'93 한국자동제어학술회의논문집 (1993. 10. 20~22)

충돌 회피를 위한 다중 로봇 시뮬레이터

° 이재용, 이범희
서울대학교 제어계측공학과

Multi-Robot Simulator for Collision Avoidance

° Jae-Yong Lee, Bum-Hee Lee
Dept. of Control & Instrumentation Eng., Seoul National University

Abstract

Robots working in the multiple robot system can perform the variety of tasks compared to the single robot system, while they are subject to the various tight constraints such as the precise coordination and the mutual collision avoidance during the task execution.

In this paper, we provide an algorithm and graphical verification for collision avoidance between two robots working together. The algorithm calculates the minimum time delay for collision avoidance and the graphical verification is performed through the 3-D graphic simulator.

1. 서론

최근 생산 시스템 분야에서는 생산비의 절감과 다양하게 변하는 생산 요구를 만족시키는 다품종 소량 생산 체제로의 전환을 위하여 생산 설비의 자동화, 고속화, 유연화를 추구하고 있다. 따라서 FMS 및 이의 가장 핵심적 역할을 하는 로봇의 기술은 더욱 발전되고 있다. 로봇은 조립 작업, 반송 작업, 기구 작업 등 다양한 동작 기능을 하는데, 이로써 생산성 증가, 원가 절감, 품질 향상을 이룰뿐만 아니라 자동화된 기계들과 함께 무인 공장 개념으로 발전되고 있다.

다중 로봇 시스템에서는 작업 공간내에서 로봇의 동작을 방해하는 여러가지 물체- 콘베이어 벨트, 작업대 등 -와 다른 로봇들이 존재하게 된다. 그러므로 로봇이 작업 도중 이것들과 충돌하여 안전 사고나 로봇의 파손 등이 야기될 수 있으므로 이들 장애물과의 충돌을 피하는 동작 계획 시스템이 필요하다. 이러한 충돌 회피 동작 계획에 대한 연구는 1970년대 후반부터 중요성이 대두되어 왔다.

작업 공간내에서 장애물과의 충돌 회피에서 먼저 고려 되어야 할 것은 작업 공간 내부에서의 물체의 존재 여부이며, 존재시에는 그 물체의 위치, 형태, 방향 등의 기하학적 성질, 기구학적 성질과 물체의 유동 여부에 관한 정보와, 유동 물체인 경우에는 이밖에 그 물체의 경로와 궤적에 대한 정보가 필요하다. 또한, 로봇의 모델도 충돌 회피를 위해서 필요한 정보이다.

로봇은 이러한 정보와 자신의 기하학적 정보, 경로 및 궤적을 비교하여 충돌 감지를 하여 만약 충돌 가능성이 있으면 충돌 위치와 시간 등을 구하여 충돌 회피 알고리즘을 수행하여 새로운 충돌 회피 경로와 궤적을 구성한다. 이런 일련의 과정을 충돌 회피 동작 계획이라 한다.

동작 계획은 경로 계획(path planning)과 궤적 계획(trajectory planning)으로 구성된다[1]. 대부분의 경로 계획은 작업 공간 안의 고정, 정지 물체를 피하는 문제에 이용된다. 보통 시불변 장애물(time invariant obstacle)에 대한 충돌 회피 동작 계획 문제는 고정 장애물에 대한 기하학적인 해석에 의한 경로 계획으로 풀 수 있다. 시변 장애물(time varying obstacle)에 대해서는 경로 계획 이외에 적절한 궤적 계획이 뒤따라야 한다. 이것은 시간에 따른 변화 즉, 속도를 고려한다는 것이다. 시변 장애물에 대한 충돌 회피에 관한 연구는 다음과 같다.

Lee[2]는 물체의 속도를 고려하여 작성된 충돌 지도를 기반으로 로봇의 동작을 시간 스케쥴링하여 구로 모델링된 로봇 팔목의 시변 환경 내에서의 충돌 회피에 관한 알고리즘을 제안 하였다. Kant[3]은 경로 계획과 속도 계획 문제를 분리하여 충돌 회피 동작 계획을 수행하는 알고리즘을 제안하였다. Lee와 Kant의 알고리즘은 속도 변경법을 이용하여 시변 장애물과의 충돌을 회피하는 동작 계획이다.

Erdmann[4]과 Lozano-Perez[5]는 로봇이 다른 시변 장애물에 대한 충돌 가능한 형상 범위를 단위 표본 시간(sampling time)마다 구성하여 작성된 형상공간-시간지도(configuration space-time map)를 이용하여 충돌 회피 동작 계획을 하는 방

법을 제안했다. Shin[6]은 형상 공간을 확장한 형상공간-시간 지도를 구성하여 여러대의 로봇이 주어진 우선 순위에 따라 시작점에서 목적점까지 일정한 속도로 동작하는 알고리즘을 제안했다.

Freud[7]는 계층 코오디네이션 제어(hierarchical coordination control)를 이용하여 다중 로봇 시스템의 동작을 온라인으로 제어하는 알고리즘을 제시 하였다.

본 논문에서 동작 계획 시스템의 최종 목표는 주어진 작업 공간안에서 두 대 이상의 로봇을 동작 시킬 때 다중 로봇 시스템의 특성을 살린 다양한 작업을 안전하게 수행할수 있도록 각 로봇의 동작 계획을 얻는 것이다. 이에 따라 만들어진 새로운 로봇 경로나 궤적에 대한 검사가 필요한데 어떤한 위험 부담없이 검사를 하기 위해 그래픽 시뮬레이터를 이용한다. 시뮬레이터는 여러가지 출력을 보여주게 되나 기본적으로 로봇의 동작을 보여주게 되는데 이를 위해 그래픽 패키지의 개발이 필요하게 된다.

이전에 개발된 시뮬레이터들로는 Stanford University에서 개발된 SIMULATOR, 1983년 동경대에서 개발된 EARLS-2, Nortingham University에서 개발된 GRASP, 그외에 ROBEX, 1984년 Cornell University에서 개발된 I-GRIP, 1986년 서독의 Scarland 대학에서 연구 개발된 ROBOSIM[8]이 있다. 그러나 이 시뮬레이터들은 충돌 발생을 배제 하였거나 충돌 발생에 대한 정보 출력이 미약하다. 이에 따라 다중 로봇의 충돌 회피를 위한 시뮬레이터의 개발이 필요하며, 더욱 확장되어 새로운 경로 계획을 자동적으로 세울수 있는 시뮬레이터로 확장이 필요하다.

본 논문에서 3차원 그래픽 기능을 가지는 충돌 회피를 위한 다중 로봇 시뮬레이터를 구성하기 위해 제 2 장에서 로봇 모델링 방법, 화면 구성 방법을, 제 3 장에서는 충돌 감지 방법, 충돌 회피 동작 계획 방법을 제시한다. 이로써 오프-라인에서 로봇 동작을 검증하여 충돌 발생시 이를 회피하는 경로, 궤적을 구할 수 있는 그래픽 시뮬레이터를 구성할 수 있다.

2. 로봇의 모델링 및 애니메이션(Animation)

2.1 로봇 링크의 모델링(Link Modeling)

3차원 형상의 물체를 2차원 스크린상에 나타내는 방법에는 어떤 물체를 직선과 곡선의 집합체로 표현한 다음 투영을 통해 테두리를 표시하는 와이어 프레임 모델링(wire frame modeling)과 은면 제거 알고리즘(hidden surface removal algorithm)이나 쉐이딩 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 나누어 진다. 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 연결 정보(connectivity information)만이 필요하다. 그러나 보이지 않는 선을 제거(hidden line removal)하여 와이어 프레임 모델을 확장하려면 위 정보외에 변(edge)들로 구성되는 평면(surface)에 대한 정보가 필요하다.

본 논문에서는 각 링크 모델링은 링크에 대한 기준 좌표계를 설정하고 기준 좌표계에 대한 각 꼭지점들의 상대적인 3차원 좌표값과 면을 구성하는 꼭지점들에 대한 정보를 저장하여 모델링 하였다. 이때 물체를 구성하는 꼭지점들의 순서는 우선 밑면을 바깥 방향에서 수직으로 보아 밑면을 구성하는 꼭지점들을 반 시계 방향으로 정하고 윗면에 대해 같은 방향으로 설정하였다. 각 면을 구성하는 꼭지점 순서에 대하여는 각 면을 바깥 방향에서 수직으로 보아 반 시계 방향 순서로 결정한다[9]. 그림2.1은 기준 좌표계에 대한 3차원 물체의 모델링 예이다.

2.2 로봇 모델링(Robot Modeling)

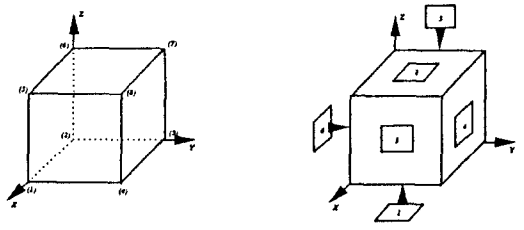
로봇은 여러 형태의 링크들이 축(joint)을 기준으로 특수한 형태로 결합되어 구성된다. 연속적인 두 링크는 축을 기준으로 회전하거나 직선 이동을 하며, 각 링크는 특수한 형태를 갖는 3차원 요소로 이루어 진다. 각 링크의 연결 상태는 링크 파라미터(link parameter)에 의해 결정되는 동차변환(homogeneous transformation)에 의해 결정된다. 그림 2.2는 로봇 모델링에 의해 묘사된 PUMA560의 모습이다.

2.3 월드 모델링(World Modeling)

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 동차 변환으로 나타낼수 있다.

2.4 애니메이션

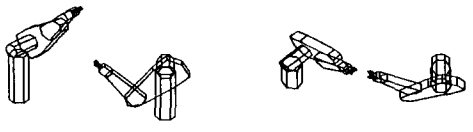
로봇의 움직임을 시각적으로 묘사하기 위한 기초로써 위에서 제시한 링크 모델링, 로봇 모델링, 월드 모델링 정보를 사용한다. 로봇의 움직임은 동작 계획으로 부터 얻어진 연속적인 각 축의 회전 각도 θi (i=1, … , 6)에 의해 표본 시간(sampling time)마다 로봇의 모습을 스크린에 그린다. 관찰자의 좌표(viewing), 확대/축소(zooming) 등 여러 정보로 부터 같은 동작에 대한 여러가지 묘사를 얻을 수 있다. 그림 2.3은 애니메이션에 대한 흐름도이다. 또한 유저-인터페이스(user-interface)를 이용하여 여러가지 변수값을 사용자가 쉽게 바꿀수 있으며 이런 변수값에 대한 정보도 쉽게 볼수 있도록 화면을 구성하였다. 그림2.4는 이런 화면 구성에 대한 그림이다.



	X	Y	Z
VERTEX 1	100	0	0
VERTEX 2	0	0	0
VERTEX 3	0	100	0
VERTEX 4	100	100	0
VERTEX 5	100	0	100
VERTEX 6	0	0	100
VERTEX 7	0	100	100
VERTEX 8	100	100	100

	VERTEX NUMBER
SURFACE 1	1 2 3 4
SURFACE 2	5 8 7 6
SURFACE 3	1 4 8 5
SURFACE 4	3 7 8 4
SURFACE 5	2 6 7 3
SURFACE 6	1 5 6 2

Fig2.1 Modeling Example



phi = 60 *phi = 30*

Fig2.2 PUMA560 Display Example

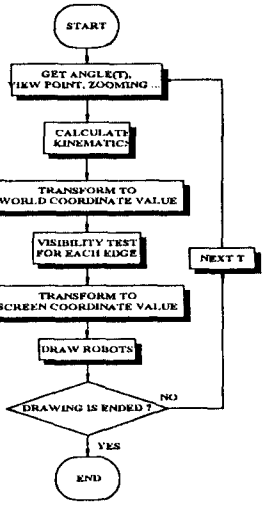


Fig2.3 Animation Algorithm

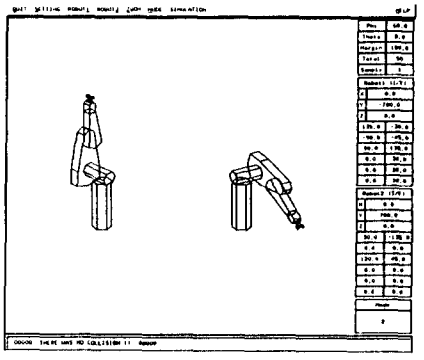


Fig2.4 Screen Display of Simulator

3. 시간 지연을 이용한 충돌 회피 동작 계획

현재 개발된 충돌 회피 알고리즘은 두 로봇들의 경로 정보를 이용하여 서로 간의 충돌을 감지하고 만약 충돌이 생길 경우 어느 한 쪽 로봇의 출발 시간을 지연시킴으로써 전체적으로 충돌을 방지하는 알고리즘이다. 이를 위해 다음과 같은 가정을 한다.

[가정]

① 대상은 같은 규격의 두 대의 PUMA560 로봇이다.

② 로봇1과 로봇2의 충돌은 엔드 이펙터(end-effector)와 링크 3,4,5,6에서만 발생한다.

③ 일정 시간(tsample)마다 축의 회전 각도 θ1~θ6을 받거나 경로 계획 루틴을 이용할 수 있다.

위 가정들은 조금씩 보완하면 일반화 시킬 수 있다.

3.1 링크의 근사화(Link Approximation)

가정에 의한 충돌 발생 가능 링크에 대해서는 충돌 감지를 위해 링크를 원통형으로 근사화 시켰다. 이는 계산량을 줄여 빠른 충돌 감지를 하기 위함이다. 각 링크에 대한 근사화는 다음과 같다.

[링크3,4 근사화]

링크3,4의 근사화는 그림3.1과 같이 링크3의 좌표계에서 원통형 근사화를 한다. 링크3, 4를 한 원통으로 근사화한 이유는 θ4의 변화에 따라 링크3, 4의 외부적 모양이 별로 변하지 않기 때문이다.

[링크5,6 근사화]

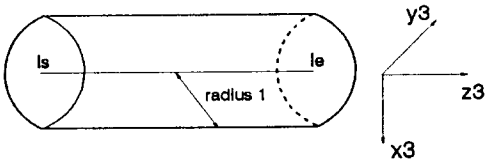
링크5,6의 근사화는 그림3.2와 같이 링크5의 좌표계에서 구형(sphere) 근사화한다. 링크5,6을 한 구로 근사화한 이유는 θ5의 변화에 따라 링크5,6의 외부적 모양 변화가 근사화된 구의 범위를 벗어나지 않기 때문이다. 또한 구로 근사화 되었으므로 엔드 이펙터의 회전은 근사화에 영향을 미치지 않는다.

3.2 충돌 감지(Collision Detection)

링크3,4는 원통형으로 근사화되어 졌으며 링크5,6은 구로 근사화되었으므로 가정에 의해 충돌은 원통과 원통, 원통과 구, 구와 구 사이에만 발생한다고 할수있다. 링크3,4간의 충돌 감지는 축간의 최단거리를 이용, 링크3,4와 링크5,6간의 충돌 감지는 축과 중심간의 최단거리를 이용, 링크5,6간의 충돌 감지는 중심간의 거리를 이용한다. 최단 거리는 다음과 같이 구한다.

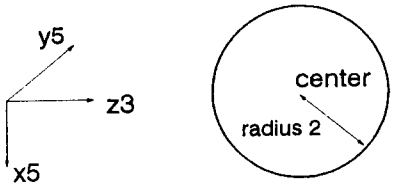
[축간의 최단거리]

그림3.3과 같이 링크3,4간의 최단 거리는 근사화된 링크3,4 중심축간의 최단거리, 즉 최단거리 d=min(d1, d2, d3, d4)로 한다. 이때 d>(2*radius1+Margin)이면 안전한 상태라 하고, d≤(2*radius1+Margin)이면 충돌 상태라 한다. 여기서 마진(Margin)은 임의의 입력값으로 근사화로 인해 발생하는 오차를 고려하기 위한것이다.



(ls.x, ls.y, ls.z) = (0, 0, -123)
(le.x, le.y, le.z) = (0, 0, 473)
LENGTH OF AXIS = 596mm
RADIUS 1 = 134.87mm

Fig3.1 Approximation For Link3,4



(center.x, center.y, center.z) = (0, 0, 58)
RADIUS 2 = 58mm

Fig3.2 Approximation For Link5,6

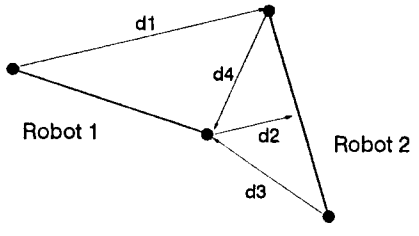


Fig3.3 Minimum Distance Between Axes

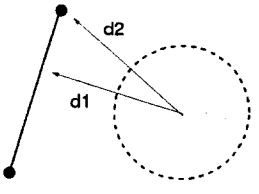


Fig3.4 Minimum Distance Between Axis And Center

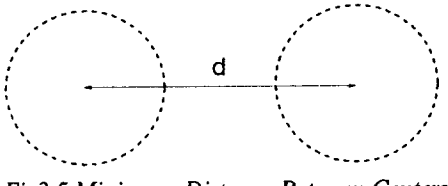


Fig3.5 Minimum Distance Between Centers

[축과 중심간의 최단거리]

그림3.4와 같이 링크3,4와 링크5,6간의 최단 거리는 근사화된 링크3,4의 중심축과 근사화된 링크5,6의 중심간의 최단 거리, 즉 최단 거리 d=min(d1, d2)로 결정한다. 이때 d>(radius1+radius2+Margin)이면 안전한 상태라 하고, d≤(radius1+radius2+Margin)이면 충돌 상태라 한다.

[중심간의 최단거리]

그림3.5와 같이 링크5,6간의 최단 거리는 근사화된 링크5,6의 중심간의 거리, 즉 최단거리 d=중심간의 거리로 한다. 이때 d>(2*radius2+Margin)이면 안전한 상태라 하고, d≤(2*radius2+Margin)이면 충돌 상태라 한다.

그림3.6은 충돌 감지 루틴의 흐름도이다.

3.3 충돌 회피 알고리즘(Collision Avoidance Algorithm)

본 논문에서 제안하는 충돌 회피 알고리즘은 매 표본 시간마다 경로 정보를 받아 두 로봇간의 충돌 여부를 결정하여 만약 충돌 발생시에는 어느 한쪽 로봇의 출발 시간을 지연시켜 충돌 방지를 한다. 즉 충돌 발생시 이 충돌을 피할수 있는 가장 작은 지연 시간을 구해 출발을 지연시키는 방법이다. 그림 3.7는 사용자 프로그램과 충돌 발생시 충돌 회피 루틴을 이용하고 난 후에 수정된 프로그램을 보여주고 있다. 또 근사화된 두 로봇의 충돌 회피 알고리즘의 주 흐름도는 그림3.8과 같다.

본 논문에서는 준최적 지연 시간(suboptimal delay time)을 구하기 위해 두가지의 탐색 알고리즘 - 블라인드 탐색(blind searth), 휴리스틱 탐색(heuristic search) - 을 제안한다.

앞에서 제시한 충돌 회피 동작 알고리즘의 흐름도는 일반적인 블라인드 탐색 방법을 나타내며, 또 주어진 경로에 대해 충돌을 방지하기 위한 준최적 지연 시간을 더욱 빨리 구하기 위한 휴리스틱 탐색 방법은 다음과 같다. 이와 휴리스틱 탐색 방법은 블라인드 탐색 방법에 비해 아주 빨리 solution을 구할 수 있으나, 전체 궤적중 충돌 발생 구간이 두구간 이상일때는 정확한 해를 구해낼수 없는 경우가 발생한다.

[Step1] 경로 정보와 전체 운동시간(T)을 입력한다.
(단 Td=0, Ttmp=0, Ts.d=0)

[Step2] 데이터에 따라 충돌 감지를 한다.

[Step3] 충돌 발생시 Ttmp ← T/2, Td ← T/2
그렇지않으면 [Step7]로 간다.

[Step4] Ttmp≥Ts이면 데이터에 따라 충돌 감지를
한다. 그렇지않으면 [Step7]로 간다.

[Step5] Ttmp ← Ttmp/2
충돌 발생시 Td ← Td + Ttmp
그렇지않으면 Ts.d ← Td, Td ← Td - Ttmp

[Step6] [Step4]로 간다.

[Step7] Ts.d를 출력한다.

(여기서 Td : delay time
Ts.d : suboptimal delay time
Ts : sampling time)

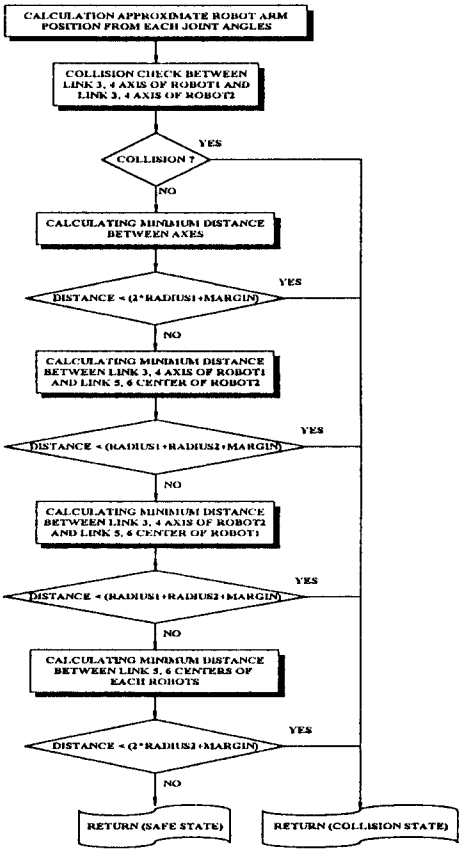


Fig3.6 Collision Detection Algorithm

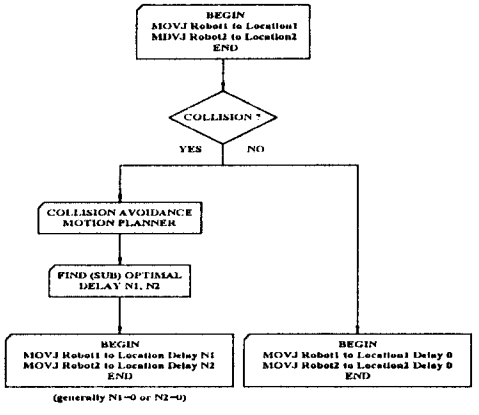


Fig3.7 Modified Programming By Collision Avoidance Motion Planning

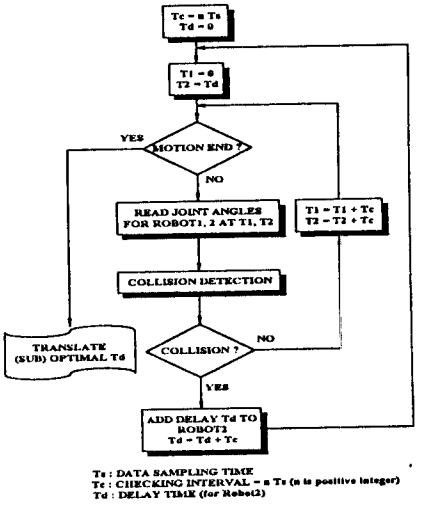


Fig3.8 Collision Avoidance Motion Plan Algorithm

앞에서 제시한 휴리스틱 탐색 방법과 일반적인 블라인드 탐색 방법의 비교 예는 표1,2와 같다.

표1. search algorithm 비교 예 (1)

	INITIAL CONFIGURATION (degree)
ROBOT 1	(180, 0, 90, 0, 0, 0)
ROBOT 2	(0, 0, 90, 0, 0, 0)
	FINAL CONFIGURATION (degree)
ROBOT 1	(0, 0, 90, 0, 0, 0)
ROBOT 2	(-180, 0, 90, 0, 0, 0)
TOTAL DURATION STEP	50 step * sampling time
SUBOPTIMAL DELAY TIME	17 * sampling time
HEURISTIC SEARCH	5 step
BLIND SEARCH	17 step

표2. search algorithm 비교 예 (2)

	INITIAL CONFIGURATION (degree)
ROBOT 1	(180, -90, 90, 0, 0, 0)
ROBOT 2	(0, -45, 45, 30, 60, 60)
	FINAL CONFIGURATION (degree)
ROBOT 1	(30, 0, 120, 0, 0, 0)
ROBOT 2	(-180, 0, 90, 0, 0, 0)
TOTAL DURATION STEP	50 step * sampling time
SUBOPTIMAL DELAY TIME	7 * sampling time
HEURISTIC SEARCH	4 step
BLIND SEARCH	7 step

표1,2에서와 같이 준최적 지연 시간이 크면 클수록 휴리스틱 탐색 방법이 훨씬 빨리 해를 찾아냄을 알 수 있다.

4. 시뮬레이션 결과

4.1 주어진 궤적에 대한 충돌 회피 동작

첫번째 시뮬레이션은 주어진 궤적에 따라 두 로봇이 동작을 할때 충돌이 발생한다. 이때 앞에서 제안한 방법으로 어떻게 궤적을 바꾸어 충돌을 회피하는지를 보여준다. 결과는 표3,4와 같다.

표 3. 수정 전 주어진 로봇의 궤적

주어진 로봇의 궤적(degree)			
로봇 1		로봇 2	
$\theta_1(t)=0.00168t^3-0.126t^2+135$		$\theta_1(t)=0.00168t^3-0.126t^2-30$	
$\theta_2(t)=0.00144t^3+0.108t^2-90$		$\theta_2(t)=-0.00072t^3+0.054t^2-45$	
$\theta_3(t)=-0.00048t^3+0.036t^2+90$		$\theta_3(t)=0.00144t^3-0.108t^2+135$	
$\theta_4(t)=0$		$\theta_4(t)=0.00048t^3-0.036t^2+30$	
$\theta_5(t)=0$		$\theta_5(t)=0.00048t^3-0.036t^2+30$	
$\theta_6(t)=0$		$\theta_6(t)=0.00048t^3-0.036t^2+30$	
로봇 베이스 위치			
ROBOT 1	(0, -750, 0)	ROBOT 2	(0, 750, 0)

표 4. 충돌 회피를 위해 수정된 로봇의 궤적

수정 후 로봇 궤적(degree)	
로봇 1 ($0 \leq t \leq 50$)	로봇 2 ($11 \leq t \leq 61$)
$\theta_1(t)=0.00168t^3-0.126t^2+135$	$\theta_1(t)=0.00168(t-11)^3-0.126(t-11)^2-30$
$\theta_2(t)=0.00144t^3+0.108t^2-90$	$\theta_2(t)=-0.00072(t-11)^3+0.054(t-11)^2-45$
$\theta_3(t)=-0.00048t^3+0.036t^2+90$	$\theta_3(t)=0.00144(t-11)^3-0.108(t-11)^2+135$
$\theta_4(t)=0$	$\theta_4(t)=0.00048(t-11)^3-0.036(t-11)^2+30$
$\theta_5(t)=0$	$\theta_5(t)=0.00048(t-11)^3-0.036(t-11)^2+30$
$\theta_6(t)=0$	$\theta_6(t)=0.00048(t-11)^3-0.036(t-11)^2+30$

여기서 θi는 i번째 축의 회전 각도를 나타내며 총 동작 시간은 (50*sampling time)이며 마진은 100mm이다. 이때 제시된 충돌 회피 동작 알고리즘에 의해 구해진 준최적 지연 시간은 (11*sampling)이다. 그림4.1과 그림4.2는 이 회피 동작에 대한 그림이다.

4.2 로봇 베이스 변화에 따른 지연 시간 비교

두번째 시뮬레이션은 로봇 베이스간의 거리 변화에 따라 준최적 지연 시간을 비교해 보았다. 모든 시뮬레이션 환경은 첫번째 시뮬레이션 환경과 동일하다(단, 베이스 거리 제외). 이때의 결과는 그림4.3과 같다.

결과로부터 베이스간의 거리가 1800mm 이상이 되면 충돌이 발생하지 않음을 알수 있다. 이런 결과를 이용하여 로봇 최적 배치 및 공동 작업 영역 설정 등에 적용할 수 있다.

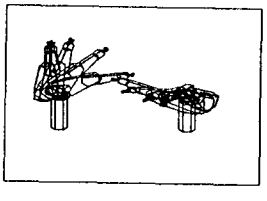


Fig4.1 Display for Collision State

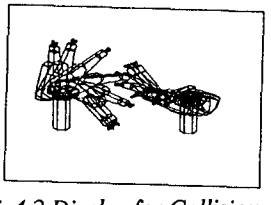


Fig4.2 Display for Collision Avoidance Motion

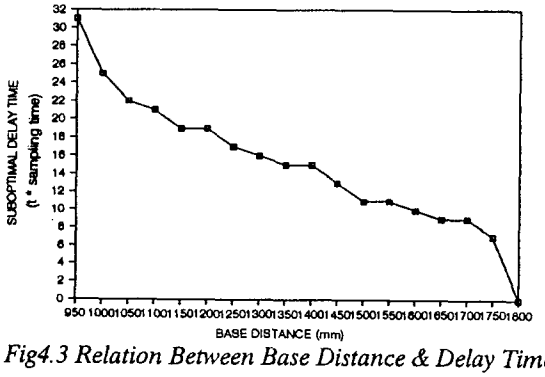


Fig4.3 Relation Between Base Distance & Delay Time

5. 결론

이 논문에서는 3차원 작업 공간에서 작업하는 다중 로봇 시스템에서 로봇간의 충돌 검색, 출발 시간 지연을 이용한 충돌 회피 동작 계획 방법 및 적용 방안을 제안하였다. 또한 이런 모든 과정을 오프-라인에서 안전하게 검증할 수 있게 3차원 그래픽 시뮬레이터와 접목하여 시각적으로 검증할 수 있다. 특히 여러가지 충돌 회피 동작 계획이 2차원상이나 작은 자유도의 로봇에만 실제 적용되었으나 본 논문에서는 3차원 PUMA560 로봇을 대상으로 알고리즘을 적용시켰으며 이를 확장하여 여러개의 시변 장애물에 대한 충돌 회피 알고리즘으로 적용 시킬 수 있다.

그러나 로봇의 최종 기하학적인 위치가 다른 로봇의 경로상에 위치할 때는 이 알고리즘으로는 충돌을 피할 수 없는 단점이 있다. 그러므로 앞으로 여러가지 다른 충돌 회피 동작 계획 방법을 적용시키고, 그래픽 부분을 좀더 발전시킨다면 현장의 다중 로봇 시스템에서 유용한 시스템이 될것이다.

6. 참고 문헌

[1] B.H.Lee, C.S.G.Lee, "Collision -free motion planning of two robots", IEEE Transactions on Systems, Man, and Cybernatics, 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 Conf. Decision and Control, Athens, Greece, pp.429-434, Dec. 1986.

[3] Kamal Kant, S.W.Zucker, "Toward efficient trajectory planning: The path-velocity decomposition", The International Journal of Robotics Research, Vol.5, No.3, pp.72-89. Fall 1986.

[4] Elmer 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.

[5] Tomas Lozano-Perez, "Spatial planning : A configuration space approach", IEEE Journal of Robotics and Automation, Vol. RA-3, No.3, pp.224-238, June 1987

[6] 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

[7] E.Fruend, "Hierarchical Control of Guided Collision Avoidance for Robots in Assembly Automation", Proc. of 4th Int'l Conf. on Assembly Automation, pp.91-10, Sept. 1983.

[8] D.W.Wloka, "ROBSIM-A Robot Simulation System", IEEE Int'l Conf. on Rob. and Auto. pp.1859-1864. 1986.

[9] J.McGregor, A.Watt, The Art of Graphics for the IBM PC, Addison-Wesley, 1986.