유도에 관계되고, 형상공간 상의 경로계획은 이동형 로봇 및 고정형 다관절 로봇 등 모든경우에 적용할 수 있다. 그러나 형상공간이 3차원을 넘을 경우, 즉 로봇이 3자유도 이상일 경우, 형상공간의 구축뿐만 아니라 경로계획 자체가 복잡해진다. 그래서 다관절 로봇의 경로계획의 경우 3자유도 이내로 단순화하여 경로계획을 하게 된다.^(4,11,24) 이동형 로봇의 경로계획은 piano mover's problem^(5,19,20)이라고도 불리는데 이것은 이동형 로봇의 경로계획이 피아노를 주변의 가구나 물체에 걸리지 않도록 이동하는 것과 유사하기 때문이다.

로봇 경로계획의 알고리즘의 대표적인 것으로는 visibility graph search, free space search, potential field method 등이 있다. Visibility graph search^(3,10,12,24)는 connectivity graph search로 불리기도 하는데 장애물의 각 꼭지점과 출발점, 종착점사이를 직선으로 연결하여 연결선 중에서 장애물과 교차하지 않는 최단 연결선을 찾는 방법이다. 출발점과 장애물의 모서리 및 종착점을 연결한 연결선을 트리구조의 연결선으로 구성하여, 이 트리구조의 연결선으로부터 최단거리 경로를 찾는다. Visibility graph search의 문제점은 연결선으로 이루어진 트리를 구축하는데 많은 시간이 걸리는 것인데, 이것을 해결하기 위해 국부적으로 트리를 생성하여 재층적으로 경로를 찾는 방법⁽³⁾ 및 형상공간내에서 연결선 트리의 구축 및 경로검색에 heuristic algorithm을 적용하여 경로계획을 하는 방법⁽²⁴⁾이 제안되었다.

Free space search는 작업공간을 4개의 셀로 나누어(2차원 문제의 경우), 각 셀내의 장애물 존재여부로 셀을 빈셀(empty cell), 혼합셀(mixed cell)

및 충전셀(full cell)로 구분하고, 혼합셀은 반복적으로 분할한다. 각 단계마다 빈셀을 트리구조로 구성하여 출발점과 종착점을 연결하는 경로를 찾는다.^(1,10,17) 이것은 셀을 격자형으로 나누어 경로계획을 하기 때문에 grid based 경로계획 또는 quad tree based 경로계획이라 불리기도 한다. 이러한 quad tree based 경로계획의 문제점은 작업공간을 완전한 quad tree로 변환하고 이 트리로부터 경로를 검색하는데 많은 시간이 걸리는 것이다. 이것을 보완한 것이 heuristic algorithm을 적용하여 국부적으로 셀의 분할이 이루어지도록 한 방법⁽¹⁾이 제안되었다.

Potential field 방법^(6~9,14,22)에서는 장애물을 척력(repulsive pole)으로 나타내고, 종착점을 인력(attractive pole)으로 나타낸다. 장애물회피 궤적은 출발점과 종착점을 연결하는 궤적으로 optimization search method를 사용하여 구한다. Potential field 방법의 단점은 국부적인 최저점(local minima)의 존재인데, 장애물의 배치 및 출발점과 종착점의 위치에 따라 경로가 국부최저점에 빠질 경우 이를 극복하기 어렵다는데 있다. 이를 개선하기 위해 superquadric potential field를 이용하는 방법⁽⁸⁾이 제안되었다. 로봇의 관절이 n개인 여자유도 2차원 로봇의 경로계획과 장애물회피에 대해, 로봇관절과 장애물사이에 penalty function 개념을 도입하여 간섭을 방지하고 경로계획하는 것⁽⁹⁾을 제안하였다.

이상의 방법외에 Bug's 알고리즘⁽¹³⁾은 벌레의 움직임을 본따서 출발점으로부터 종착점까지를 직선과 장애물의 외면으로 연결한다. 출발점으로부터 종착점으로 움직여가며, 장애물을 만나면 장애물의 벽을 따라 종착점으로 움직인다. 그래서 다시 종착점으로 연결된 직선을 만나면 그 궤적을 따라 종착점까지 움직여간다. 또한 광학의 원리를 이용하여 최단거리 경로를 구하는 방법⁽¹⁸⁾은 출발점에서 원주방향으로 빛을 투사하여, 장애물근처에서 회절한 빛이 종착점까지 도달하는 경로를 로봇 경로계획에 이용한 방법, 로봇의 경로계획에 spline curve를 사용하여 이동로봇의 제어성능을 향상시키는 방법⁽²⁰⁾도 제안되었다. 그리고 로봇의 경로계획에 유전 알고리즘을 적용하는 방법⁽²³⁾ 및 connectivity graph search와 유사한 방법으로서 작업공간내의 장애물의 voronoi diagram을 이용하여 경로계획을 하는 방법⁽²¹⁾도 제안되었다. 이상에서 살펴본 바와 같이

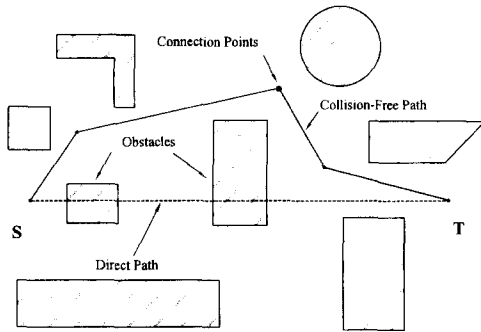


Fig. 1 A series of connected line-segment for a collision-free path

로봇의 경로계획에 있어서, 주어진 알고리즘에 따른 최단거리 경로를 구하거나 가능한 하나의 경로를 구하게 된다.

본 논문에서는 장애물회피 경로는 출발점과 종착점을 연결하는 직선선분의 조합이거나, 직선과 곡선의 조합 또는 곡선의 조합으로 간주하였다(Fig. 1 참조). 즉 장애물회피 경로를 공간곡선으로 간주하여 경로계획을 하였고, 가능한 모든 경로를 주어진 공간곡선의 변수공간에 나타내는 방법을 개발하였다. 매개곡선을 로봇 경로계획에 적용한 것이 Namgung⁽¹⁶⁾이 소개되었고, 이차매개곡선을 이용하여 가능한 충돌회피 경로를 연결점 공간으로 투영하는데 대해⁽¹⁵⁾ 다루었다. 본 논문에서는 계산을 단순화하기 위하여 일차매개곡선을 이용하여 충돌회피 경로를 연결점 공간으로 투영하는데 대하여 다루었다. 그리고 최종경로가 하나의 중간연결점만을 가지고 있다고 가정함으로써, 분제를 단순화시켰다. 여기서 중간연결점의 위치를 극좌표로 (θ, ρ) , 나타내어 경로와 장애물의 간섭여부를 중간연결점 공간으로(CPS, connection point space) 투영하였고, 기하투영(geometry mapping)으로 정의하였다. 이렇게 정의된 원형작업공간(CWS, circular workspace)이 실제작업공간(AWS, actual workspace)을 전부 포함해야 한다. (Fig. 2 참조)

2. 일차매개곡선을 이용한 경로계획

매개변수를 이용한 곡선은 다중 값을 갖는 함수 및 미분 값이 무한으로 발산하는 경우에도 수학적으로 어려움이 표현할 수 있고 방향성을 가지고 있는 여러 가지 이점을 가지고 있다.⁽³⁾ 그래서 Fig. 2의 직선제적 ST를 매개변수, s 를 이용하여 다음과 같

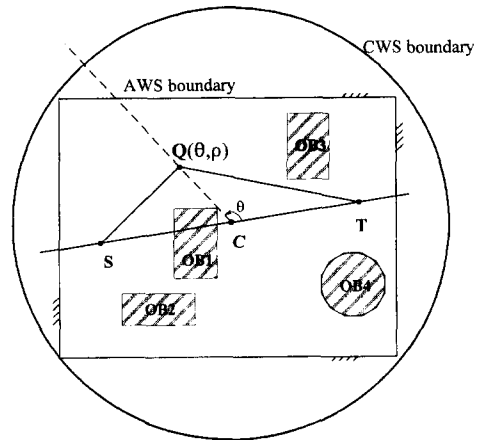


Fig. 2 Polar coordinate representation of an intermediate connection point in circular work space

이 표현한다.

$$r(s) = (T - S)s + S \tag{1}$$

변수 s 를 0에서 1로 변화시켜 가면 $r(s)$ 로 표현되는 경로는 출발점, S , $r(s)_{s=0}$,에서 종착점, T , $r(s)_{s=1}$,으로 움직여 간다. 따라서 매개변수로 표현된 경로는 바로 로봇의 세이에 적용될 수 있다. 또한 매개곡선의 이러한 방향성은 두 직선간의 간섭을 간단히 표현할 수 있게 한다. 여기서, 선분 ST 와 장애물의 모서리 P_j, P_i 간의 교차여부는 다음의 식을 이용하여 구한다.

$$(T - S)s + S = (P_j - P_i)t + P_i \tag{2}$$

위 식 (2)에서 s 는 선분 ST 를 나타내는 매개변수이고, t 는 장애물의 모서리를 나타내는 매개변수이다. 식 (2)로부터 s 와 t 에 대한 식을 얻을 수 있다.

$$s = \frac{(P_{ix} - S_x)(P_{jy} - P_{iy}) - (P_{iy} - S_y)(P_{jx} - P_{ix})}{(T_x - S_x)(P_{jy} - P_{iy}) - (T_y - S_y)(P_{jx} - P_{ix})} \tag{3}$$

$$t = \frac{(T_x - S_x)(S_y - P_{iy}) - (T_y - S_y)(S_x - P_{ix})}{(T_x - S_x)(P_{jy} - P_{iy}) - (T_y - S_y)(P_{jx} - P_{ix})} \tag{4}$$

만일 s 와 t 가 모두 $0 \leq s \leq 1$ 과 $0 \leq t \leq 1$ 의 범위 안에 들 경우, 선분 ST 는 모서리 P_j, P_i 와 교차하게 되고, 그렇지 않은 경우 교차는 일어나지 않는다. 이식에서 분모가 0이 될 경우, s 와 t 는 무한대로 발산하게되는데, 이것은 ST 와 P_j, P_i 가 평행할 경우 발생한다. 따라서 식 (3)과 (4)의 계산은 분자와

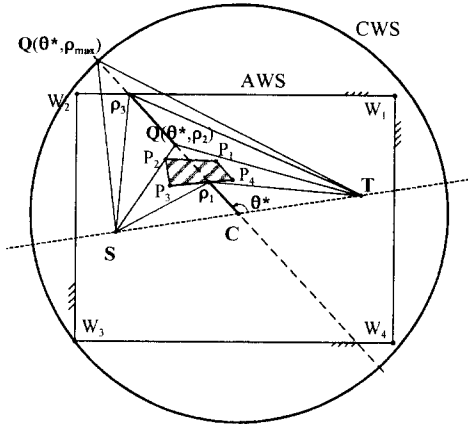


Fig. 3 Interference between path and obstacles

분모를 비교하여, 만일 분자가 클 경우 선분 ST와 P_1P_2 의 교차가 일어나지 않으므로 s 와 t 의 계산은 불필요하다.

3. 원형작업공간의 구성 및 장애물의 기하투영

3.1 원형작업공간의 구성 및 경로와 장애물의 간섭

앞장에서는 출발점과 종착점을 연결하는 직선경로와 장애물과의 교차에 대해 다루었다. 만일 그 경로가 장애물과 간섭을 일으킬 경우 하나의 제어점, Q 을 중간연결점으로 하여 Fig. 3과 같이 경로를 구성한다. 여기서 연결점을 극좌표 (θ, ρ) 로 나타내고, 매개변수의 정의 역은 $0^\circ < \theta < 360^\circ (\theta \neq 0^\circ, 180^\circ, 360^\circ)$ 와 $0 < \rho \leq \rho_{\max}$ 로 정의하여 유클리드공간에서의 실제작업공간 전체를 포함하도록 원형작업공간을 정의한다. θ 가 $0^\circ, 180^\circ, 360^\circ$ 일 경우 경로는 선분 ST와 일치하므로 의미가 없어 제외한다. 이 원형작업공간의 중심, C ,은 선분 ST의 중심점으로 정의한다. 원형작업공간을 구하기 위해서는 실제작업공간을 구성하는 외곽점과 원형작업공간의 중심점으로부터 실제작업공간의 최대 내접반경, R_{\max} ,을 구하여 원형작업공간의 반지름으로 한다. 이렇게 구한 R_{\max} 로부터 θ 값에 따른 Q_{\max} ($Q_{\max,x}, Q_{\max,y}$)을 다음과 같이 정의한다.

$$Q_{\max,x} = \frac{R_{\max}}{D} \{ (T_x - S_x) \cos \theta - (T_y - S_y) \sin \theta \} + C_x$$

$$Q_{\max,y} = \frac{R_{\max}}{D} \{ (T_x - S_x) \sin \theta + (T_y - S_y) \cos \theta \} + C_y$$

(5)

여기서, D 는 선분 ST의 길이, $D = |T - S|$ 이다. 이것을 사용하여 최외곽을 지나는 연결점의 위치를 구할 수 있다.

주어진 각도에 따라 최외곽 연결점의 위치를 식 (5)를 이용하여 구하고, 장애물과 간섭을 일으키는 변수 ρ 의 값을 구한다. 장애물과의 간섭은 선분 SQ, QT와 장애물의 모서리 사이의 교차여부를 판별하여 결정한다. Fig. 3에 나타난 바와 같이 장애물과의 간섭은, (i) 캐적이 장애물의 모서리를 지나는 경우와, (ii) 연결점이 장애물의 모서리와 접촉하는 경우로 구분된다.

첫번째 경우는 식 (6), (7)의 관계식으로 나타낸다.

$$\{ (Q_{\max} - C) \rho_a + C - S \} r + S = P \quad (6)$$

$$\{ (Q_{\max} - C) \rho_b + C - T \} s + T = P \quad (7)$$

위 식에서, 변수 r 와 s 는 각각 선분 SQ와 QT를 나타내고, 변수 ρ_a 와 ρ_b 는 연결점 Q의 위치를 나타내고, 점 P 는 장애물의 모서리점을 나타낸다. 위 식에는 변수 ρ_a 와 ρ_b 는 변수 r 과 s 를 소거함으로써 다음과 같이 유도할 수 있다.

$$\rho_a = \frac{(P_x - S_x)(C_y - S_y) - (P_y - S_y)(C_x - S_x)}{(Q_{\max,x} - C_x)(P_y - S_y) - (Q_{\max,y} - C_y)(P_x - S_x)} \quad (8)$$

$$\rho_b = \frac{(P_x - T_x)(C_y - T_y) - (P_y - T_y)(C_x - T_x)}{(Q_{\max,x} - C_x)(P_y - T_y) - (Q_{\max,y} - C_y)(P_x - T_x)} \quad (9)$$

식 (8)은 선 SQ 상에서 Q의 위치를 나타내는 ρ_a 를 계산하고, 식 (9)는 선 QT 상에서 점 Q의 위치를 나타내는 ρ_b 를 계산한다. 이 ρ_a 와 ρ_b 값은 점 P 가 CQ_{\max} 선상에 있지 않는 한 서로 다르다. 만일 선 SP와 CQ_{\max} 또는 TP와 CQ_{\max} 가 서로 평행하면 식 (8) 또는 식 (9)의 분모는 0이 된다. 이러한 조건은 ρ_a 와 ρ_b 가 $0 \leq \rho \leq \rho_{\max}$ 이므로, 분자와 분모를 바로 비교함으로써 해결할 수 있다. 만일 ρ_a 와 ρ_b 가 모두 $0 \leq \rho \leq \rho_{\max}$ 일 경우, 점 P 는 삼각형 $SQ_{\max}T$ 의 내부에 존재하고, ρ_a 와 ρ_b 중 큰값에 해당하는 선분이 점 P 와 교차하고 이것이 실제 접촉이 일어나는 ρ 값이다. 이것의 예는 Fig. 3의 점 P_2 와 선분 SQ 및 선분 QT 사이의 접촉점 계산에 해당하고, 계산결과 ρ_2 를 구한다.

두번째의 경우, 연결점이 장애물의 모서리와 접촉하는 경우, 선분 CQ_{\max} 는 모서리와 교차하게 된다. 이 경우 ρ 는 식 (2)를 이용하여 다음과 같이

나타낼 수 있다.

$$(Q_{\max} - C)\rho + C = (P_2 - P_1)t + P_1 \quad (10)$$

식 (10)에서 변수 ρ 와 t 는 각각 선분 CQ_{\max} 와 모서리 P_1P_2 를 나타낸다. 식 (10)으로부터 변수 ρ 와 t 는 다음과 같이 식 (11)과 (12)로 정리할 수 있다.

$$\rho = \frac{(P_{1,x} - C_x)(P_{2,y} - P_{1,y}) - (P_{1,y} - S_y)(P_{2,x} - P_{1,x})}{(Q_{\max,x} - C_x)(P_{2,y} - P_{1,y}) - (Q_{\max,y} - C_y)(P_{2,x} - P_{1,x})} \quad (11)$$

$$t = \frac{(Q_{\max,x} - C_x)(C_y - P_{1,y}) - (Q_{\max,y} - C_y)(C_x - P_{1,x})}{(Q_{\max,x} - C_x)(P_{2,y} - P_{1,y}) - (Q_{\max,y} - C_y)(P_{2,x} - P_{1,x})} \quad (12)$$

이 식에서 선분 CQ_{\max} 와 모서리 P_1P_2 가 서로 평행할 때 분모는 0이 된다. 만일 ρ 와 t 가 둘다 $0 < \rho < \rho_{\max}$, $0 < t < 1$ 이면 두 선분간의 교차가 일어난 것이다. 이것의 예는 Fig. 3의 선분 P_3P_4 와 선분 $C_{\max}Q$ 사이의 교차점 계산 및 작업영역 경계를 나타내는 선분 W_1W_2 와 선분 $C_{\max}Q$ 사이의 교차점 계산에 해당하고, 계산결과 ρ_1 과 ρ_3 을 각각 구한다.

이상과 같이 두가지 경우의 계산을 적용하여, 경

로 SQ, QT와 장애물과의 간섭을 판별할 수 있다.

3.2 원형작업공간의 장애물 분류

앞에서 취급한 기하부형은 주어진 각도(θ)에 따른 장애물과의 간섭여부를 변수 ρ 의 값으로 계산하는 것이었다. 본절에는 이러한 기본 작업에 바탕한 장애물의 이미지를 구축하는데 필요한 장애물의 분류에 대해 기술한다. 장애물의 분류는 장애물의 위치에 따라 간섭이 일어나는 ρ 의 영역을 판별하기 위한 것이다. 본논문에서는 장애물을 convex하다고 가정하여 분류하였다. 만일 장애물이 concave인 경우 장애물을 간단히 분류할 수 없다. 그래서 장애물이 concave인 경우는 여러 개의 convex 장애물로 나누어 장애물의 분류를 할 수 있다. 이러한 경우는 concave인 장애물을 convex로 나누는 작업을 별도로 수행해야 한다. 장애물의 위치에 따른 분류는 다음과 같은 방법으로 이루어진다. Case 1의 경우(Fig. 4의 장애물 OB1에 해당) 장애물이 삼각형 SQT의 외부에 있는 경우이고, 이것은 장애물의 모든 모서리점이 삼각형 SQT의 외부에 존재할 경우이고 따라서 간섭은 일어나지 않는다. Case 2의 경우(Fig. 4의 장애물 OB2에 해당)

Table 1 Classification of obstacles according to their location and interference interval of ρ

	Interference condition	Interference interval	Related Fig.
Case 1 (OB1)	Outside of triangle SQT	None	Fig. 4
Case 2 (OB2)	Intersect with line segment ST	$0 \leq \rho \leq \rho_1$	Fig. 4
Case 3 (OB3)	Intersection with line ST and with line segment CQ_{\max}	$\rho_2 \leq \rho \leq \rho_{\max}$	Fig. 4
Case 4 (OB4)	Intersect with outside of line segment ST and with one of line segment SQ_{\max} or TQ_{\max}	$\rho_3 \leq \rho \leq \rho_{\max}$	Fig. 4
Case 5 (OB5)	Contained within triangle $SQ_{\max}T$ (ρ_1 edge contact)	$\rho_1 \leq \rho \leq \rho_2$	Fig. 5
Case 6 (OB6)	Contined within triangle $SQ_{\max}T$ (ρ_1 vertex contact)	$\rho_3 \leq \rho \leq \rho_4$	Fig. 5
Case 7 (OB7)	Located above line ST, intersect with line segment CQ_{\max} , TQ_{\max} or SQ_{\max}	$\rho_2 \leq \rho \leq 1$	Fig. 6
Case 8 (OB8)	located above line ST, intersect with line segment TQ_{\max} or SQ_{\max}	$\rho_2 \leq \rho \leq 1$	Fig. 6
Case 9 (AWS)	Contact with AWS boundary, contact with AWS vertex point	$\rho_{AWS} \leq \rho \leq 1$	Fig. 6

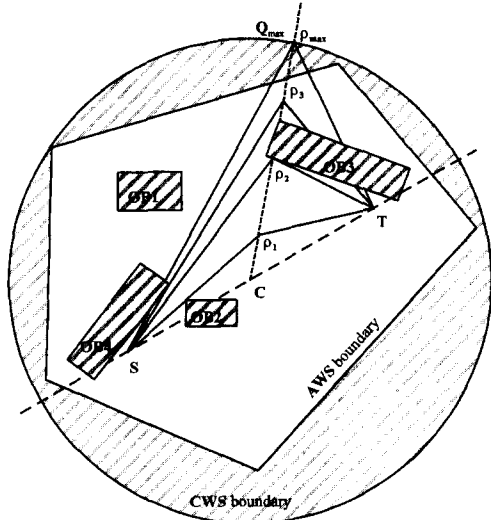


Fig. 4 Classification of obstacles by its location and interference interval of ρ

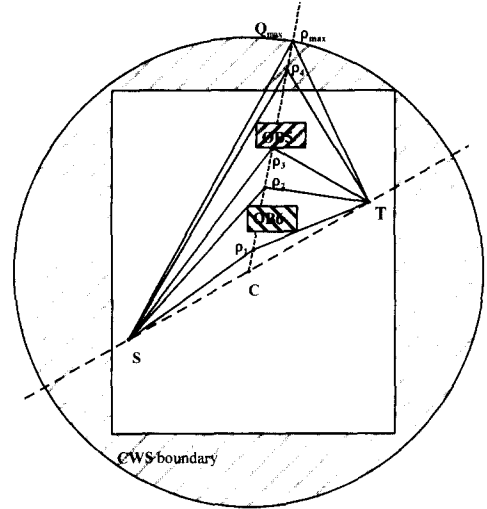


Fig. 5 Classification of obstacles by its location and interference interval of ρ

장애물이 선분 ST와 교차하는 경우이고, 이것은 장애물의 변이 선분 ST와 교차하는 지를 계산하여 결정하고, 간섭구간은 Fig. 4의 $0 \leq \rho \leq \rho_1$ 이다. Case 9는(Fig. 6 참조) 실제작업공간(AWS)에 의한 간섭구간을 계산하기 위한 것으로 일반 장애물과는 별도로 취급한다. 실제 작업공간은 원형작업공간(CWS) 내에 포함되고, 실제작업공간의 외부는 장애물로 처리하여 경로가 실제작업공간 내부에 국한되도록 한다. 따라서 실제작업공간의 경계를 장애물의 변과 같이 취급해야한다. Table 1에 장애물의 위치에 따른 모든 경우의 분류를 하여 조건과 간섭구간을 나타내었다. Figs. 4~6에 장애물의 위치에 따른 분류를 나타내었고 장애물과의 간섭구간을 표시하였다.

장애물의 기하투영은 장애물의 위치에 따른 분류를 수행하여, 간섭구간에 대한 정보를 구한 후, 그 간섭구간의 하한, ρ_{min} ,과 상한, ρ_{max} 에 대한 계산을 한다. 따라서 각각의 장애물에 대해 각도, θ , 를 0° 에서 360° 로 증가시키며, 간섭구간을 나타내는 변수 ρ 를 연결점 공간에 매핑함으로써 장애물의 이미지를 구한다. 이렇게 구한 장애물의 이미지는 서로 겹치는 이미지를 형성한다. 여기서 장애물의 이미지가 투영되지 않은 빈 공간이 나타나는데, 이 공간은 장애물과 간섭이 없는 모든 궤적을 나타낸다. (Figs. 7~12 참조)

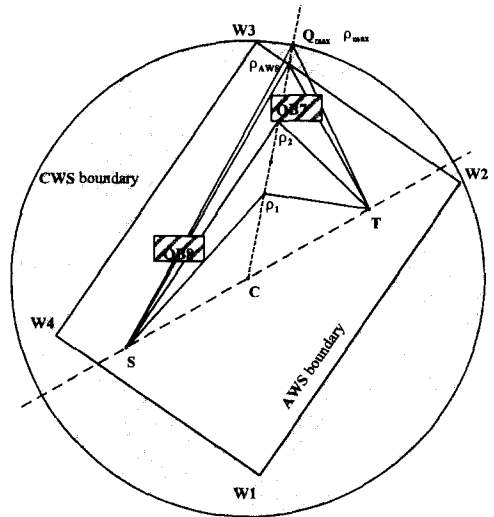


Fig. 6 Classification of obstacles by its location and interference interval of ρ

4. 타원좌표 연결점 공간의 구성과 기하투영

4.1 타원작업공간 구성 및 경로와 장애물의 간섭 앞장에서 극좌표 연결점공간 구성과 그에 따른 기하투영 알고리즘을 전개하였다. 본 절에서는 연결점을 타원좌표로 구성한 기하투영에 관하여 다룬다. 연결점이 타원좌표로 구성된 경우, 출발점에서 종착점에 이르는 총거리는 연결점의 타원상의 위치에 관계없이, 즉 각도에 관계없이 일정하다. Fig.

7에 타원좌표 중간연결점, 타원작업공간(EWS, elliptic work space) 및 실제작업공간에 대해 보인다. 타원작업공간의 중심점 C는 출발점과 종착점을 잇는 선분 ST의 중심점으로 정의한다. 타원연결점은 두 개의 변수 θ (각도) 및 δ (반경 관련변수, Fig. 7 참조)로 정의할 수 있고, 다음의 식 (13), 및 (14)로 표현된다.

$$\frac{x^2}{(d+\delta)^2} + \frac{y^2}{\delta^2+2d\delta} = 1 \quad (13)$$

$$x = R_x(\delta, \theta) = (d+\delta) \cos \theta$$

$$y = R_y(\delta, \theta) = \sqrt{\delta^2+2d\delta} \sin \theta \quad (14)$$

식 (13)과 (14)에서 상수 d 는 점 C와 T(또는 S) 사이의 거리, $d = |S-T|/2$ 이다. 타원작업공간을 정의하는 변수 δ_{max} 는 실제작업공간이 타원작업공간에 포함 될 수 있도록 설정되어야 한다. 실제작업공간을 포함하는 최소의 타원작업공간을 구하기 위해서 식 (13)를 이용한다. 임의의 초기값 δ 를 설정하여 실제작업공간을 규정하는 모서리점의 좌표 $W(W_x, W_y)$ 를 식 (13)에 삽입하여 좌변 값이 가장 큰 모서리점을 찾고, 그 좌표를 이용하여 식 (13)으로부터 변수 δ_{max} 를 구한다.

타원좌표 기하투영은 δ_{max} 을 구한후, δ 를 0에서부터 δ_{max} 까지 증가시키며, 장애물과 간섭을 일으키지않는 타원의 각도구간을 계산하여 장애물과 간섭을 일으키는 각도구간을 구한다. 타원좌표계가 식 (13), (14)와 같이 타원을 중심으로한 국부 좌표계로 표현되어 있기 때문에, 장애물과 경로의 간섭계산에서 장애물, 출발점 및 종착점 등은 타원작업공간에 따른 국부좌표계로 변환하여 계산해야 한다. 타원과 장애물 사이의 간섭에는 두가지 경우가

있다. 하나는 장애물과 타원이 두점에서 교차하는 것이고, 다른 하나는 장애물이 타원과 한점에서 접촉하는 것이다. (Fig. 8 참조)

첫번째 경우는 식 (13)의 (x, y) 에 장애물의 모서리를 매개함수의 형태로 표현하여 삽입하여 계산한다.

$$\frac{\{(P_{2x}-P_{1x})\mu+P_{1x}\}^2}{(d+\delta)^2} + \frac{\{(P_{2y}-P_{1y})\mu+P_{1y}\}^2}{(\delta^2+2d\delta)} = 1 \quad (15)$$

식 (15)에서 미지의 매개변수 δ 와 μ 는 각각 타원과 선분 P_1P_2 를 의미한다. 이것을 μ 의 내림차순으로 정리하면,

$$K\mu^2 + 2L\mu + M = 0 \quad (16)$$

여기서, K, L, M 은

$$\begin{aligned} K &= (P_{2x}-P_{1x})^2(\delta^2+2d\delta) + (P_{2y}-P_{1y})^2(d+\delta)^2 \\ L &= P_{1x}(P_{2x}-P_{1x})(\delta^2+2d\delta) + P_{1y}(P_{2y}-P_{1y})(d+\delta)^2 \\ M &= P_{1x}^2(\delta^2+2d\delta) + P_{1y}^2(d+\delta)^2 - (\delta^2+2d\delta)(d+\delta)^2 \end{aligned} \quad (17)$$

주어진 δ 값에 대해 K, L 및 M 을 계산할 수 있고, 식 (16)으로부터 μ 값을 계산한다. μ 값이 $0 \leq \mu \leq 1$ 일때 실제 장애물과의 간섭이 일어나고, 교차점이나 접촉점, $P(P_x, P_y)$ 은 다음의 식 (18)로 구한다.

$$\begin{aligned} P_x &= (P_{2x}-P_{1x})\mu + P_{1x} \\ P_y &= (P_{2y}-P_{1y})\mu + P_{1y} \end{aligned} \quad (18)$$

이제 P 점의 타원상의 각도는 식 (13)과 (14)로부터 구한다. 단순히 주어진 타원과 선분의 교차여부는 식 (16)의 판별식을 이용하여 교차여부를 확인

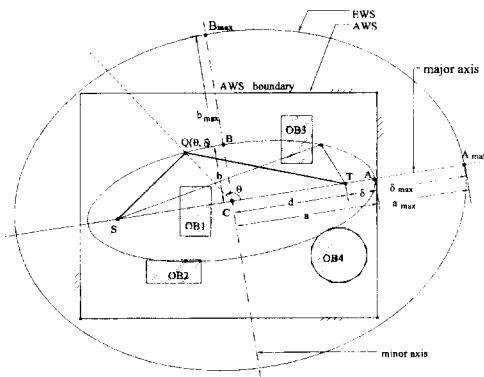


Fig. 7 Elliptic work space and intermediate connection point in elliptic coordinate

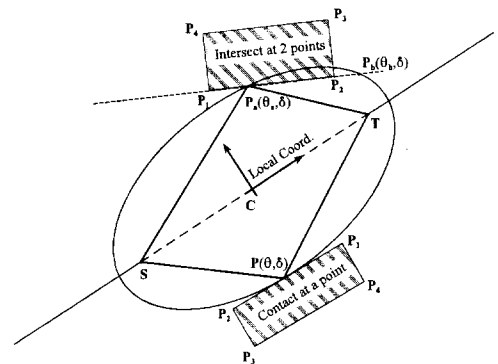


Fig. 8 Contact and intersection between path and obstacles

할수있다.

타원과 장애물과 접촉이 일어난 경우는 식(16)의 판별식이 $L^2=KM$ 인 경우이다. 그래서 이것을 조건으로하여 장애물의 모서리에 접하는 타원의 크기 δ 를 결정할 수 있다.

$$L^2 - KM = \{ (P_{2x} - P_{1x}) P_{1x} (\delta^2 + 2d\delta) + (P_{2y} - P_{1y}) P_{1y} (d + \delta)^2 \}^2 - \{ (P_{2x} - P_{1x})^2 (\delta^2 + 2d\delta) + (P_{2y} - P_{1y})^2 (d + \delta)^2 \} \{ (P_{1x}^2 (\delta^2 + 2d\delta) + P_{1y}^2 (d + \delta)^2) - (\delta^2 + 2d\delta) (d + \delta)^2 \} = 0 \quad (19)$$

식 (19)를 δ 의 내림차순으로 정리하면,

$$P\delta^2 + 2Pd\delta + R = 0 \quad (20)$$

여기서, P 와 R 은 다음과 같다.

$$P = (P_{2x} - P_{1x})^2 + (P_{2y} - P_{1y})^2 \\ R = P_{1x}^2 d^2 - \{ (P_{2x} - P_{1x}) P_{1x} - (P_{2y} - P_{1y}) P_{1y} \}^2 \quad (21)$$

식 (21)과 (20)을 이용하여 구한 δ 값이 $0 \leq \delta \leq \delta_{max}$ 인 경우 실제 접촉이 일어난 경우이고, 접촉점의 위치는 다음의 식 (22)와 식 (17)을 이용하여 구한다.

$$\mu = -\frac{L}{K} \quad (22)$$

그리고 접촉점의 각도는 식 (13) 과 (14)를 이용하여 얻는다.

앞의 첫번째 경우에서와 같이 타원의 크기가 미리 주어진 경우, 타원과 장애물과 직접적인 교차는

일어나지 않지만 중간연결점이 타원상에 있는 관계로 경로와 장애물 사이의 직접간섭이 발생할수 있다(Fig. 9 참조). 이렇게 경로와 장애물이 접촉하는 경우 출발점이나 종착점과 장애물의 모서리점을 연결한 선이 타원과 접촉하고, 그 각도를 장애물과 간섭이 일어난 각도로 간주할 수 있다. Fig. 9의 예는 출발점과 타원상의 연결점을 연결한 경로가 장애물의 모서리와 접촉하는 경우이다. 극부 타원 좌표로 나타낸 출발점의 좌표는 $S(-d, 0)$ 이고, 따라서 출발점과, $S(-d, 0)$, 모서리점을, $P(P_x, P_y)$, 연결하는 매개곡선은 식 (23)으로 표현된다.

$$x = (P_x + d)\mu - d \\ y = P_y\mu \quad (23)$$

식 (23)을 식 (14)에 대입한후 μ 의 내림차순으로 정리하면

$$A\mu^2 - 2B\mu - D = 0 \quad (24)$$

여기서, A, B 와 D 는 다음과 같다.

$$A = (d + \delta)^2 P_y^2 + (\delta^2 + 2d\delta) (P_x + d)^2 \\ B = (\delta^2 + 2d\delta) (P_x + d) d \\ D = (\delta^2 + 2d\delta)^2 \quad (25)$$

δ 는 주어진 값이므로 식 (25) 와 (24)로부터 μ 를 계산할 수 있다. 식 (24)의 판별식은, $B^2 + AD$, 양이므로 항상 두 개의 근을 갖는다. 이것은 Fig. 9에서 보인 바와 같이 출발점 S와 장애물의 모서리를 연결하는 선과 타원은 항상 두 개의 교차점을 있음을 증명한다. 여기서 구하고자 하는 μ 값은 두 개의 값중 큰 것이다. (Fig. 9 참조)

4.2 타원작업공간의 장애물의 위치에 따른 분류

이상에서 유도한 세가지 장애물간섭 계산은 장애물과의 간섭이 일어나는 각도를 구하는 기하투영의 가장 기본이 되는 작업이다. 기하투영을 하기 위해서는 장애물과의 간섭이 일어나는 각도구간을 구해야 하는데, 이것은 장애물의 위치에 따라 분류를 하여 실제 간섭이 일어나는 각도구간을 구한다. 이 작업은 앞절에서 다룬 원형 기하투영시 장애물분류와 같은 개념이다. 장애물의 모든 모서리 점이 타원의 외부에 존재하는 경우에도 타원과 실제교차가 일어날 수 있다(Fig. 10의 OB2 참조). 이런 경우에도 선 ST와 교차하는 양상에 따라 몇가지로 구분할 수 있다. 즉 case 1은 장애물 OB1의 모든 모서리점이 타원의 외부에 존재하고, 장애물의 모서

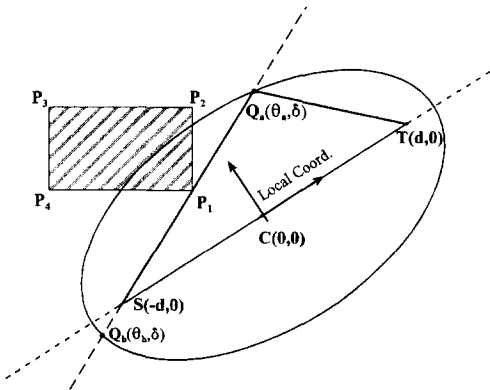


Fig. 9 Direct intersection of path

Table 2 Classification of obstacle according to their location and interference interval of θ

	Interference Condition	Interference Angle (θ°)	Related Fig.
Case 1 (OB1)	All vertices of obstacle located outside of ellipse, Edge of obstacle doesn't intersect obstacle	None	Fig. 10
Case 2 (OB2)	All vertices of obstacle located outside of ellipse, Obstacle intersect line-segment ST	all ($0^\circ \leq \theta \leq 360^\circ$)	Fig. 10
Case 3 (OB3)	All vertices of obstacle located outside of ellipse, Edge of obstacle intersect ellipse, Obstacle intersects part of line ST from S to outside	$0 \leq \theta \leq \theta_{2_OB3}$ $\theta_{1_OB3} \leq \theta \leq 60^\circ$	Fig. 10
Case 4 (OB4)	All vertices of obstacle located outside of ellipse, Edge of obstacle intersect ellipse, obstacle intersects part of line ST from T to outside	$\theta_{1_OB4} \leq \theta \leq \theta_{2_OB4}$	Fig. 10
Case 5 (OB5)	All vertices of obstacle located outside of ellipse, Edge of obstacle intersect ellipse, Obstacle doesn't intersects line ST	$\theta_{1_OB5} \leq \theta \leq \theta_{2_OB5}$	Fig. 10
Case 6 (OB6)	Obstacle intersects ellipse, Obstacle doesn't intersects line ST	$\theta_{1_OB6} \leq \theta \leq \theta_{2_OB6}$	Fig. 11
Case 7 (OB7)	Obstacle intersects ellipse, Obstacle intersects part of line ST from T to outside	$0^\circ \leq \theta \leq \theta_{2_OB7}$ $\theta_{1_OB7} \leq \theta \leq 360^\circ$	Fig. 11
Case 8 (OB8)	Obstacle intersects ellipse, Obstacle intersects part of line ST from S to outside	$\theta_{1_OB8} \leq \theta \leq \theta_{2_OB8}$	Fig. 11
Case 9 (OB9)	Obstacle intersects ellipse upper part and lower part. Obstacle intersects line-segment ST .	All ($0 \leq \theta \leq 360^\circ$)	Fig. 11
Case 10 (OB10)	Obstacle intersects ellipse upper part. Obstacle intersects line-segment ST .	$\theta_{1_OB10} \leq \theta \leq \theta_{2_OB10}$	Fig. 12
Case 11 (OB11)	Obstacle intersects ellipse lower part. Obstacle intersects line-segment ST .	$\theta_{1_OB11} \leq \theta \leq \theta_{2_OB11}$	Fig. 12
Case 12 (OB12)	Obstacle contained inside of ellipse, Obstacle located upper part of ellipse.	$\theta_{1_OB12} \leq \theta \leq \theta_{2_OB12}$ $\theta_{3_OB12} \leq \theta \leq \theta_{4_OB12}$	Fig. 13
Case 13 (OB13)	Obstacle contained inside of ellipse, Obstacle located lower part of ellipse.	$\theta_{1_OB13} \leq \theta \leq \theta_{2_OB13}$ $\theta_{3_OB13} \leq \theta \leq \theta_{4_OB13}$	Fig. 13
Case 14 (OB14)	Obstacle contained inside of ellipse, Obstacle intersects part of line ST from S to outside.	$0^\circ \leq \theta \leq \theta_{1_OB14}$ $\theta_{2_OB14} \leq \theta \leq 360^\circ$	Fig. 13
Case 15 (OB15)	Obstacle contained inside of ellipse, Obstacle intersects part of line ST from S to outside.	$\theta_{1_OB15} \leq \theta \leq \theta_{2_OB15}$	Fig. 13
Case 16 (OB16)	Obstacle contained inside of ellipse, Obstacle intersects line-segment ST .	$\theta_{1_OB16} \leq \theta \leq \theta_{3_OB16}$ $\theta_{2_OB16} \leq \theta \leq \theta_{4_OB16}$	Fig. 13

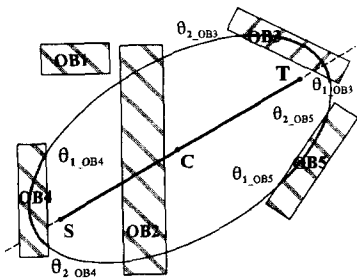


Fig. 10 Classification of obstacle

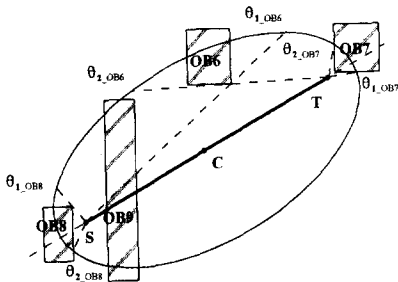


Fig. 11 Classification of obstacle

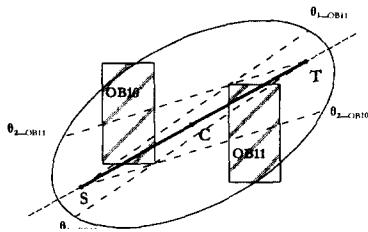


Fig. 12 Classification of obstacles

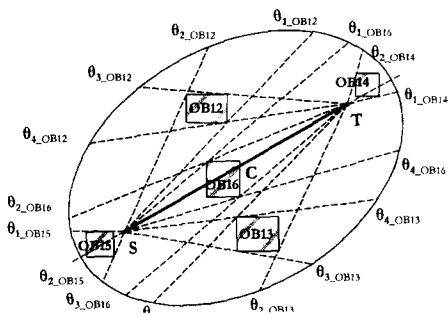


Fig. 13 Classification of obstacles

리가 타원과 교차하지 않고 간섭각도 구간은 없다. Case 2는 장애물 OB2의 모든 모서리점이 타원의 외부에 존재하고, 장애물이 선분 ST와 교차하고 간섭 각도구간은 전체 ($0^\circ \leq \theta \leq 360^\circ$)이다. 타원좌표 기하투영의 경우 장애물의 분류는 위치에 따라 16가지로 분류된다. 실제작업공간(AWS)은 타원작업

공간(EWS)에 포함되고 선분 ST를 포함하기 때문에 δ 의 크기에 따라 cases 1, 3 또는 4로 분류할 수 있다. Table 2에 모든 경우의 장애물의 위치에 따른 분류를 정리하였다. Figs. 10~13에 장애물의 위치에 따른 분류를 나타낸다.

타원좌표 기하투영의 순서는 주어진 δ 에 대하여 Table 2에 따라 장애물을 분류하여 간섭이 일어나는 각도 구간을 결정한 후, 간섭이 일어나는 각도를 계산하여 연결점 공간으로 플랫폼한다. 이렇게 연결점 공간에 투영된 장애물의 이미지는 장애물과 간섭을 일으키는 경로를 나타낸다. 따라서 연결점 공간의 빈공간은 장애물 회피경로를 나타낸다. 즉 유클리드 공간상의 경로계획 문제가 연결점 공간의 빈공간으로부터 점선정의 문제로 변환된 것이다. 그리고 연결점 공간의 빈 공간은 모든 장애물회피 경로를 나타낸다.

5. 기하투영 모의실험

5.1 원형작업공간의 기하투영 모의실험

기하투영의 모의실험에서 장애물과 실제작업공간이 convex라는 가정하에 장애물의 분류를 하였다. 실제 작업공간과 장애물이 concave인 경우 Table 1이나 2와 같이 간단히 분류할 수 없다. 그래서 장애물이 concave인 경우 몇 개의 convex 장애물로 나누어 처리해야 한다. Fig. 14에서 장애물 OB2와 OB3는 임의로 분리된 장애물의 예이다. 장애물의 위치에 따라 기하투영의 이미지는 몇개의 특이한 패턴으로 구분된다. 기하투영의 예로써, Fig. 14의 윗부분은 작업공간을 나타내고, 아랫부분은 장애물의 이미지를 나타낸다. 여기서, 실제 작업공간의, AWS, 외곽모서리는 연결점 공간의 상부를 점유하고, 각 모서리는 곡선으로 대응된다. 그리고 선분 ST와 교차하는 장애물은, OB1, 연결점 공간의 하부 및 각도가 $0^\circ, 180^\circ$ 및 360° 인 부근을 차지한다. OB2와 OB3는 합쳐서 하나의 concave 장애물을 형성하고 각각 말굽형태의 이미지를 형성한다. OB5는 ST의 상부에 위치하고, 말굽형태의 이미지를 90° 근처에 형성한다. OB4의 이미지는 두 부분으로 나뉘어지는데, 실제로는 이미지가 360° 를 주기로 반복되므로 하나이다. OB6는 선 ST를 S점 좌측에서 교차하므로 그 이미지는 180° 근처에 형성된다. 장애물의 숫자와 출발점/종착점의 위치를 달리 하여 모의실험을 수행하여 그 결과를 Table 3 및

Figs. 14~19에 나타내었다. Table 3의 실행 시간은 기하투영 각도간격을 3°로 하여 전체적인 기하투영에 소요된 시간을 의미한다.

모의실험 프로그램은 Boland C++/4.0을 사용하여 작성하였고, DOS/6.2 real mode와 윈도우 95상에서 수행되었다. 윈도우 95에서의 모의실험은 윈도우의 특성상 일관적인 결과를 얻을 수 없었고, DOS에 비해 약 두배 정도의 수행시간이 요구되었다. Pentium/90 MHz와 16 MB RAM을 장착한 PC에서 DOS real mode에서의 모의실험에 대한 결과가 Table 3에 정리되어 있다. 장애물의 숫자가 증가할수록 수행시간이 증가하고, 출발점과 종

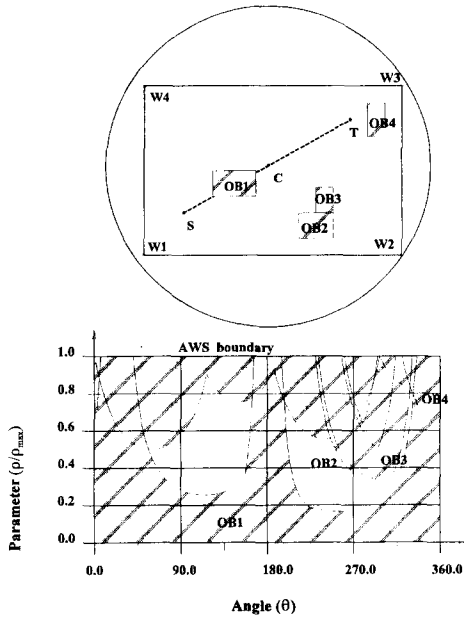
착점의 위치에 따라 수행시간이 달라졌다. 그림의 플롯은 모의실험 결과를 DXF(AutoCad Drawing Interchange File Format)파일 형식으로 출력하여 문서상으로 바로 읽어 들인 것이다. 이를 위해 선긋기, 원그리기, 다각선 그리기, 점그리기, 좌표변환, 문자출력 등의 기본함수를 포함하는 DXF 라이브러리 함수가 개발 사용되었다. 모의실험에 있어서 기하투영시 각도간격을 3°로 하였다.

5.2 타원작업공간의 기하투영 모의실험

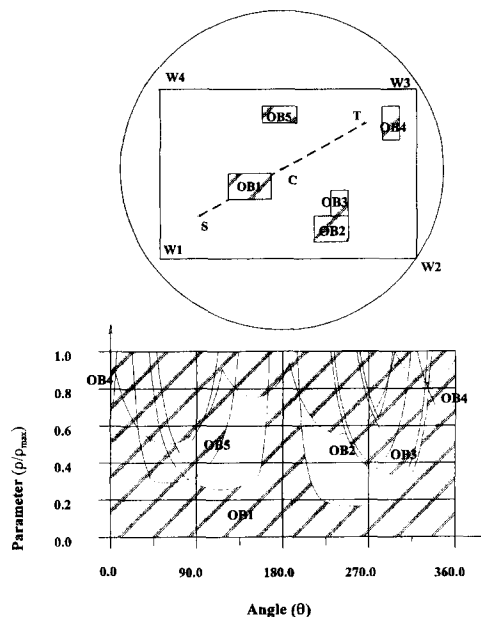
타원작업공간의 기하투영은 δ 를 0에서 δ_{max} 로 증가시켜가며 수행하였다. 원형작업공간의 기하투영

Table 3 Simulation results of circular work space geometry mapping

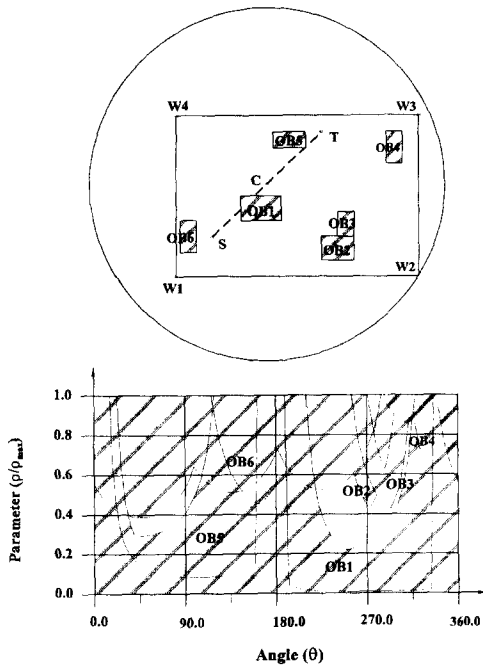
Case	Execution time (sec)	Angular resolution of GM	Execution time for unit angle(x 1/120)	Remark
Fig. 14	0.0736	3°	0.613×10^3	Success
Fig. 15	0.0489	3°	0.408×10^3	Success
Fig. 16	0.0599	3°	0.499×10^3	Success
Fig. 17	0.0703	3°	0.586×10^3	Success
Fig. 18	0.0665	3°	0.554×10^3	Success
Fig. 19	0.1044	3°	0.870×10^3	Fail



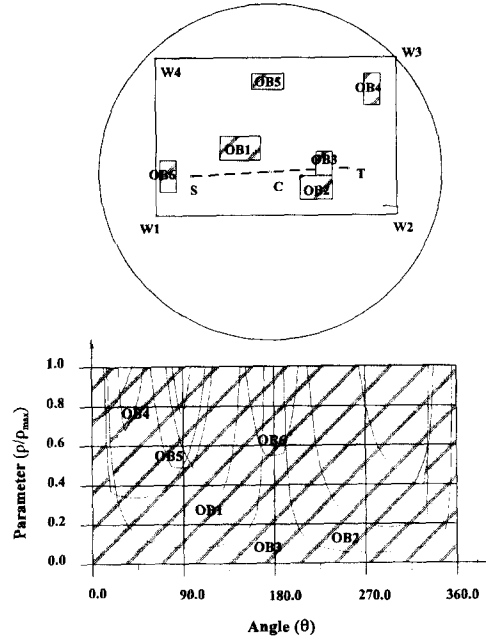
CWS Geometry Mapping of Obstacles
Fig. 14 Simulation of GM-CWS case 1



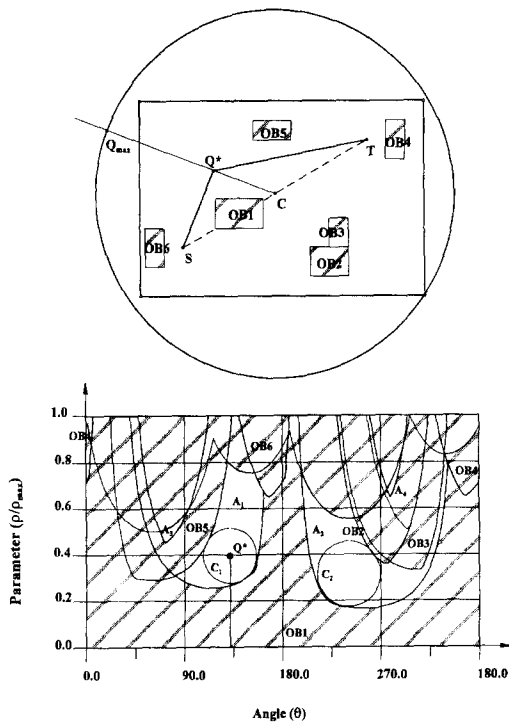
CWS Geometry Mapping of Obstacles
Fig. 15 Simulation of GM-CWS case 2



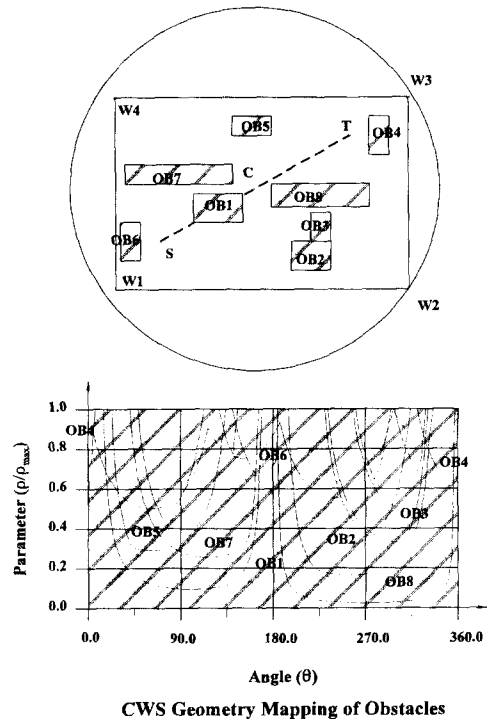
CWS Geometry Mapping of Obstacles
Fig. 16 Simulation of GM-CWS case 3



CWS Geometry Mapping of Obstacles
Fig. 18 Simulation of GM-CWS case 5



CWS Geometry Mapping of Obstacles
Fig. 17 Simulation of GM-CWS case 4



CWS Geometry Mapping of Obstacles
Fig. 19 Simulation of GM-CWS case 6

Table 4 Simulation results of elliptic work space geometry mapping

Case	Execution time (sec)	Resolution of δ/δ_{max}	GM time of unit value of δ/δ_{max} (x1/50)	Remark
Fig. 20	0.1291	0.02	2.5282×10^3	Success
Fig. 21	0.0956	0.02	1.912×10^3	Success
Fig. 22	0.1176	0.02	2.352×10^3	Success
Fig. 23	0.1291	0.02	2.582×10^3	Success
Fig. 24	0.1137	0.02	2.274×10^3	Success
Fig. 25	0.1924	0.02	3.848×10^3	Fail

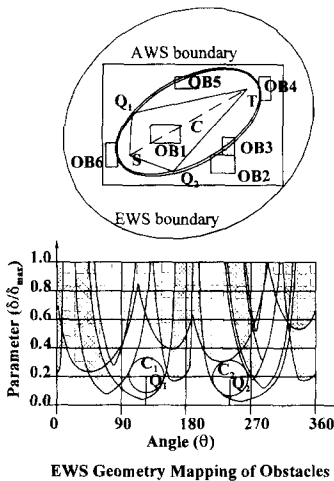


Fig. 20 Simulation of GM-EWS case 1

시 변수 θ 를 증가시키며 기하투영을 하였으나, 타원작업공간의 기하투영에서는 δ 가 독립변수가 되고, θ 가 종속변수가 된다. Table 4에 타원작업공간 기하투영 모의실험에 대한 결과를 보인다. Figs. 20~25의 설정환경은 Figs. 14~19에 보인 원형작업공간 기하투영의 작업환경과 동일하다. Fig. 3과 4를 비교하면 기하투영에 소요되는 실행시간은 타원작업공간 원형작업공간에 비해 약 2배정도 더 걸린다. 그러나 Table 3에 나타낸 원형작업공간의 기하투영은 θ 를 3°하였고, Table 4의 타원작업공간 기하투영은 δ/δ_{max} 의 간격을 0.02로 하여 실행한 것이다. 기하투영의 실행시간은 δ 와 θ 의 간격크기에 따라 달라지므로 직접적인 비교는 어렵다. 그러나 수이진 각도 및 δ/δ_{max} 에 따른 실행시간을 비교하면 타원작업공간 기하투영이 대략 4~5배정도 더 많은 시간이 걸림을 알 수 있다. 그러나 앞에서 설

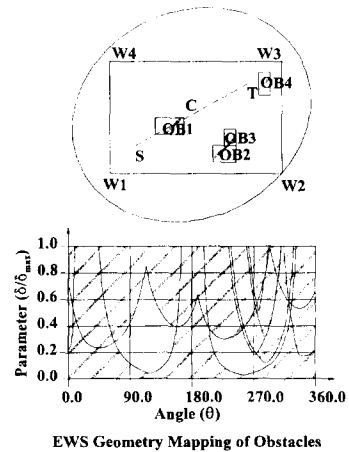


Fig. 21 Simulation of GM-EWS case 2

명한 바와 같이 타원좌표 기하투영은 기하투영을 완전히 수행하는 것보다 장애물회피 경로중에서 가장 경로길이가 짧은 것을 찾는 데 이용될 수 있다. 즉, δ 를 증가시키며 장애물회피 경로를 탐색하여 완전한 기하투영을 하기전에 장애물회피 경로중 가장 경로길이가 짧은 경로를 찾을 수 있는 잇점이 있다.

기하투영 시간은 작업공간의 복잡도 및 출발점/종착점의 위치와 관련되고, 이러한 기하매핑 후 후속처리를 통하여 가장 안전한 궤적, 최단거리 궤적 등 추가적 요건을 만족시키는 궤적을 연결점 공간의 빈 공간으로부터 검색할 수 있다. 그리고 기하매핑과 검색을 동시에 수행함으로써 충돌회피 궤적 검색시간을 단축시킬 수 있다. 또한 각도간격(resolution)을 크게 함으로써, 수행시간을 단축시킬 수도 있다. Fig. 19 및 25는 연결점 공간이 장애물의

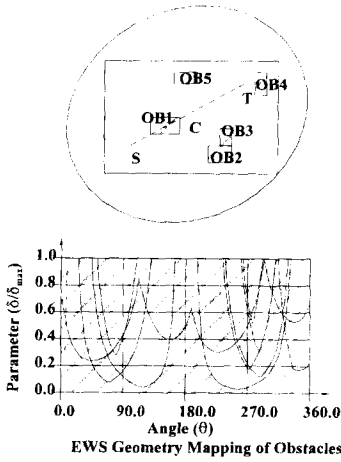


Fig. 22 Simulation of GM-EWS case 3

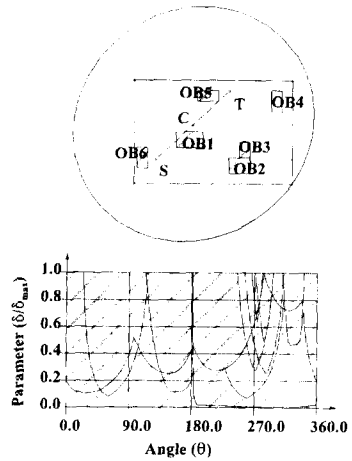


Fig. 23 Simulation of GM-EWS case 4

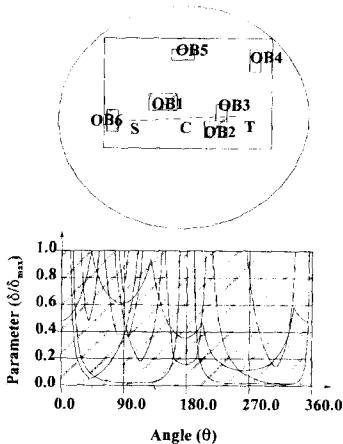


Fig. 24 Simulation of GM-EWS case 5

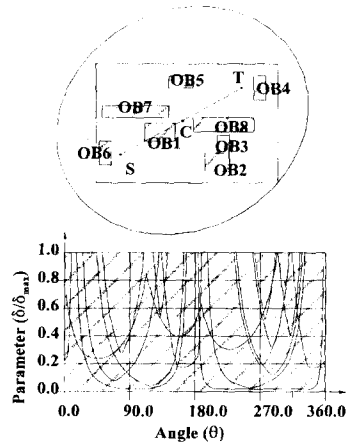


Fig. 25 Simulation of GM-EWS case 6

이미지로 완전히 투영된 경우를 보여주고, 충돌회피 궤적이 없음을 나타낸다.

6. 결 론

주어진 두점 사이를 연결하는 장애물과 간섭하지 않는 경로를 구하는 문제는 상위 레벨 로봇제어를 위한 중요한 문제의 하나이다. 본 논문에서는 유클리드 공간상의 경로계획의 문제를 연결점 공간상의 빈 공간상의 점선성 문제로 변환하는 기하투영 방법을 제시하였다. 본 논문에서는 하나의 연결점이 있는 경우로 국한하여 기하투영의 알고리즘을 개발하였다. 즉, 출발점과 종착점 사이에 하나의 연결점을 도입하여 출발점과 중간연결점 그리고 종착점 사이를 직선으로 연결하여 경로를 구성하였다. 그

리고 경로와 장애물 사이의 간섭여부를 연결점 공간으로 투영하여 장애물의 이미지를 구하였다. 장애물 이미지가 투영되지 않은 빈 공간은 장애물과의 간섭이 일어나지 않는 경로를 나타낸다. 따라서 모든 장애물회피 경로를 기하투영에 의해 연결점 공간에 나타낼 수 있다. 그리고 경로계획의 성공 및 실패는 연결점 공간의 빈 공간 존재여부로 간단히 판별할 수 있다.

연결점 공간은 중간연결점을 구성하는 방식에 따라 달라지는데, 중간연결점을 극좌표(θ, ρ) 및 타원좌표(θ, δ)로 구성한 경우의 기하투영을 보였다. 타원좌표로 구성한 경우 경로의 길이는 각도에 관계없이 δ 에 따라 변한다. 따라서 최단거리 경로를 찾기 위해서는 δ 를 작은값부터 증가시키며 경로를 구한다. 연결점 공간의 차원은 중간연결점의 차

원과 개수에 비례하여 증가한다. 본 논문에서 다룬 것은 2차원의 경우이고 하나의 연결점만 고려 하였으므로, 연결점 공간은 2차원이 된다. 그러나 경로 계획이 실패한 경우 연결점의 숫자를 증가 시켜야 하고, 그에 따라 연결점 공간의 차원이 증가한다.

본 논문에서는 이동로봇(mobile robot)을 점으로 가정하여 2차원 경로계획법을 제시하고 모의실험 하였는데, 일반적인 이동로봇 및 로봇매니퓰레이터의 경우 기하투영을 형상공간(configuration space)에 적용하여 장애물회피 경로계획을 할 수 있다. 이 경우 형상공간을 구한후, 형상공간의 장애물에 대해 기하투영을 수행하여 장애물회피 경로를 구해야 한다.

기하투영 알고리즘을 모의실험을 통해 구현하고 입증하였다. 실제 기하투영에 걸리는 시간은 장애물의 모양이 복잡하고 숫자가 많을수록, 기하투영의 각도간격(극좌표 기하투영인 경우) 또는 반경간격(타원좌표 기하투영인 경우)이 작을수록 더 많은 계산시간을 요한다. 그리고 출발점이나 종착점의 위치에 따라 장애물이 분류가 달라지므로 계산시간이 변한다. 그러나 장애물과 작업공간의 크기는 기하투영 수행시간과는 관계없다.

본 논문의 후속과제로는 3차원으로 기하투영을 확장하는 것, 연결점의 숫자를 증가시키는 것, 경로의 기반곡선으로 고차곡선을 이용하는 것, 기하투영과 결부하여 최단시간 또는 최단거리 경로선택과 같은 추가적인 과제가 남아 있다.

참고문헌

- (1) Barbehenn, M. and Hutchinson, S., 1995, "Efficient Search and Hierarchical Motion Planning by Dynamically Maintaining Single-Source Shortest Paths Trees," *IEEE, Transactions on Robotics and Automation*, Vol. 11, No. 2, pp. 198~214.
- (2) Faux, I. D. and Pratt, M. J., 1979, *Computational Geometry for Design and Manufacture*, John Wiley & Sons.
- (3) Fu, L. C. and Liu, D. Y., 1990, "An Efficient Algorithm for Finding a Collision-Free Path Among Polyhedral Obstacles," *Journal of Robotic Systems*, Vol. 7(1), pp. 129~137.
- (4) Gini, G., Massa, R. and Negretti, R., 1995, "Toward Efficient Path Planning for Articulated Robots," *Journal of Robotic Systems*, Vol. 12(2), pp. 93~104.
- (5) Hopcroft, J. E., Schwartz, J. T. and Sharir, M., 1984, "On the Complexity of Motion Planning for Multiple Independent Object; PSPACE-hardness of the 'Warehouseman's Problem'," *Int. J. Robotics Research*, Vol. 3, No. 4, pp. 76~88.
- (6) Hwang, Y. K. and Ahuja, N., 1992, "A Potential Field Approach to Path Planning," *IEEE, Transactions on Robotics and Automation*, Vol. 8, No. 1, pp. 23~32.
- (7) Khatib, O., 1986, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *Int. J. of Robotic Research*, Vol. 5, No. 1, pp. 90~98.
- (8) Khosla, P. and Volpe, R., 1988, "Superquadric Artificial Potential for Obstacle Avoidance and Approach," *IEEE Proc. Int. Conf. on Robotics and Automation*, Philadelphia, pp. 1778~1784.
- (9) Li, J. Z. and Trabia, M. B., 1996, "Adaptive Path Planning and Obstacle Avoidance for a Robot with a Large Degree of Redundancy," *Journal of Robotic Systems*, Vol. 13(3), pp. 163~176.
- (10) Liu, Y. H. and Arimoto, S., 1992, "Path Planning Using a Tangent Graph for Mobile Robots Among Polygonal and Curved Obstacles," *Int. J. Robotics Research*, Vol. 11, No. 4, pp. 376~382.
- (11) Lozano-Pérez, T., 1983, "Spatial Planning: A Configuration Space Approach," *IEEE Trans. Computers*, Vol. C-32, No. 2, pp. 108~120.
- (12) Lozano-Pérez, T. and Wesley, M. A., 1979, "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles," *Comm. ACM*, Vol. 22, No. 10, pp. 560~570.
- (13) Lumelsky, V. J. and Stepanov, A. A., 1986, "Dynamic Path Planning for a Mobile Automation with Limited Information on the Environment," *IEEE, Trans. Automatic Control*, Vol. AC-31, No. 11, pp. 1058~1063.
- (14) Mayorga, R. V., Janabi-Sharifi, F. and Wong, A. K. C., 1995, "A Fast Approach for the Robust Trajectory Planning of Redundant Robot Manipulator," *Journal of Robotic Systems*, V. ol

- 12(2), pp. 147~161.
- (15) Namgung, I., 1996, "A Global Collision-Free Path Plannig Using Parametric Parabola Through Geometry Mapping of Obstacles in Robot Work Space," *KSME Journal*, Vol. 10, No. 4, PP. 443~449.
- (16) Namgung, I., 1989, "Planning Collision-Free Paths with Applications to Robot Manipulators," Ph. D. Dissertation, University of Florida.
- (17) Noborio, H., Naniwa, T. and Arimoto, S., 1990, "A QuadtreeBased PathPlanning Algorithm for a Mobile Robot", *Journal of Robotic Systems*, Vol. 7(4), pp. 555~574.
- (18) Rosa, F., Laufier, C. and Nájera, J., 1996, "Robust Path Planning in the Plane," *IEEE, Transactions on Robotics and Automation*, Vol. 12, pp. 347~352.
- (19) Schwartz, T. J. and Sharir, M., 1983, "On the Piano Movers' Problem: I. The Case of a Two-Dimensional Rigid Polygonal Body Moving Amidst Polygonal Barriers," *Comm. Pure and Applied Mathematics*, Vol. 36, pp. 345~398.
- (20) Schwartz, J. T. and Sharir, M., 1983, "On the Piano Movers' Problem : II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds," *Adv. Applied Mathematics*, Vol. 4, pp. 298~351.
- (21) Takahashi, O. and Schilling, R. J., 1989, "Motion Planning in a Plane Using Generalized Voronoi Diagrams," *IEEE, Transactions on Robotics and Automation*, Vol. 11, No. 5, pp. 143~150.
- (22) Xiao, J. and Zhang, L., 1996, "Toward Obtaining All Possible Contacts Growing a Polyhedron by its Location Uncertainty," *IEEE, Transactions on Robotics and Automation*, Vol. 12, No. 4, pp. 553~565.
- (23) Zelinsky, A., 1994, "Using Path Transforms to Guide the Search for Findpath in 2D," *Int. Journal of Robotics Research*, Vol. 13, No. 4, pp. 315~325.
- (24) Zhu, D. and Latombe, J. C., 1991, "New Heuristic Algorithms for Efficient Hierarchical Path Planning," *IEEE, Transactions on Robotics and Automation*, Vol. 7, No. 1, pp. 9~20.