

충돌 회피를 위한 로봇 시뮬레이터의 개발에 관한 연구

이주형 · 정원희 · 신용탁 · 안두성

부경대학교

(1996년 6월 29일 접수)

A Study on Development of Robot Simulator for Collision Avoidance

Ju - Hyung LEE, Won - Hee JUNG, Yong - Tak SHIN and Doo - Sung AHN

National Pukyung University of Pusan, Dept. of Mechanical Engineering

(Received 29 June, 1996)

Abstract

This paper presents a robot simulator which can automatically generate a smooth collision-free path. This simulator has the characterstisc of the object-oriented programming which is coded in Borland C++. Using General Inverse Algorithm, the inverse kinematics solutions of any kind of robots can be found generally. This simulator also uses Bezier Functions for the description of a smooth collision-free path. In addition, GUI(Graphic User Interface)technique is employed for user's convenience.

서 론

현재 산업계에서 겪고 있는 심각한 인력난을 해결하고, 생산비의 절감과 다품종 소량 생산 체제로의 전환을 위하여 공장 자동화가 산업분야 전반에 걸쳐 진행되고 있으며, 그 실현 과정에서 가장 핵심적 역할을 하는 로봇의 기술이 더욱 발전하고 있다.

로봇 시스템에서는 작업 공간내에서 로봇의 동작을 방해하는 여러가지 물체나 콘베이어 벨트, 작업대 등 여러가지 장애물이 존재하게 된다. 그러므로 로봇이 작업 도중 이것들과 충돌하여 안전 사고나 로봇의 파손 등이 야기될 수 있으므로 이러한 장애물들과의 충돌을 회피하는 동작 계획 시스템이 필요하다. 작업 공간 내에서 장애물과의 충돌 회피에서 먼저 고려되어야 할 것은 장애물 내부에서 카메라로 인식되어진 물체 또는 고정된 위치

에서의 물체의 존재 여부이며, 존재시에는 그 물체의 위치, 형태, 방향 등의 기하학적 성질, 기구학적 성질과 물체의 유동 여부에 관한 정보와, 유동 물체의 경우 그 물체의 경로와 궤적에 대한 정보가 필요하다.

로봇은 이러한 정보와 자신의 기하학적 정보, 경로 및 궤적을 비교하여 로봇과 충돌 가능성을 판단하여 만약 충돌 가능성이 있으면 충돌 예정 위치와 시간 등을 구하여 충돌 회피 알고리즘을 수행하여 새로운 충돌 회피 경로와 궤적을 구성한다. 대부분의 충돌 회피에 대한 경로 계획은 작업 공간 안의 고정, 정지 물체를 피하는 문제에 이용된다^{1,2)}.

현재 대부분의 경우 로봇의 운동 경로 계획은 반복 학습 방법에 의하여 이루어지고 있으며 일련의 작업 계획을 온라인(on-line) 방식으로 수행하면 새로운 작업을 부여할 때마다 로봇의 작업을 정지시켜야 할 뿐 아니라, 작업 영역 내의 장애물과의

충돌을 예측하기 힘들고 이와 더불어 로봇 운동 경로의 최적화를 고려하기 어려운 난점이 있다. 지난 몇년간 로봇의 경로 계획을 온라인으로 수행하고자 하는 연구가 이루어져 왔으나 이는 계산 시간이 많이 걸리기 때문에 실시간 운영이 불가능하여 비효율적인 방법으로 간주 되었으며 따라서 로봇의 이용율을 높이고 운동 경로를 보다 최적화 시키기 위하여서는 로봇의 운동경로 계획을 실제 작업과 유사한 환경을 가진 시뮬레이터인 오프라인 프로그래밍(off-line programming) 시스템으로 수행하는 것이 바람직하다. 시뮬레이터는 기본적으로 로봇의 동작을 보여주게 되고, 로봇 동작의 결과를 나타내는 그래프를 보여주는데, 이를 위해 그래픽 패키지의 개발이 필요하게 된다.

본 논문에서는 로봇 이용의 자동화를 위하여 필수적 과정인 로봇의 운동 경로 계획에 관하여 연구하고 작업 영역 내에 고정, 정지된 장애물이 있는 경우에 로봇과 장애물과의 충돌 회피를 위한 정량적인 양으로 두 물체간의 최소거리 및 충돌 깊이를 효율적으로 계산할 수 있는 방법을 제안하며 이를 이용하여 로봇이 장애물과 충돌하지 않고 주어진 작업을 수행할 수 있는 경로를 자동적으로 생성함으로써 로봇을 장애물이 있는 환경 내에서도 운용하기 위한 기초를 마련하는데 목적이 있다.

충돌회피

1. 물체의 모델링 및 최소거리의 계산

로봇의 작업영역내 장애물이 존재하는 경우의 운동 경로를 계획하기 위하여, 물체 사이의 최소거리 및 충돌 깊이에 대한 정량적인 양의 계산은 필수적이며, 이는 물체를 모델링하는 방법에 따라 달라지게 된다.

1) 물체의 모델링

일반적으로 물체의 기하학적 형상을 표현하는 방법에는 여러가지가 있으며 이는 두 물체 사이의 거리를 계산하는데 큰 영향을 끼친다. 간단한 기하학적 형상을 가진 물체의 경우 해석적인 방법으로 물체를 표현할 수 있으며 이들 물체 사이의 최소거리 역시 해석적인 방법으로 간단히 구할 수 있다

그러나 해석적으로 표현하는 방법으로 물체를 표현하는데는 그 형상에 한계가 있으므로 물체를 수치적으로 표현하는 방법이 요구된다. 본 논문에서는 반평면(planar half space)을 이용하여 물체를 볼록다면체로 표현하였으며 다음 식으로 나타낸다.

$$C_i = \{x \mid \{n_{ik}^T x - b_{ik} \leq 0, k=1, \dots, N_i\} \quad (1)$$

여기서, C_i : 표현된 물체

n_i : 반평면의 수직 벡터

b_{ik} : 원점에서 반평면까지의 거리

N_i : 물체를 이루는 반평면의 갯수

x : 병진 이동 벡터

T : transpose

i : 표현된 물체의 index

물체의 위치(position, orientation)는 그 물체가 지니고 있는 자유도에 따라 위치가 변한다. 예를 들어, 컨베이어 벨트상의 물체는 컨베이어 벨트의 이동에 따라 움직이게 되고, 자유도에 따른 위치는 식(2)과 같다.

$$C_i(q,t) = \{x \mid (R_i(q,t)n_{ik})^T(x - P_i(q,t)) - b_{ik} \leq 0, k=1, \dots, N_i\} \quad (2)$$

여기서, R_i : 회전 변환 행렬

P_i : 병진 변환 행렬

x : 병진 이동 벡터

q : 물체의 위치

T : Transpose

C : 위치와 시간에 대한 함수

일반적으로 하나의 물체를 하나의 볼록다면체로 표현할 경우 모델링 오차가 커지게 되므로 하나의 볼록다면체로 구성하기 힘든 오목다면체(concave polyhedron)의 경우 이를 여러개의 볼록다면체로 구성된 하나의 물체로 표현할 수 있으며 다음 식으로 나타낸다.

$$C_i = C_i - 1UC_i \quad (3)$$

$i=2, \dots, I$

여기서,

I : 물체를 이루는 볼록다면체의 갯수

A, B : 볼록다면체를 이루고 있는 두 물체

두 볼록다면체 사이의 최소 거리를 d_{ij} 라 하고, 각각의 물체가 I_A, I_B 개의 볼록다면체로 구성되어 있다면, 두 물체(A, B)사이의 최소거리는 다음식으로 구할 수 있다.

$$d(C_A, C_B) = \min\{d_{ij} \mid i \in I_A, j \in I_B\} \quad (4)$$

여기서, d : 두 물체 사이의 거리
 C_A, C_B : 물체를 표현한 볼록다면체

2) 두 물체 사이의 최소거리의 계산

두 물체 사이의 거리는 장애물이 있는 경우 운동 경로 계획에 중요한 양(measure)이 된다. 그러나 두 물체 사이의 기하학적 거리는 서로 충돌하는 경우에는 0이 되므로 충돌 여부에 대한 판단은 불가능하나, 충돌 깊이에 대한 정보를 얻을 수 없다. 따라서 두 물체가 충돌시 그 충돌 깊이까지 고려하기 위하여 두 물체 사이의 최소 거리를 한 물체가 다른 물체와 단순 접촉(just contact)시키기 위하여 이동시켜야 할 병진 운동량으로 정의하면 다음 식으로 나타낼 수 있다.

$$d(A, B) = \min\{x \mid \text{contact}(A, B+x)\} \quad (5)$$

여기서, d : 이동시켜야 할 병진 운동량
 x : 병진 이동 벡터
 $B+x$: Minkowski set sum
 (Minkowski 집합의 합)

단순집합(Compact set)(K)에 대한 지지함수(support function) h_K 는

$$h_K(n) = \max\{n^T x \mid x \in K\} \quad (6)$$

이며, 접촉함수(contact function) SK는

$$S_K(n) = \{x \mid n^T x = h_K, x \in K\} \quad (7)$$

여기서, n : 반평면의 수직벡터
 이며, 이때 접촉함수($S_K(n)$)는 n 방향으로 compact set(K)상의 가장 먼 점을 나타낸다.

지지범함수 및 접촉함수는 물체의 기하학적 형상을 이용하여 다음 식으로 구할 수 있다.

$$h_K(n) = \max\{n^T x_i \mid x_i = 1, \dots, r\} \quad (8)$$

$$S_K(n) = x_{ij} = \min\{i \mid n^T x_i = h_K(n)\} \quad (9)$$

여기서, x_i : 물체의 꼭지점 좌표

주어진 두 개의 compact set(A, B)에 대하여 Minkowski set difference(Minkowski 집합의 차이)를 $K=A-B$ 라 하면, K는 다음 식으로 정의할 수 있다.

$$K = \{z \mid z = x - y, x \in C_A, y \in C_B\} \quad (10)$$

이때 K의 지지범함수 및 접촉함수는 다음 식으로 구할 수 있다.

$$h_K(n) = h_A(n) + h_B(-n) \quad (11)$$

$$S_K(n) = S_A(n) + S_B(-n) \quad (12)$$

여기서, n : 반평면의 전체 수직 벡터

식 (11)과 (12)를 이용하여 두 물체의 Minkowski set difference 역시 반평면으로 표현할 수 있다. 두 물체 (A, B)사이의 최소거리는 항상 두 물체의 경계면 사이에 존재하게 되므로 이 최소거리를 식 (5)와 식 (10)을 이용하여 나타내면 다음식으로 나타낼 수 있다.

$$d(K) = \min\{\|x\|, x \in bd(K)\} \quad (13)$$

K : 두 물체(A, B)의 Minkowski set difference($K=A-B$)

$bd(K)$: (K)의 경계(boundary of set)

두 물체(A, B)사이의 최소거리는 식 (13)으로 나타나며 두 물체(A, B)의 Minkowski set difference($K=A-B$)는 식 (11)과 식 (12)를 이용하여 반평면으로 표현이 가능하다. 두 물체가 서로 충돌하는 경우, 원점은 두 물체(A, B)의 Minkowski set difference($K=A-B$)의 내부에 존재하게 되며, 다음 식을 만족한다.

$$h_K(n_i) \geq 0, i = 1, \dots, N_K \quad (14)$$

따라서, 두 물체가 서로 충돌하는 경우의 최소거리는 식 (13), 식 (14)를 이용하여 다음 식으로 구할 수 있다.

$$d(K) = -\min\{h_K(n_i) : i = 1, \dots, N_K\} \quad (15)$$

N_K 는 두 물체의 Minkowski set difference(K)를 표현하기 위한 반평면의 갯수이고, 식 (15)의 음부호는 두 물체가 서로 충돌하였음을 나

타내는 지표이다. 식 (14)를 만족하지 않는 경우는 두 물체가 서로 충돌하지 않고 서로 떨어져 있는 경우를 나타내며 이 경우 두 물체 사이의 최소거리는 식 (13)을 이용하여 다음식의 최소화 문제로 귀결된다.

$$\text{최소화 함수} : f(x) = \min(|x|), x \in K \quad (16)$$

$$\text{구속조건} : g_i(x) \leq 0 \quad (17)$$

여기서, $g_i(x)$ 는 K 의 경계를 나타내는 반평면 방정식이 되며 식 (11)과 식 (12)을 이용하여 구할 수 있다. 식 (17)의 제한조건을 갖는 식 (16)의 목적함수에 대하여 Kuhn Tucker조건을 적용시키면 다음 식과 같다.

$$-\frac{x}{\sqrt{x^T x}} = \sum_{i=1}^{N_g} \lambda_i \bar{V}g_i(x) \quad \lambda_i \geq 0 \quad (18)$$

여기서, λ_i : 반평면의 수직벡터의 계수
 이때, x 는 K 의 경계상의 한점이며, $\bar{V}g_i(x)$ 는 x 에서 K 를 표현하는 반평면의 수직벡터가 된다.

3) 운동 경로 계획 방법

물체 사이의 최소 거리를 계산하는 목적은 장애물이 있는 경우에, 물체가 장애물과 충돌하지 않고, 주어진 초기위치에서 최종위치까지 운동하는 운동 경로를 계획하기 위함이다. Myers는 운동 경로 계획을 먼저 임의의 경로를 가정하고, 그 경로를 따라 충돌여부를 판단하며, 충돌이 일어나지 않는 경우에는 수행을 끝낸다. 충돌이 일어나는 경우 새로운 중간점(intermediate point)을 충돌이 일어나는 장애물의 경로 사이의 한 점에 수직인 평면에 있는 점으로 택하여 원래의 경로를 두 개로 나누어, 다시 각각의 경로에 대하여 충돌여부를 반복 수행하면서 경로를 계획하는 방법을 제안하였다⁵⁾. 본 논문에서는 두 물체 사이의 충돌 여부에 대한 판단 기준으로 두 물체 사이의 최소 거리를 이용하고, 충돌회피를 위한 경로의 변경을 물체가 지닌 형상 변수에 대한 두 물체 사이의 최소거리의 변화율을 이용하여 새로운 경로를 찾는 방법을 이용하였다.

3-1) 1 충돌 여부 판단

원래의 가정된 임의의 경로의 변경 여부는 물체

가 초기 위치에서 최종 위치까지 충돌하지 않고, 운동할 수 있는가에 달려 있다. 따라서, 원래의 경로를 따라 물체를 조금씩 이동시켜 가며 매 단계에서 물체와 장애물간의 충돌여부를 판단하여야 되는데, 이때 물체의 이동량이 너무 커지면 실제 충돌이 일어나는 경우를 간과할 수 있고, 또 이를 피하기 위하여 이동량을 작게 할 경우에는 전체적으로 계산 소요 시간이 많이 걸리게 되는 단점이 있다⁶⁾. 두 물체 사이의 최소 거리를 충돌 판단의 기준으로 사용하면, 물체의 이동량을 이들 거리에 의하여 조절할 수 있다. 예를 들면, 두 물체 사이의 거리가 큰 양수를 나타낼 경우, 이는 두 물체가 서로 멀리 떨어져 있는 경우를 가리키며, 따라서, 이 단계에서는 좀 더 큰 이동량을 가지고 운동할 수 있고, 작은 수일 경우, 물체가 장애물과 매우 근접한 경우를 나타내므로 이 단계에서는 좀 더 세밀한 운동을 시켜야 할 필요성이 있다는 판단기준이 될 수 있다.

3-2) 2 충돌 회피 경로 계획

미리 가정된 경로를 따라 물체를 이동시켜가며, 충돌 여부를 판단하여 충돌이 일어날 경우 원래의 경로를 수정할 필요가 있다. 이때, 두 물체의 최소 거리(이 경우 충돌 깊이)와 그 때 물체의 형상을 알고 있으므로, 이를 이용하여 경로를 수정하게 된다. 본 논문에서는 가정된 경로를 물체의 초기위치와 최종위치의 형상 변수값을 직선 보간법(linear interpolation)에 의하여 계획한다. 물체와 장애물이 충돌하지 않을 구속 조건은 다음과 같다.

$$d_i(x) - d_i^0 \geq 0, i = 1, \dots, N_0 \quad (19)$$

여기서, d_i : i 번째 장애물과의 최소거리

d_i^0 : 충돌 안전거리

N_0 : 장애물의 갯수

따라서, 물체와 장애물이 서로 충돌할 경우 위 식을 만족하지 않으므로 그 때의 형상변수 값을 다음식을 만족하는 방법으로 변화시켜 가며, 위 식과 아래 식을 만족하는 새로운 형상 변수의 값을 구한다.

$$V_{d_i}(x) \geq 0 \quad (20)$$

변수값을 결정한 뒤에는 curve fitting(곡선 조정)에 알맞은 Bezier function(Bezier 함수)를 사

용해서 생성하게 되는데, 이 함수의 특징은 지정된 경유점을 통과하여 각 구간의 연속성을 보장하고 근사직선의 생성을 쉽게 할 수 있다는 것이다.

2. Bezier 함수를 이용한 궤적 생성

Bezier 함수는 매개 함수로써 경로의 형성과 속도 조절이 용이하며, 연속성이 우수하다. 이러한 Bezier 함수의 장점을 이용하면 장애물 회피를 위한 궤적을 자동 생성할 수 있다^{7),8)}. Bezier 함수는 식(21)처럼 f_i 의 가중합이다.

$$f(t) = \sum_{i=1}^n f_i \binom{n}{i} t^i (1-t)^{n-i}, \quad 0 \leq t \leq 1 \quad (21)$$

여기서,

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}, \quad f_i : \text{Bezier points}$$

속도는 식 (22)과 같이 유도된다. 시작점과 끝점의 속도는 각각 양 끝단에서의 점들을 연결하는 기울기와 같다.

$$\frac{df(t)}{dt} = \sum_{i=0}^n (f_i \binom{n}{i} (it^{i-1}(1-t)^{n-i}) - t^i(n-i)(1-t)^{n-i-1}) \quad (22)$$

3. Bezier 함수의 특징

Bezier 함수는 미분값의 연속성을 보장하면서 Fig. 1과 같이 Bezier점을 어떻게 설계를 하느냐에 따라서 여러 형태의 Bezier곡선을 나타낸다.

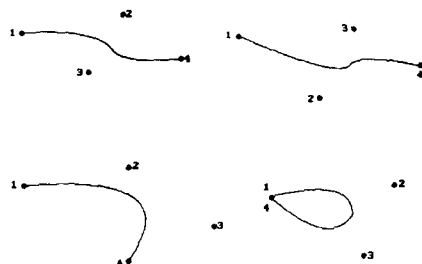


Fig. 1. The Bezier Curve

로봇 모델링 및 시뮬레이터

3차원 형상의 물체를 2차원 스크린상에 나타내는 방법에는 어떤 물체를 직선과 곡선의 집합체로 표현한 다음 투영을 통해 테두리를 표시하는 와이어 프레임(wire frame modeling)과 은면 제거 알고리즘(hidden surface removal algorithm)이나 셰이딩 알고리즘(shading algorithm)을 가미하여 보다 현실감 있게 물체를 표현하는 표면 모델링(surface modeling), 그리고 수학적인 고체(solid)로 어떤 물체를 표현하는 솔리드 모델링(solid modeling)으로 나누어진다. 본 논문에서는 필요한 정보의 양도 상대적으로 적고 계산 시간에서도 유리한 와이어 프레임 모델링 방법을 이용한다.

3차원 물체를 와이어 프레임 모델로 나타내기 위하여는 물체를 구성하는 꼭지점(vertex)들의 3차원적 좌표와 각 꼭지점 사이의 연결정보(connectivity information)만이 필요하다. 그러나 보이지 않는 선을 제거(hidden line removal)하여 와이어 프레임 모델을 확장하려면, 위 정보외에 면(edge)들로 구성되는 평면(surface)에 대한 정보가 필요하다.

1. 로봇 모델링

로봇은 여러 형태의 링크들이 축(joint)을 기준으로 특수한 형태로 결합되어 구성된다. 연속적인 두 링크는 축을 기준으로 회전하거나 직선이동을 하며, 각 링크는 특수한 형태를 갖는 3차원 요소로 이루어진다. 각 링크의 연결 상태는 링크 파라메타(link parameter)에 의해 결정되는 동차변환(homogeneous transformation)에 의해 결정된다.

본 논문에서는 손쉬운 로봇 모델링을 위하여, 사용자가 정의된 명령에 따라 명령어와 입력 데이터를 링크 순서대로 입력하여 file(*.MOD) 단위로 저장하면 인터프리터(interpreter)형식으로 3차원 로봇 형상이 나타내어진다. 보편적으로 쓰는 로봇 링크 및 tool 모델링 대상물을 형태별로 분류하여 각각 표준적인 모델을 만들고자 하였고, 본 논문에서는 가장 대표적인 6가지 형태의 표준 링크 및 tool을 선정하여 모든 형태의 로봇을 모델링하게

Table. 1 The standard link name and input data

링크명	입력 데이터
CUBE TYPE LINK	출발점, X, Y, Z축의 길이
CYLINDER TYPE LINK	원기둥 중심점, 반지름, 높이
PUMA TYPE LINK	링크축 중심점, 큰원, 작은원 반지름, 링크길이, 링크폭
SCARA TYPE LINK	링크축 중심점, 큰원, 작은원 반지름, 링크길이, 링크폭
GRIP TOOL	GRIP 중심점, X, Y, Z축의 길이
PRIS CYLINDER TYPE	원기둥 중심점, 원기둥 반지름, 기둥길이

하였다. 표준링크의 종류 및 입력 데이터는 Table. 1과 같다.

2. 월드 모델링

작업 공간내에서의 월드 좌표계(world coordinate system)에서 각 로봇의 위치와 방향을 나타내는 기준 좌표계의 절대적 위치와 이들 사이의 상대적 위치를 모델링 한다. 각 좌표계 사이의 관계는 동차변환으로 나타낼 수 있다.

3. 3차원 그래픽

본 논문에서 작성한 프로그램은 3차원 그래픽으로, 3차원적 구현의 필요성은 시뮬레이션한 결과를 사용자가 쉽게 이해하기 위해서는 현실감 있는 가시화가 필요하다. 실세계 좌표계(X_w, Y_w, Z_w)상에서 모델링한 좌표를 식 (23)으로 시각 좌표계(X_e, Y_e, Z_e)상으로 모델링한 후, 식 (24)로 스크린 좌표계(X, Y)상으로 이미지 투영을 함으로써, 로봇이 운동하는 매 시점에 대한 실제 좌표값들을 계산하여 가시화 할 수 있다.

$$\begin{bmatrix} X_e \\ Y_e \\ Z_e \\ 1 \end{bmatrix} = \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (23)$$

$$\begin{bmatrix} -\sin(\theta) & -\cos(\phi)\cos(\theta) & -\sin(\phi)\cos(\theta) & 0 \\ \cos(\theta) & -\cos(\phi)\sin(\theta) & -\sin(\phi)\sin(\theta) & 0 \\ 0 & \sin(\theta) & -\cos(\phi) & 0 \\ 0 & 0 & \rho & 1 \end{bmatrix}$$

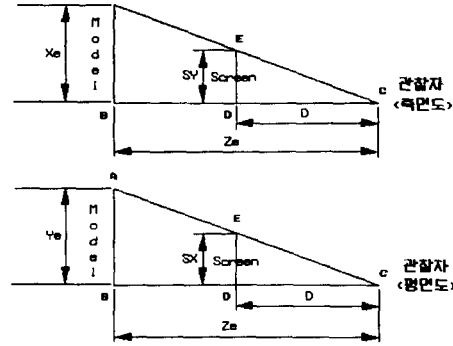


Fig. 2. The Projection of Perspective Change

여기서, θ, ϕ, ρ : 관절 파라미터

$$SX = \frac{D}{Z_E} \times X_E + CAMX_X / 2$$

$$SY = \frac{D}{Z_E} \times Y_E + CAMX_Y / 2 \quad (24)$$

4. 시뮬레이터

로봇의 움직임을 시각적으로 묘사하기 위한 기초로서 위에서 제시한 링크 모델링, 로봇 모델링, 월드 모델링 정보를 사용한다. 로봇의 움직임은 동작 계획으로부터 얻어진 연속적인 각 축의 회전 각도 $\theta_i (i=1, \dots, 6)$ 에 의해 표본시간(sampling time)마다 로봇의 모습을 스크린에 그린다. 관찰자의 좌표(viewing), 확대/축소(zooming) 등 여러 정보로부터 같은 동작에 대한 여러가지 묘사를 얻을 수 있다. 또한 유저-인터페이스(user-interface)를 이용하여 여러가지 변수값을 사용자가 쉽게 바꿀 수 있으며 이런 변수값에 대한 정보도 쉽게 볼 수 있도록 화면을 구성하였다. 그리고, 초기 화면에서 사용자는 로봇의 형태를 선택할 수 있으며, 선택된 로봇에 대한 역기구학 및 동역학 알고리즘을 추가, 변경하지 않아도 궤적이 자동생성이 된다.(Fig. 3. - Fig. 6.)

관절 공간에서 생성된 부드러운 궤적을 Fig.7.에서 보여주고 있다.

본 논문의 특징 :

1. GUI 방식(마우스,키보드 동시 입력 가능) 채택.
2. 순쉬운 Data verifying 기능.
3. 3차원 Cartesian 경로 Display.

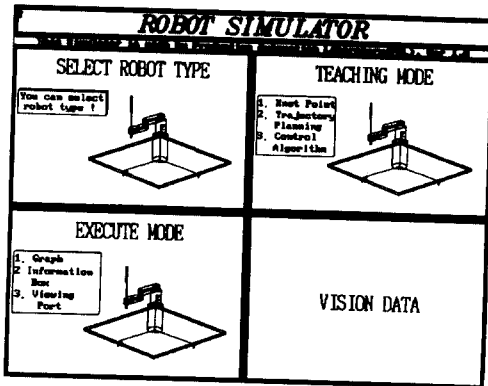


Fig. 3. The Menu Screen

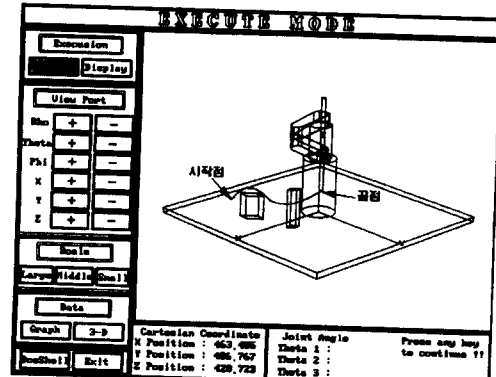


Fig. 6. The Bezier Curve at the Cartesian Space

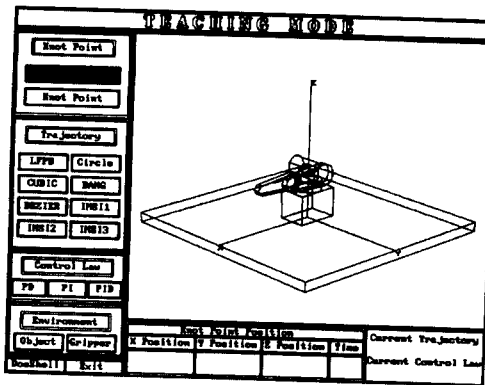


Fig. 4. The PUMA Robot

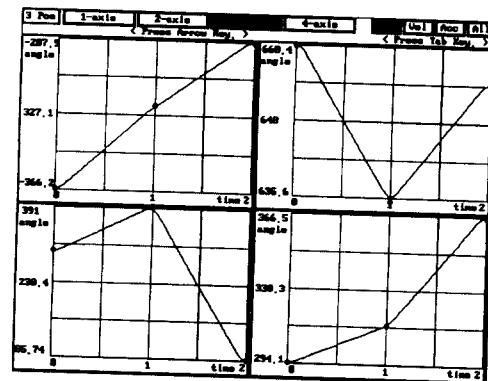


Fig. 7. The Graph of Joint Angle

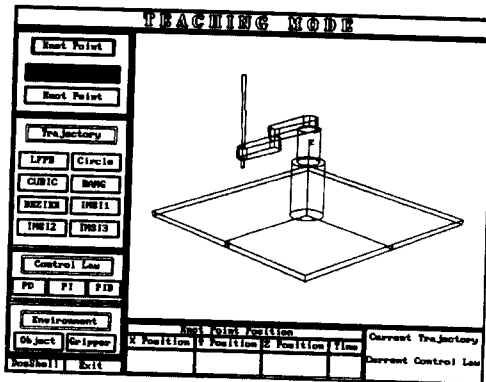


Fig. 5. The SCARA Robot

결 론

본 논문에서는 충돌 검색 및 장애물 회피 기능을 가진 알고리즘을 제시하였다. 본문의 Fig. 6과 Fig. 7에서 보는 바와 같이 장애물과의 충돌을 자동 검색, 회피 경로를 Bezier Curve를 사용하여 최단거리에 근접하면서도 부드러운 궤적을 자동적으로 생성할 수 있음을 검증해 보았다. 또한, GUI 방식을 채택함으로써 사용자가 작업 지정시 편리성을 제공하였으며, Borland C++의 객체지향적 특성을 최대한 살려 프로그래밍하였다.

그러나, 최단거리를 보다 더 효율적으로 구할 수 있는 알고리즘의 보강 및 실시간 장애물 회피에 관한 추가 연구를 향후 연구 방향이라 하겠다.

4. Teaching 시 설정된 로봇의 Work Space Display 기능.

5. General Inverse Kinematics, Dynamics algorithm 채택.

후 기

본 연구가 이루어질 때까지 격려해주고 연구비를 지원해 주신 늘원 재단에 감사의 뜻을 전하며, 이에 관계된 모든 분들께 감사드립니다.

참고 문헌

- 1) B. H. Lee, C. S. G. Lee, "Collision-free motion planning of two robots", *IEEE Transactions on systems, Man, and Cybernetics*, Vol. 17, No. 1, pp. 21-32, Jan/Feb. 1987.
- 2) B. H. Lee, "Wrist Collision Avoidance of Two Robots: A Collision Map and Time Scheduling Approach", *In Proc. 25th IEEE Cong. decision and Control, Athens, Greece*, pp. 429-434, Dec. 1986.
- 3) Kamal Kant, S. W. Zucher, "Toward efficient trajectory planning: The path-velocity decomposition", *The International Journal of Robotics Research*, Vol. 5, No. 3, pp. 72-89, Fall 1986.
- 4) Ching Long Shin, Tsu-Tian Lee, William A. Gruver, "A Unified Approach for Robot Motion Planning with Moving Polyhedra Obstacles", *IEEE Trans, on Systems, Man, and Cybernetics*, Vol. 20, No. 4, pp. 903-915, July/August 1990.
- 5) Myers John, K., "A Supervisory Collision Avoidance System for Robot Controller", *Master Thesis, Carnegie-Mellon University*, 1981.
- 6) E. Freund, "Hierarchical Control of Guided Collision Avoidance for Robots in Assembly Automation", *Proc. of 4th Int'l Conf. on Assembly Automation*, pp. 91-100, Sept. 1983.
- 7) Eliner G. Gilbert, aniel W. Johnson, "Distance function and their application to robot path planning in the presence of obstacles", *IEEE Journal of Robotics and Automation*, Vol. 1, No. 1, pp. 21-30, Mar. 1985.
- 8) Gilbert E. G and Johnson D. W, "Distance Function and Their Application to Robot Path Planning in the Presence of Obstacles", *IEEE Journal of Robotics and Automation*, Vol. RA-1, pp. 21-30, 1985.