

거리 함수를 이용한 로봇의 시변 장애물 회피 동작계획

Robot Motion Planning for Time-Varying Obstacle Avoidance Using Distance Function

전 홍주 고 낙용 남 윤석 이 범희 고 명삼

서울 대학교 대학원 제어계측공학과, 자동화 시스템 공동 연구소

Jeon, Heung-Joo Ko, Nak-Yong Nam, Yun-Seok Lee, Bum-Hee Ko, Myoung-Sam

Automation and Systems Research Institute and

Dept. of Control & Instrumentation Eng. Seoul National Univ.

ABSTRACT

A robot motion planning algorithm for time-varying obstacle avoidance is proposed. The robot motion planning problem is replaced with the optimization problem by using the distance function with the divided configuration space. To divide the configuration space, the polar coordinate system is used. For each divided configuration space, the admissible region where the robot can reach without collisions is obtained using the distance function. For an object moving in a plane, the admissible region is described by linear constraints on the polar coordinate system. A numerical algorithm that solves the optimization problem is shown and the computer simulation is carried out.

1. 서론

로봇의 작업 공간 (Work Space) 내에는 동작 대상인 물체 이외에 장애물이 존재한다. 작업중인 로봇과 다른 물체와의 충돌은 안전 사고나 로봇의 피손등을 야기하므로 로봇은 장애물과의 충돌을 피하며 작업을 진행해야 한다. 특히 좁은 공간내에 많은 로봇이 설치되어 있는 경우나 두 대 이상의 로봇이 공동 작업을 할 경우 충돌 회피를 위한 알고리즘의 필요성은 더욱 커질 것이다.

로봇의 동작 계획(Motion Planning)은 경로 계획(Path planning)과 궤적 계획(Trajectory Planning)으로 구성된다. 회피하고자 하는 장애물이 시불변 장애물(Time-Invariant Obstacle)인가 시변 장애물(Time-Varying Obstacle)인가에 따라 접근 방법에 약간의 차이가 있다. 시불변 장애물만을 회피하고자 하는 경우는 적절한 경로를 찾으면 되나 시변 장애물을 고려하는 경우에는 경로이외에 적절한 궤적도 구해야 한다. 왜냐하면 시변 장애물이 있는 환경에서는 같은 경로상의 속도가 다른 두 동작이 상이한 결과를 가져올 수 있기 때문이다. 결국 시불변 장애물

에 대한 회피 동작은 로봇과 장애물의 기하학적인 정보로부터 가능하지만 시변 장애물이 있는 경우에는 기하학적인 정보뿐만 아니라 시변 장애물과 로봇의 운동에 따른 변화를 고려해야 한다. 이것이 시변 장애물을 회피하는 동작을 계획할 때의 어려운 점이다.

Lozano-Perez[8]는 물체의 형상 인자(configuration parameter)를 좌표축으로 하는 형상 공간(configuration space)에 장애물에 대한 기하학적 제한을 표현하는 방법을 제안하였다. Gilbert[4]는 거리 함수 (Distance Function)를 정의하고 이를 이용하여 충돌 회피 경로 계획의 문제를 최적화 문제로 구성하였다. Lee[7]는 물체의 속도를 고려하여 작성된 충돌 지도(Collision Map)를 기반으로 한 시변 환경내에서의 충돌 회피에 관한 알고리즘을 제안하였다. Shin[10]은 형상 공간을 확장한 형상 공간 - 시간 지도(Configuration Space-Time Map)를 구성하여 여러대의 로봇이 주어진 우선 순위에 따라 시작점에서 목적점까지 일정한 속도(Piecewise Constant Velocity)로 동작하는 알고리즘을 제안하였다.

본 연구의 대상은 여러개의 시불변/시변 장애물이 있는 환경에서 로봇의 충돌 회피를 위한 동작 계획으로 한다. 로봇의 동작 계획 문제를 최적화 문제로 구성하고 구성된 최적화 문제를 푸는 알고리즘을 제안하며 시뮬레이션을 통하여 이를 검증한다.

2. 거리 함수 (Distance Function)의 정의와 성질[4]

거리 함수는 잠재적인 충돌 가능성을 지닌 물체사이의 충돌회피를 표시하기 위하여 제안되었으며 이를 이용하여 충돌 회피 경로 계획의 문제를 상태 변수와 입력의 제한 조건을 지닌 최적 제어 문제로 구성한 연구가 있었다[4]. 거리 함수를 사용하여 로봇과 장애물간의 충돌 여부를 파악하는 방법의 장점은 기하학적인 부분은 물체사이의 거리를 구하는 알고리즘 내부에만 존재하므로 n 차원의 공간으로 확장이 용이하며 시변 장애물 회피로의 적용이 쉽다

는 점이다. 또한 거리 함수를 계산하는 효율적인 알고리즘[3]이 개발되어 있다.

로봇이나 작업 공간내의 장애물은 그것이 차지하는 공간상의 부피로 생각할 수 있으며 R^n 상의 집합으로 표현된다. 로봇이나 시변 장애물의 경우 그것이 차지하는 공간상의 부피는 로봇의 운동과 시간에 따라 변화하므로 다음과 같이 로봇의 일반화된 좌표계 (Generalized Coordinate) 와 시변 장애물의 위치/자세를 결정하는 좌표계의 함수가 된다.

$$K = K(q) \quad (1)$$

식 (1) 을 이용하면 R^n 상의 두 집합 K_i 와 K_j 간에 충돌이 일어나지 않을 조건은 다음과 같다.

$$K(x(t)) \cap K_j(x(t)) = \emptyset \quad (2)$$

R^n 상의 콤팩트 (compact) 한 두 집합 K_i, K_j 간의 거리는 식 (3) 와 같이 정의할 수 있으며, 이를 이용하면 충돌이 일어나지 않을 조건 (2) 는 식 (4) 로 변환된다.

$$d(K_i, K_j) = \min \{ \|z_i - z_j\| : z_i \in K_i, z_j \in K_j \} \quad (3)$$

$$d(K_i, K_i) > 0 \quad (4)$$

가정1 : 로봇트및 장애물은 볼록 집합(convex set) 또는 볼록 집합의 합집합 (union) 으로 표시할 수 있다. 또한 매 순간 로봇트및 장애물이 차지하는 공간상의 집합은 R^n 상의 콤팩트 집합(compact set) 이다.

T_1 에서 T_2 까지의 시간동안 로봇트가 N_0 개의 시변/시불변 장애물과 충돌이 일어나지 않는 조건은 식 (4) 로부터 다음과 같이 표현할 수 있다.

$$d(K(q), K_i(q')) > 0 \quad (5)$$

이때,

$K(q(t))$: 로봇트가 차지하는 공간상의 부피

$K_i(q'(t))$: 장애물이 차지하는 공간상의 부피

$i \in \{1, 2, \dots, N_0\}$

$t \in [T_1, T_2]$

로봇트가 몇개의 볼록 집합의 합집합으로 표현이 되거나 장애물이 볼록 집합의 합집합으로 표시가 되는 경우에 거리 함수는 식 (6) 와 같다.

$$d(K_i, K_j) = \min \{ d(K_{i_n}, K_j) \} \quad (6)$$

이때,

$$K_i = \cup K_{i_n} \quad n = 1, 2, \dots, M$$

식 (3) 으로 정의된 거리 함수의 성질은 K_i 와 K_j 의

성질에 의하여 결정된다. 가정1이 만족이 되면 거리 함수 $d(K_i, K_j)$ 는 K_i, K_j 의 위치/자세를 결정하는 좌표계에 대하여 Lipschitz 연속(Continuous)이다.

3. 형상 공간의 분할과 극 좌표계의 도입

형상 공간 (Configuration Space) 의 각 좌표축은 로봇의 위치 (Position) 와 자세 (Orientation) 를 결정하는 자유도 (Degrees of Freedom) 를 나타낸다. 형상공간 내에서 로봇의 동작을 계획하면 역기구학 (Inverse Kinematics) 이 필요 없으며 그에 의한 다중 기구학해의 발생이 없다는 장점이 있다. 또한 형상 공간내에서 로봇의 운동은 한점이 움직이는 것이므로 다루기 편한 장점도 가지고 있다.

형상 공간내의 각 점은 로봇의 위치와 자세를 나타내므로 로봇이 초기 상태에서 목적 상태로 움직이는 것은 결국 형상 공간내의 한 점에서 다른 점, 즉 초기 상태를 나타내는 점에서 목적 상태를 나타내는 점으로 이동하는 것이다. 로봇이 초기 상태(Initial State)에서 목적 상태(Final State)로 운동하는 것은 매순간 로봇의 현재 상태를 나타내는 형상 공간상의 점과 목적 상태를 나타내는 점의 거리 (Euclidian Norm) 가 감소하는 것으로 생각할 수 있다. 이러한 개념을 사용하기 위하여 형상 공간이 2차원인 경우에, 로봇의 목적 상태를 나타내는 형상 공간의 한 점을 원점으로 하는 극좌표계(Polar Coordinate)를 도입한다 (3차원의 경우에는 로봇의 목적 상태를 원점으로 하는 구좌표계(Spherical Coordinate)가 된다). 이와 같이 극좌표계를 도입하면 형상 공간내의 각 점은 원점에서의 거리 r 과 편각 θ 로 표현된다(3차원의 경우에는 r 과 ϕ, θ 로 표현된다).

$t = t_k = k \cdot \Delta t$ 일때 로봇트는 목적 상태로부터 r_k 떨어진 곳에 위치한다고 가정한다. 목적 위치로부터 r_k 떨어진 곳은 2차원 상에서 원궤적을 이루므로 극좌표계로 표현된 형상 공간을 원점을 중심으로 하고 반지름이 r_k 인 N 개의 원으로 분할한다.

$$r_k = (N+1-k)/(N+1) \cdot D \quad k = 1, 2, \dots, N \quad (7)$$

이때,

D 는 초기 상태와 목적 상태 사이의 거리.

공간 분할에 사용된 각각의 원은 극 좌표계상에서 간단하게 표현된다. 극 좌표계는 원점에서의 거리, 즉 원의 반지름과 편각으로 평면상의 각 점을 표현하는 것이므로 각 원의 원주상의 점은 원점으로부터의 거리가 같고 편각이 일정한 범위내에 드는 것이다.

먼저 각 원의 원주를 따라 관절 제한 조건을 만족하는 편각 θ_k 의 범위를 구한다. 즉,

$$J_{C1, \min} \leq r_k \cdot \cos \theta_i \leq J_{C1, \max} \quad (8)$$

$$JC_{2, \min} \leq r_k \cdot \sin \theta_i \leq JC_{2, \max} \quad (9)$$

이때,

$$r_k = (N+1-k)/(N+1) \cdot D, \quad k = 1, 2, \dots, N$$

$J_{C1, \min}, J_{C1, \max}$: 첫번째 형상인자의 하한치와 상한치

$J_{C2, \min}, J_{C2, \max}$: 두번째 형상인자의 하한치와 상한치

을 만족하는 $\theta_k, k = 1, 2, \dots, N$ 의 범위를 구한다. 이때 윗 식을 만족하는 $\theta_k, k = 1, 2, \dots, N$ 의 범위와 같이 주어진다.

$$a_k \leq \theta_k \leq b_k \quad (10)$$

형상 공간의 분할에 사용된 원은 시간을 포함하고 있는 것이므로 시변 장애물에 대해서도 시불변 장애물과 같이 생각할 수 있는 이점이 있다. 즉 시변 장애물의 움직임을 일정한 시간마다 잘라서 (Sampling) 그 시간 간격동안 시변 장애물이 쓸고 지나간 체적 (Swept Volume) 을 그 시간대에서의 회피해야 할 장애물로 구성하는 것이다. 관결 제한 조건을 만족하는 θ_k 의 범위에서 다시 장애물과의 충돌을 일으키지 않는 θ_k 의 범위를 구한다. 이는 시불변 장애물과 일정한 시간대의 쓸고 지나 체적으로 구성된 시변 장애물과의 거리 함수가 양수인 부분을 찾는 것이다. 거리 함수는 가정1 을 만족하면 연속이므로 장애물과의 충돌을 일으키지 않는 θ_k 의 범위는 컴팩트한 집합으로 표시할 수 있다.

$$d(M(\theta_k), O_i) > 0 \quad i = 1, 2, \dots, N_o \quad (11)$$

이때,

$$r_k = (N-k)/N \cdot D, \quad k = 1, 2, \dots, N-1$$

$$a_k \leq \theta_k \leq b_k$$

$M(\theta_k)$: $\theta = \theta_k$ 일때 로봇트가 차지하는 공간상의 부피

O_i : 시불변 장애물의 공간상의 부피, 또는 시변 장애물이 $[i \cdot \Delta t, (i+1) \cdot \Delta t]$ 의 시간동안 쓸고 지나간 체적으로 구성된 공간상의 부피

각각의 원에 대하여 관결 제한 조건과 충돌 회피 조건을 만족하는 부분을 구하면 2차원의 경우 다음과 같이 θ_k 에 대한 선형 제한 조건이 된다.

$$a_{ik} \leq \theta_k \leq b_{ik} \quad i \in I \quad (12)$$

위와 같은 과정을 통해 로봇트의 충돌 회피 동작 계획 문제는 다음과 같이 선형 제한 조건을 갖는 최적화 문제로 치환되어 수치 해석적인 방법으로 풀 수 있다.

$$\text{Minimize } J(\theta_1, \theta_2, \theta_3, \dots, \theta_N) \quad (13)$$

$$a_{ik} \leq \theta_i \leq b_{ik} \quad i \in I_k \quad (14)$$

비선형 성능 지수 (Nonlinear Performance Index) 와 선형 제한 조건 (Linear Constraints) 으로 구성되는 최적화 문제의 해법으로 널리 사용되는 것이 그라디언트 프로젝션 (Gradient Projection) 이다. 이 방법의 근본 원리는 성능 지수 함수의 음의 그라디언트를 제한 조건으로 구성되는 공간에 프로젝션하는 것이다.

(13), (14) 의 최적화 문제에서 얻어진 해는 일정한 간격의 시간에서의 위치를 나타내므로 근사식을 사용하여 간격 사이의 시간에서의 위치와 가속도, 속도를 구한다. 3차 다항식 스플라인 (Cubic Polynomial Spline) 을 이용한 근사식을 사용하면 2계 도함수 (Second Derivatives) 까지 연속인 부드러운 곡선을 얻을 수 있다.

3차원의 경우에는 J 는 ϕ_k 와 θ_k 의 함수가 되며 반지름이 r_k 인 구상의 제한 조건을 만족하는 영역을 탐색한 후 영역을 평면상으로 프로젝션 (Projection) 한후 그 내부에 포함되는 최대의 직사각형을 구하면 ϕ_i 와 θ_i 의 제한 영역을 선형으로 표시할 수 있다.

4. 동작 계획의 예

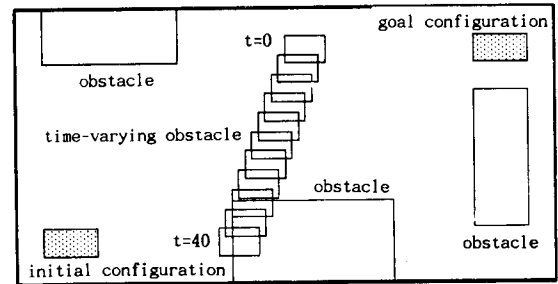


그림1. 충돌 회피 동작계획의 대상

Fig 1. A Sample Work Space

그림1. 은 로봇트의 작업 공간과 장애물을 나타낸다. 빗금친 사각형으로 표현된 것이 사각형으로 모델링된 로봇트이며 빗금친 없는 사각형은 장애물을 나타낸다. 로봇트 모델의 좌측 하단 꼭지점의 x 좌표, y 좌표를 형상 인자로 하며 로봇트의 회전 운동은 없다. x 축 형상 인자는 0 과 20 사이에 있어야 하며 y 축 형상 인자는 0 과 10 사이에 있어야 한다. 로봇트의 초기 상태는 x 축 형상 인자는 1.0, y 축 형상 인자는 1.0 이며 목적 상태는 각각 17.0, 8.0 이다. 사각형 모델의 가로, 세로는 각각 2.0, 1.0 이다. 로봇트가 목적 상태에 도달하는 시간은 40 초이다. 시불변 장애물은 세가로 꼭지점이 (1.0, 8.0) 에 위치하며 가로, 세로가 각각 5.0, 2.0 인 것, 꼭지점이 (17.0, 2.0) 에 위치하며 가로, 세로가 각각 2.0, 5.0 인 것, 꼭지점이 (8.0, 0.0) 에 위치하며 가로, 세로가 각각 6.0, 3.0 인 것

으로 구성된다. 시변 장애물은 꼭지점이 $t = 0$ 일때 $(10.0, 8.0)$ 에서 $t = 40$ 일때 $(7.5, 0, 1.0)$ 으로 동속으로 이동하며 가로, 세로의 길이는 각각 1.5, 1.0 이다.

<동작 계획 알고리즘>

step1 : 시간 간격 (Sampling Time) t_{smp} 를 정하고 형상 공간을 구성한다. 즉 사각형으로 모델링된 로봇의 어는 한 꼭지점의 x 좌표, y 좌표를 형상 인자로 한다.

step2 : 형상 공간내의 목적 상태를 나타내는 점을 원점으로 한 극 좌표계를 설정하고 반지름 r_k 인 원 ($k = 1, 2, \dots, N$) 으로 형상 공간을 분할한다.

$$\text{여기서 } N = T / t_{smp}$$

step3 : 동심원으로 표현된 부분 공간에서 원주를 따라 관절 제한 조건과 충돌 회피 조건을 만족하는 영역을 탐색한다. 시변 장애물의 경우에는 탐색 대상인 부분 공간의 시간 간격에 따라 그 시간동안 쓸고 지나간 체적으로 생각한다. 조건을 만족하는 영역은 다음과 같이 주어진다.

$$a_{ik} \leq \theta_i \leq b_{ik} \quad i = 1, 2, \dots, N$$

$$k = 1, 2, \dots, M_k$$

step4 : step3 에서 탐색된 영역으로부터 초기 상태점과 목적 상태점을 연결하는 트리를 구성한다.

step5 : 각 트리에 대하여 다음의 문제를 푼다.

$$\min \quad J(\theta_1, \theta_2, \dots, \theta_N)$$

$$\text{subject to} \quad a_i \leq \theta_i \leq b_i$$

step6 : 각각의 트리에 대한 최적화 해를 비교하여 성능 지수의 값이 가장 작은 것을 선택한다.

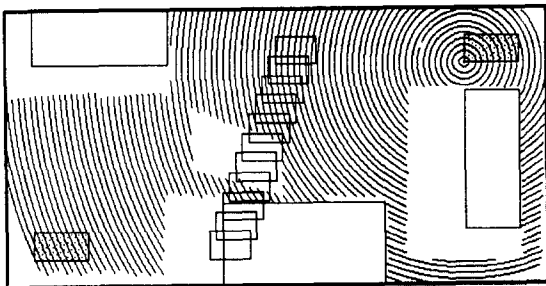


그림2. 관절 제한과 충돌회피 조건을 만족하는 θ 의 영역
Fig 2. The Range of θ from step3

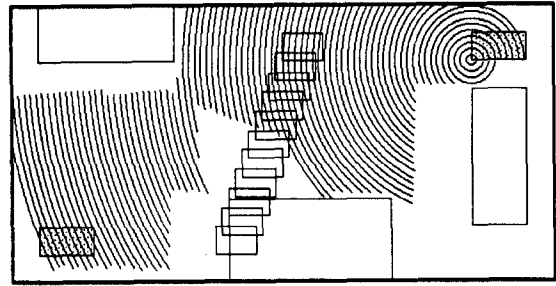


그림3. (a) 조건을 만족하는 θ 의 영역에서 구한 트리1
Fig 3. (a) The First Tree from The Range of θ

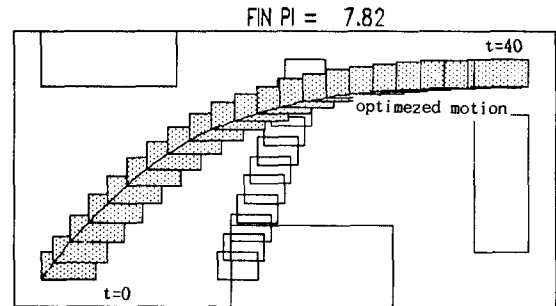


그림3. (b) 주어진 트리에서 구한 최적해
Fig 3. (b) Optimized Solution from Tree1

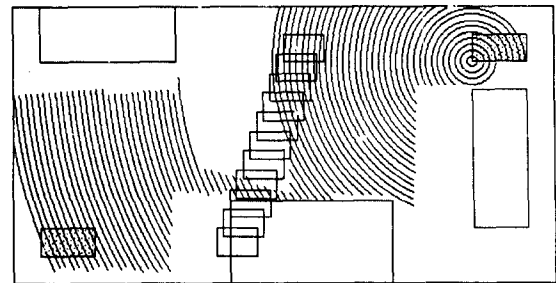


그림4. (a) 조건을 만족하는 θ 의 영역에서 구한 트리2
Fig 4. (a) The Second Tree from The Range of θ

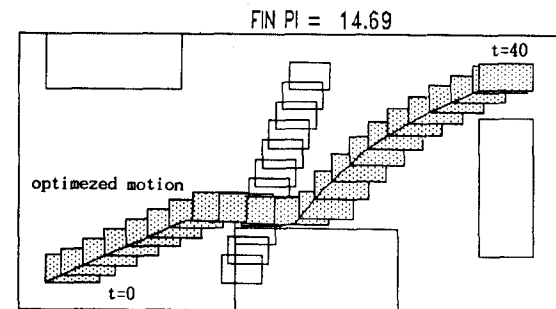


그림4. (b) 주어진 트리에서 구한 최적해
Fig 4. (b) Optimized Solution from Tree2

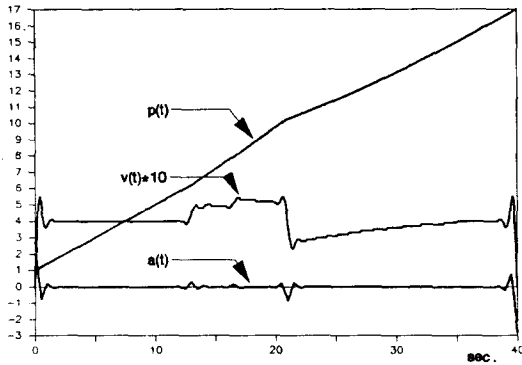


그림5. 그림3의 동작계획에서 x 축의 위치, 속도, 가속도
Fig 5. Position ,Velocity & Acceleration Profile

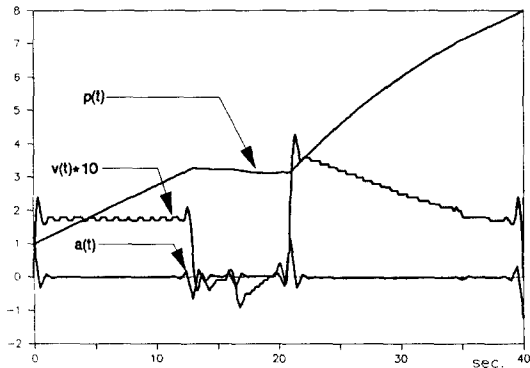


그림6. 그림3의 동작계획에서 y 축의 위치, 속도, 가속도
Fig 6. Position ,Velocity & Acceleration Profile

그림2에서 보여지는 동심원상의 호는 t_{samp} 를 0.5초로 할때 동작 계획 알고리즘의 `step1 - step3` 를 통하여 얻어진 관절 제한 조건과 충돌 회피 제한 조건을 만족하는 $\theta_i, i = 1, 2, \dots, N$ 의 범위를 나타낸다. $\theta_i, i = 1, 2, \dots, N$ 의 범위를 바탕으로 초기 상태에서 목적 상태로 진행할 수 있는 트리를 구한다. 주어진 작업 영역에서는 2개의 초기 상태와 목적 상태를 연결하는 트리가 존재한다. 그림3과 그림4는 각각의 트리와 이에 따른 최적화 동작 계획이다. 빗금친 각각의 사각형은 2초 간격으로 표시된 로봇이다. 성능 지수는 최단 이동 거리이며 이를 $\theta_i, i = 1, 2, \dots, N$ 의 함수로 표현하면 다음과 같다.

$$J(\theta_1, \theta_2, \dots, \theta_N) = \sum (\theta_i - \theta_{i+1})^2 \quad (15)$$

그림3과 그림4로 표시된 동작 계획중에서 그림3의 것이 더 작은 성능 지수값을 갖는다. 그림5와 그림6은 그림3으로 주어진 동작 계획에서 각 축의 위치($p(t)$), 속도($v(t)$), 가속도($a(t)$) 프로파일 (Position ,Velocity & Acceleration Profile) 을 나타낸다. 그림5는 x 축, 그림6은 y 축을 나타내며 속도는 10배의 크기로 표시되었

다.

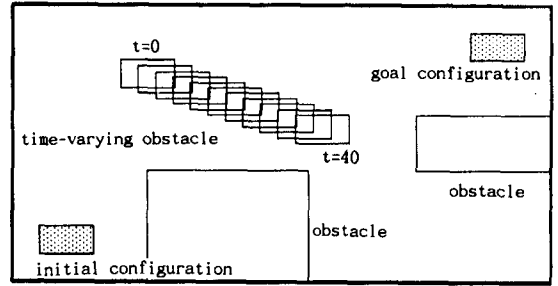


그림7. 동작 계획의 대상

Fig 7. Another Work Space

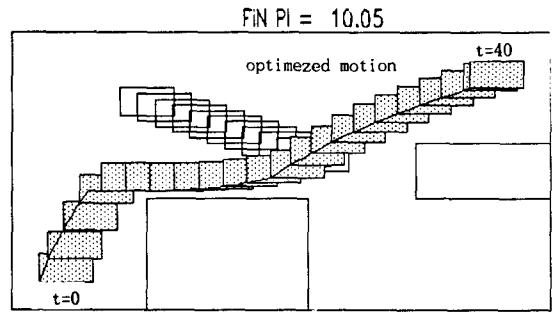


그림8. 최적화된 동작 계획

Fig 8. Optimized Motion Planning of Fig 7.

그림7은 다른 조건은 모두 같고 장애물의 위치가 다를때를 나타내며 그림8은 그림7의 작업 공간에서의 최적화된 동작 계획을 나타낸다.

5. 결론 및 연구 방향

자유도가 2이며 2차원상의 사각형(또는 다각형)으로 모델링된 로봇에 대한 충돌회피 동작 계획을 구현하였다. 형상 공간에 목적 상태를 원점으로 하는 극 좌표계를 도입하고 공간을 원 궤적으로 분할하여 최단 거리 동작 계획을 수립하였다. 성능 지수가 간단하며 θ_i 에 대한 제한을 구하는 것은 선을 따라 하는 탐색 (Line Search) 이므로 계산에는 많은 시간이 소요되지 않는다. 제안된 알고리즘을 통하여 얻어진 동작 계획은 각 축의 속도에 거의 변화가 없으며 따라서 큰 가속도를 필요로 하지 않는다.

로봇의 자유도가 3일 경우 구성되는 형상 공간은 3차원이다. 형상 공간이 3차원일때 목적 상태를 원점으로 하는 구 좌표계를 도입하여 구 궤적으로 공간을 분할하고 구면상에서 관절 제한 조건과 충돌 회피 조건을 만족하는 영역을 탐색한다. 3차원에서의 확장을 고려할때 관건은 탐색된 구면상의 영역을 어떻게 표현하며 선형화시킬 수 있을 것인가에 달려 있다.

두개의 관절을 가진 매니플레이터를 대상으로 할때는

개념적으로 문제되는 점은 없다. 2차원인 형상 공간을 분할한 후 제한 조건을 만족하는 영역을 탐색하고 위에서 언급한 최적화 알고리즘을 사용하면 된다.

4. 참고 문헌

- [1] David G. Luenberger, "Linear and nonlinear programming", Addison-Wesley publishing Co., 1984.
- [2] Mordecai Avriel, "Nonlinear programming", Prentice-Hall, Inc., 1976.
- [3] Elmer G. Gilbert, Daniel W. Johnson, S. Sathiya Keeritui, "A fast procedure for computing the distance between complex objects in three-dimensional space", IEEE Journal of Robotics and Automation, Vol.4, No.2, pp.193-202, April 1988.
- [4] Elmer G. Gilbert, Daniel W. Johnson, "Distance functions and their application to robot path planning in the presence of obstacles", IEEE Journal of Robotics and Automation, Vol.RA-1, No.1, pp.21-30, March 1985.
- [5] Elmer G. Gilbert, C. P. Foo, "Computing the distance between smooth objects in three-dimensional space", Proc. of the 1989 IEEE Int'l. Conf. on Robotics and Automation, pp.158-163, 1989.
- [6] B. H. Lee, Y. P. Chien, "Time-varying obstacle avoidance for robot manipulators : approach and difficulties", Proc. of the 1987 IEEE Int'l. Conf. on Robotics and Automation, pp.1610-1615, 1987.
- [7] B. H. Lee, "A study on the collision avoidance of two manipulators using velocity modifications", Trans. of KIEE, Vol. 37, Aug. 1988.
- [8] Tomas Lozano-Perez, "Spatial planning : A configuration space approach", IEEE Trans. on Computers, Vol. C-32, No. 2, pp.26-38, February 1983.
- [9] Tomas Lozano-Perez, "A simple motion-planning algorithm for general robot manipulators", IEEE Journal of Robotics and Automation, Vol.RA-3, No.3, pp.224-238, June 1987.
- [10] Ching Long Shin, Tsu-Tian Lee, William A. Gruver, "A unified approach for robot motion planning with moving polyhedral obstacles", IEEE Trans. on Systems, Man, and Cybernetics, Vol.20, No.4, pp.903 -915, July/August 1990.
- [11] P. M. Prenter, "Splines and variational methods", John Wiley & Sons, Inc., 1975.
- [12] William H. Press, Brian P. Flannery, Saul A. Teukolsky, William T. Vetterling, "Numerical Recipes in C", Cambridge University Press, 1988.