

# 다관절 지능 로봇시스템을 위한 장애물 우회 연구

신 현 배\*, 이 병 룡\*\*

## A Study on Collision Avoidance for Multi-link Intelligent Robots

Hyun-Bae Shin\*, Byung-Ryong Lee\*\*

### ABSTRACT

In this paper, a motion control algorithm is developed by using fuzzy control technique, which makes a robot arm avoid unexpected obstacles when the robot is moving from the start to a goal posture. During the motion, if there exist no obstacles the robot arm moves along the pre-defined path. But if some obstacles are recognized and close to the robot arm, a fuzzy controller is activated to adjust the path of the robot arm. To show the feasibility of the developed algorithm, numerical simulations and experiments are carried out. In the experiments, redundant planar robot arms are considered for the collision avoidance test, and it was proved that the developed algorithm gives good collision avoiding performance.

**Key Words** : Redundant Robot(여유자유도 로봇), Fuzzy Logic(퍼지논리), Collision Avoidance(충돌회피), Posture Optimization(자세 최적화)

### 1. 서 론

로봇이 작업 영역에서 이동 중일 때 발생하는 장애물을 우회하는데 사용되는 경로 계획법은 크게 2가지로 구분할 수 있는데, 한 방법은 상태공간(configuration space)법이며 다른 방법은 인공전위장(artificial potential field)법이다. 상태공간법<sup>(1)(2)</sup>은 로봇의 관절 운동이 n-차원(n-dimension)의 상태공간에서 점의 운동으로 변환이 되며, 장애물 또한 상태공간에서 장애물 영역으로 치환(mapping)된다. 따라서, 장애물과 충돌이 발생하지 않는 안전한 경로를 구할 수 있게 되며, 이 경로

로부터 각 관절의 회전각을 산출하는 방법이다. 이 방법은 계산 시간이 많이 들며, 장애물의 위치를 사전에 알고 있어야 하는 문제점을 가지고 있다. 인공전위장법<sup>(3)(4)</sup>은 인공적인 전위장 함수를 이용하는 방법인데 장애물이 있는 위치는 전위장값을 높게 할당하며, 로봇이 도달해야 하는 목표 위치는 전위장값을 최소값으로 할당해서 로봇이 목표 위치에 도달하게 하는 방법이다. 그러나, 이 방법은 이동용 로봇에서는 쉽게 적용이 가능하나 관절형 로봇에 적용하기는 매우 복잡하다. 또, 로봇이 목표 위치를 찾는 중에 국소최소점(local minima)<sup>(5)</sup>에 빠지는 경우가 발생할 수도 있는 문제점을 가지고 있다.

\* 울산대학교 대학원 기계공학과  
 \*\* 울산대학교 기계공학부

최근에는 퍼지 및 신경회로망 기법이 로봇의 궤적 추적 또는 장애물 우회를 위한 경로 계획에 대해 많이 적용되고 있다. 퍼지 및 신경회로망 기법은 앞에서 언급한 상태 공간법이나 인공지능위장법에 비하여 시스템을 해석하기 위한 모델링이 간단하며, 경로 탐색 알고리즘을 작성하는데 있어서 복잡한 기구학 및 동역학적 계산 식을 고려할 필요가 없으며 단지 전문가의 경험에 의해 학습을 시킬 수 있다는 장점이 있다. Raju<sup>(6)</sup>, Nedungadi<sup>(7)</sup> 등은 퍼지규칙을 이용하여 관절형 로봇의 역기구학을 계산하지 않고 임의의 시작 위치에서 목표 위치까지의 경로를 단순히 추종하도록 하는 연구를 수행하였다. Chen<sup>(8)</sup> 등은 로봇의 작업영역에서 장애물들이 존재할 때 장애물 때문에 로봇이 접근할 수 없는 영역을 신경회로망을 이용하여 학습을 시켜 로봇이 시작 위치에서 목표 위치까지 장애물을 우회하여 도달할 수 있도록 하였다. 그러나, 이 경우 돌발적으로 발생할 수 있는 미지의 장애물들과의 충돌을 예방할 수 없는 문제점이 있다. Bagchi<sup>(9)</sup> 등은 퍼지논리를 이용하여 여유자유도를 가진 로봇을 임의의 시작 위치에서 목표 위치까지 이동하면서 이동 중에 발생하는 장애물과의 충돌을 예방할 수 있음을 보였다. 그러나, 이 논문에서는 로봇 팔의 끝점의 위치가 다음 위치로 이동할 시점에서 장애물과 로봇 팔과의 충돌 가능성을 최소화하기 위한 최적화(optimization) 문제는 다루지 않았다.

본 논문에서는 로봇의 이동 중에 일정한 영역 내에서 장애물이 감지되면 퍼지논리를 이용하여 실시간적으로 장애물을 우회하도록 하며, 또 우회하는 도중에 로봇과 장애물과의 충돌 위험을 최소화하는 최적자세(optimal posture)를 유지하면서 목표 위치까지 로봇을 이동시킬 수 있는 최적화 경로 계획 알고리즘을 개발하였다. 그리고, 개발된 알고리즘을 여유자유도 로봇에 적용하여 개발된 알고리즘이 돌발적으로 발생하는 미지의 장애물을 우회하여 목표 위치까지 도달할 수 있음을 전산 모의 실험과 실험을 통해서 보였다.

## 2. 여유자유도 로봇의 장애물 우회 알고리즘

여유자유도 로봇은 작업을 수행하기 위하여 필요한 자유도(degree of freedom)보다 더 많은 자유도를 가지고 있는 경우를 말한다. 어떤 로봇 팔의 끝위치가 주어질 때 끝위치를 만족하는 로봇 팔의 자세는 무수히 많이 존재하게 된다. 따라서, 이러한 여유자유도를 이용하여 원하는 로봇의 작동요소를 최적화할 수 있다는 장점이 있다. 예

를 들어 로봇 팔의 작동 중에 소요되는 에너지를 최소화 시킨다든지, 특이점(singular point)을 피한다든지, 원하는 궤적을 따라서 속도 및 힘의 전달율(transmission ratio)을 최대화할 수 있는 등의 작업을 수행할 수 있다. 본 논문에서는 여유자유도를 가진 로봇이 움직일 때 돌발적으로 발생하는 장애물을 최적의 자세를 유지하면서 우회할 수 있는 알고리즘을 개발하고자 한다.

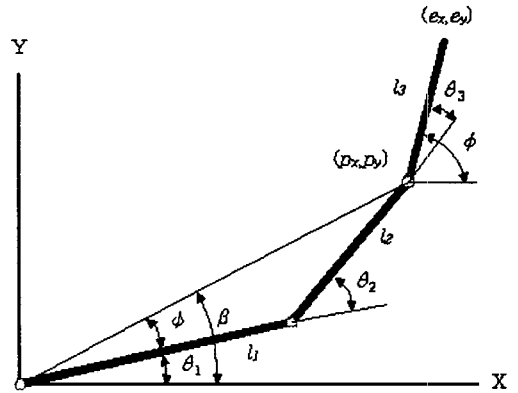


Fig. 1 Modeling of 3-axis Horizontal Multi-Joint Robot

### 2.1 역기구학(Inverse Kinematics)

여유자유도를 갖는 로봇 팔의 모델은 2차원 (X,Y) 좌표계에서 움직이는 3자유도 수평관절 로봇으로 구성할 수 있다.

Fig.1 에서 로봇 팔의 끝점의 좌표 ( $e_x, e_y$ )와 로봇 팔의 3번째 링크가 (X,Y)좌표와 이루는 각도를  $\phi$  라고 하면, 로봇 팔의 3번째 링크의 원점의 좌표 ( $p_x, p_y$ )의 위치는 식 (1)과 같이 구해진다.

$$\begin{aligned} p_x &= e_x - l_3 \cos\phi \\ p_y &= e_y - l_3 \sin\phi \end{aligned} \quad (1)$$

삼각함수의 공식으로부터

$$\begin{aligned} p_x^2 + p_y^2 &= l_1^2 + l_2^2 - 2l_1l_2 \cos(180 - \theta_2) \\ &= l_1^2 + l_2^2 + 2l_1l_2 \cos\theta_2 \end{aligned} \quad (2)$$

위 식을 정리하면

$$\begin{aligned} \cos\theta_2 &= \frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2l_1l_2} \\ \sin\theta_2 &= \sqrt{1 - \cos^2\theta_2} \end{aligned} \quad (3)$$

식 (2)로부터 2번째 관절각  $\theta_2$ 는 식 (4)와 같이 구해진다.

$$\theta_2 = a \tan 2(\sin \theta_2, \cos \theta_2) \quad (4)$$

Fig.1에서

$$\beta = a \tan 2(p_x, p_y)$$

$$\phi = \cos^{-1} \left[ \frac{p_x^2 + p_y^2 + l_1^2 - l_2^2}{2l_1 \sqrt{p_x^2 + p_y^2}} \right] \quad (5)$$

그러므로,  $\theta_1, \theta_3$ 는 다음과 같이 구해질 수 있다.

$$\theta_1 = \beta - \phi \quad (6)$$

$$\theta_3 = \phi - \theta_1 - \theta_2$$

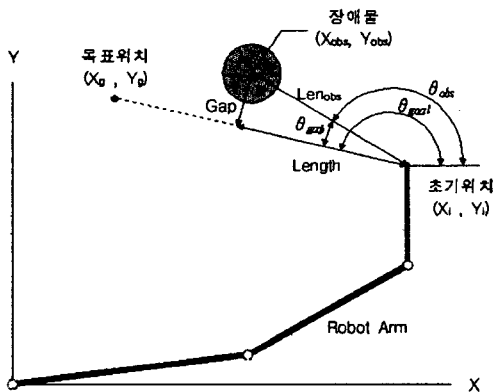


Fig. 2 Modeling of Robot Arm and Obstacle

### 2.2 퍼지논리를 이용한 장애물 우회 계획

Fig.2는 로봇이 움직일 때 돌발적으로 발생한 장애물이 로봇 팔의 끝점의 진행방향 가까이 놓여져 있는 모습을 보여주고 있다. 퍼지제어를 이용하기 위하여 필요한 입력 변수는 Gap과 Length로 선정하였다. Gap은 로봇 팔의 끝점(end effector)이 진행방향의 선상으로부터 장애물의 법선방향으로 떨어진 거리로 정의되고, Length는 장애물이 로봇 팔의 끝점에서 로봇 팔의 진행방향으로 떨어져 있는 거리로 정의된다.

Fig.2에서  $\theta_{goal}$ ,  $\theta_{obs}$  그리고  $Len_{obs}$ 는 아래와 같이 표시된다.

$$\theta_{goal} = \tan^{-1} \left( \frac{y_g - y}{x_g - x} \right) \quad (7)$$

$$\theta_{obs} = \tan^{-1} \left( \frac{y_{obs} - y}{x_{obs} - x} \right) \quad (8)$$

$$Len_{obs} = \sqrt{(x_{obs} - x)^2 + (y_{obs} - y)^2} \quad (9)$$

$\theta_{gap}$ 을 다음과 같이 정의하면

$$\theta_{gap} = \theta_{goal} - \theta_{obs} \quad (10)$$

퍼지제어의 입력변수 Gap과 Length는 다음과 같은 식으로 표시된다.

$$Gap = Len_{obs} \cdot \sin \theta_{gap} \quad (11)$$

$$Length = Len_{obs} \cdot \cos \theta_{gap}$$

그리고, 퍼지제어에 사용된 출력변수는  $\Delta\theta$ (로봇 팔의 끝점의 이동방향 수정각)로 선정하였다.

입력변수의 퍼지 세트 Gap은 NB(Negative Big), NS(Negative Small), ZE(Zero), PS(Positive Small), PB(Positive Big)으로 선정하였고, Length는

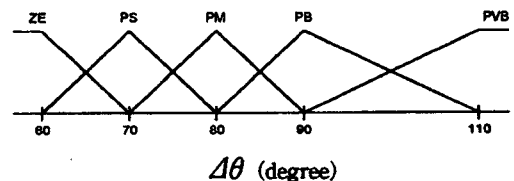
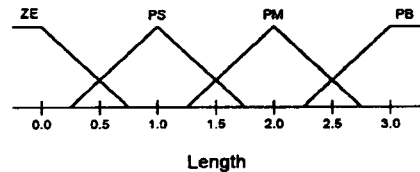
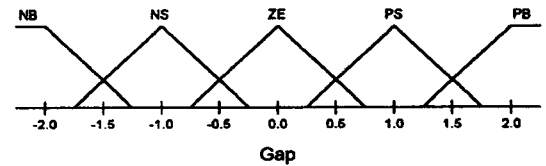


Fig. 3 Membership Function of Input and Output Variables

ZE(Zero), PS(Positive Small), PM(Positive Medium), PB(Positive Big)으로 선정하였다. 출력변수에 대한 퍼지 세트  $\Delta\theta$ 는 ZE(Zero), PS(Positive Small), PM(Positive Medium), PB(Positive Big), PVB(Positive Very Big)으로 선정하였다.

소속함수(fuzzy membership function)는 Fig.3에 나타나 있다.

입력 및 출력변수의 퍼지세트에 대한 퍼지제어규칙인 FAM(Fuzzy Associative Memory)은 표 1에 나타나 있다. FAM이란 입력변수와 출력변수의 퍼지세트간의 논리적으로 타당한 IF~THEN 규칙을 함축적으로 나열한 것이라고 볼 수 있다. 예를 들어 FAM의 좌측상단의 블록을 달리 표현하면 다음과 같은 논리규칙으로 표시된다.

IF Length is ZE and Gap is NB,  
THEN  $\Delta\theta$  is PVB.

그리고, 퍼지제어기는 FAM을 이용하여 임의의 입력변수가 입력되었을 때 비퍼지화(defuzzification method) 방법을 이용하여 명확한 출력값을 제공하게 된다

Table 1. 퍼지제어규칙(FAM)

		Length			
		ZE	PS	PM	PB
Gap	NB	PVB	PVB	PVB	PB
	NS	PVB	PVB	PB	PB
	ZE	PVB	PB	PB	PM
	FS	PB	PB	PM	PS
	PB	PB	PM	PS	ZE

### 2.3 자세 최적화 (Posture Optimization)

여유자유도를 갖는 로봇 팔의 속도는 높은 차원의 관절 공간(joint space)에서 낮은 차원의 작업공간(task space)으로 식 (12)와 같이 치환(mapping)이 된다.

$$\dot{X} = J(\Theta)\dot{\Theta} \quad (12)$$

여기서,  $X = [x_1, x_2, \dots, x_m]^T$ 는 작업좌표 벡터이고,  $\Theta = [\theta_1, \theta_2, \dots, \theta_n]^T$ 는 관절좌표 벡터이며,  $J$ 는  $m \times n$ 자코비안 행렬이다. 식 (12)에 대한 역기구학식은 식 (13)과 같이 구해진다.

$$\dot{\Theta} = J^+ \dot{X} + (I - J^+ J)K \quad (13)$$

식 (13)에서  $J^+$ 는 자코비안 행렬의 수도-인버스(pseudo-inverse)이며 아래의 식과 같이 정의된다.

$$J^+ = J^T (J J^T)^{-1} \quad (14)$$

그리고, 식 (13)에서  $K$ 는 임의의 벡터이다. 식 (13)에서  $(I - J^+ J)K$  값은 로봇 팔의 내부운동(internal motion)을 설명해 주는 항목이다. 즉, 로봇 팔의 끝점이 고정되어 있다 하더라도 여유자유도를 갖는 로봇 팔은 무수히 많은 자세를 가지게 되는데, 바로 이 현상을 설명해주는 식이다. 식 (13)의 내부운동 중에서 로봇 팔이 원하는 성능함수(performance criterion)를 최대화 시켜 주도록 벡터  $K$ 의 값을 결정하여야 한다. 성능함수를  $P(\Theta)$ 라고 했을 때,  $P(\Theta)$ 를 최대화하기 위한 벡터  $K$ 의 관계는 식 (15)와 같이 구해진다.

$$K = \frac{\partial P}{\partial \Theta} \kappa \quad (15)$$

식 (15)에서  $\frac{\partial P}{\partial \Theta} = [\frac{\partial P}{\partial \theta_1}, \frac{\partial P}{\partial \theta_2}, \dots, \frac{\partial P}{\partial \theta_n}]^T$  그리고  $\kappa$ 는 임의의 양의 상수이다. 따라서, 식 (15)의 관계가 유지되게 하는 내부운동(internal motion)은 항상  $P(\Theta)$ 를 증가시키는 방향으로 움직인다는 것을 의미한다.

본 논문에서는 로봇 팔이 이동중에 발생될 수 있는 장애물에 충돌하지 않고 안전하게 우회하여 목적지에 도달하는 것이 중요한 요소이므로, 장애물이 존재하고 로봇 팔의 끝점의 위치가 결정된 경우에 로봇 팔의 각각의 링크(link)가 Fig.4에서 보는 바와 같이 장애물로부터 가

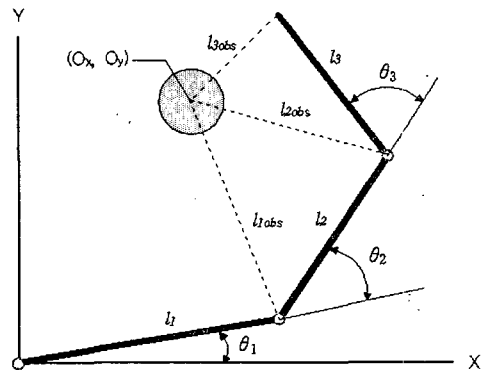


Fig. 4 Configuration of Robot Arm and Obstacle for Performance Criterion

능한 멀리 떨어져 있는 자세를 취하도록 성능함수를 설정하였다.

따라서, 성능함수(performance criterion) 식은 식 (16)과 같이 구성이 된다.

$$P(\Theta) = l_{1obs} + l_{2obs} + l_{3obs} \quad (16)$$

식 (16)에서

$$\begin{aligned} l_{1obs} &= [(l_1 \cos \theta_1 - O_x)^2 + (l_1 \sin \theta_1 - O_y)^2]^{1/2} \\ l_{2obs} &= [(l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) - O_x)^2 \\ &\quad + (l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) - O_y)^2]^{1/2} \\ l_{3obs} &= [(l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) - O_x)^2 \\ &\quad + (l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) - O_y)^2]^{1/2} \end{aligned} \quad (17)$$

그리고, 위 식들에서  $O_x, O_y$ 는 각각 장애물 중심의 x 성분 및 y 성분이다.

### 2.4 경로제어 알고리즘

Fig.5는 여유자유도를 갖는 로봇 팔의 경로제어 알고리즘에 대한 제어흐름도이다. Fig.5에서 로봇 팔이 계획된

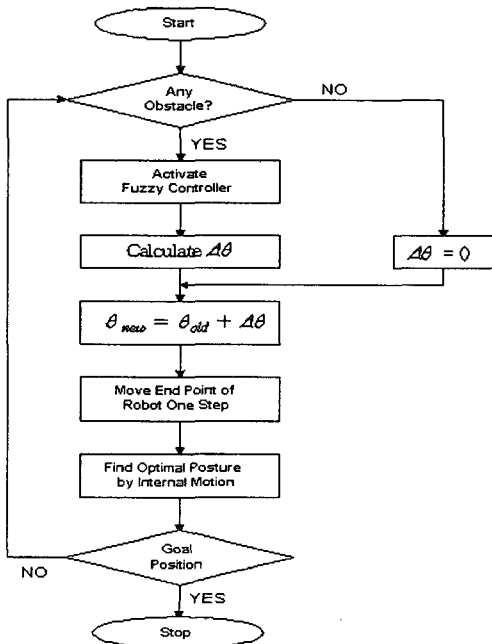


Fig. 5 Motion Control Algorithm for Redundant Robot Arm

경로를 따라 움직일 때 규정된 반경 이내에 장애물이 감지되지 않으면 로봇 팔은 계획된 경로를 수정하지 않고 목표 위치에 도달하게 되지만, 장애물이 감지되면 퍼지제어기(fuzzy controller)가 작동이 되어 계획된 경로에서 방향 수정에 필요한 수정각  $\Delta\theta$ 를 계산하게 된다. 따라서, 로봇 팔의 끝점의 진행 방향이 결정되며 이 방향으로 한 스텝만큼 움직이게 하는 정보를 주게 된다. 그러나, 로봇 팔의 끝점에서 여유자유도 로봇은 수많은 자세(posture)가 존재하므로 주어진 목적에 맞는 성능함수(performance criterion)를 최대화시키는 자세를 구하게 된다.

### 3. 실험 및 고찰

#### 3.1 실험장치의 구성

실험에 사용된 주요 주변장치의 모델명과 사양은 다음의 표 2와 같다.

Table 2. 실험장치의 구성

컴퓨터	제품명	비고
모터	486DX2-50	
	NSK Direct-Drive Motor 1010	Max Torque: 16kgf-m Resolution: 3840count/rev
	NSK Direct-Drive Motor 0408	Max Torque: 1kgf-m Resolution: 25600count/rev
카운트 보드	Harmonic Drive motor HS-250-2-RH-14-6002	Max Torque: 0.33kgf-m Resolution: 40000count/rev
	자체 제작	16bit 5 channel up/down Counter
D/A 보드	Real Time Device ADA-2110	A/D Convert : 16 channel D/A Convert : 2 channel
	Real Time Device DA-720	D/A Convert : 4 channel

#### 3.2 제어시스템의 구성

카운트 보드는 모터의 엔코드에서 입력되는 신호를 디지털 신호로 변환하여 컴퓨터에 입력시키고, 컴퓨터는 입

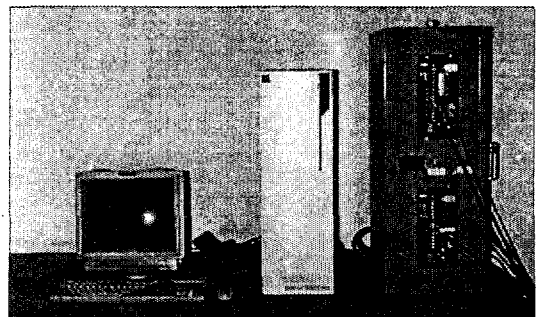


Fig. 6 Photograph of Robot Motion Controller

력된 정보를 이용하여 로봇의 현재 위치를 계산하여 원하는 위치만큼 이동하라는 제어신호를 내보낸다. 이 디지털 신호를 D/A 보드가 아날로그 신호로 변환하여 모터 드라이버에 보내어 로봇을 구동시키게 된다.

### 3.3 여유자유도 로봇의 장애물 우회

3관절 로봇은 여유자유도를 가지기 때문에 내부운동의 최적화 과정이 필요하다. 실험을 위하여 사용된 3관절 로봇 팔의 크기는  $l_1 = 60$ ,  $l_2 = 40$ ,  $l_3 = 25$ 로 선정하였다. 시작 위치에서의 로봇 팔의 끝점의 좌표를  $(ex, ey) = (113, 23)$ 으로 목표 위치에서 로봇 팔의 끝점의 좌표를  $(ex, ey) = (50, 70)$ 으로 선정하였고, 장애물의 형상은 원의 형태로 가정하였으며 장애물의 크기와 위치는 원의 반경과 중심의 위치로 표현하였다. 그리고 장애물의 위치와

로봇팔의 끝점의 간격이 장애물 직경의 5배 이내로 가까워졌을 때 퍼지제어기가 작동하여 로봇 팔의 끝점의 진행 방향을 수정하도록 하였고, 이때부터 퍼지제어와 동시에 로봇 팔의 최적화 자세를 찾는 과정을 수행하였다.

Fig.7은 3관절 로봇의 장애물 우회 모의 실험이다. 그림에서 보듯이 장애물의 위치와 크기에 관계없이 장애물을 성공적으로 우회하여 목표 위치에 도달하였다. 그리고 장애물이 목표 위치 가까이 있을 경우에는 두 번째 링크가 장애물에서 가능한 한 멀어지려는 것을 볼 수 있다. 그러나 실제 실험에 적용한 결과 최적자세를 찾는 과정에서 많은 시간이 소모되어 실시간적으로 제어가 되지 않았다. 그래서 최적화 과정을 생략하고, 대신 3관절과 X축이 이루는 각도  $\phi$ 를 초기궤적에 수직(본 실험에서  $\phi = 50^\circ$ 로 하였다.)하게 하여 장애물 우회실험을 실시하였다.

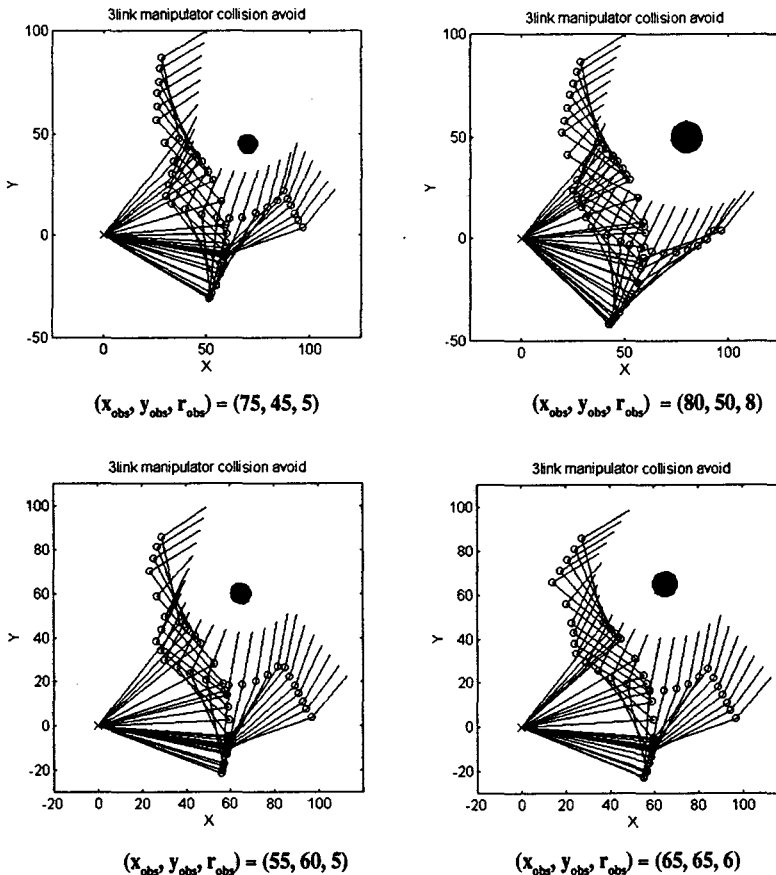


Fig. 7 Collision Avoidance Simulation for 3-Axis Robot Arm

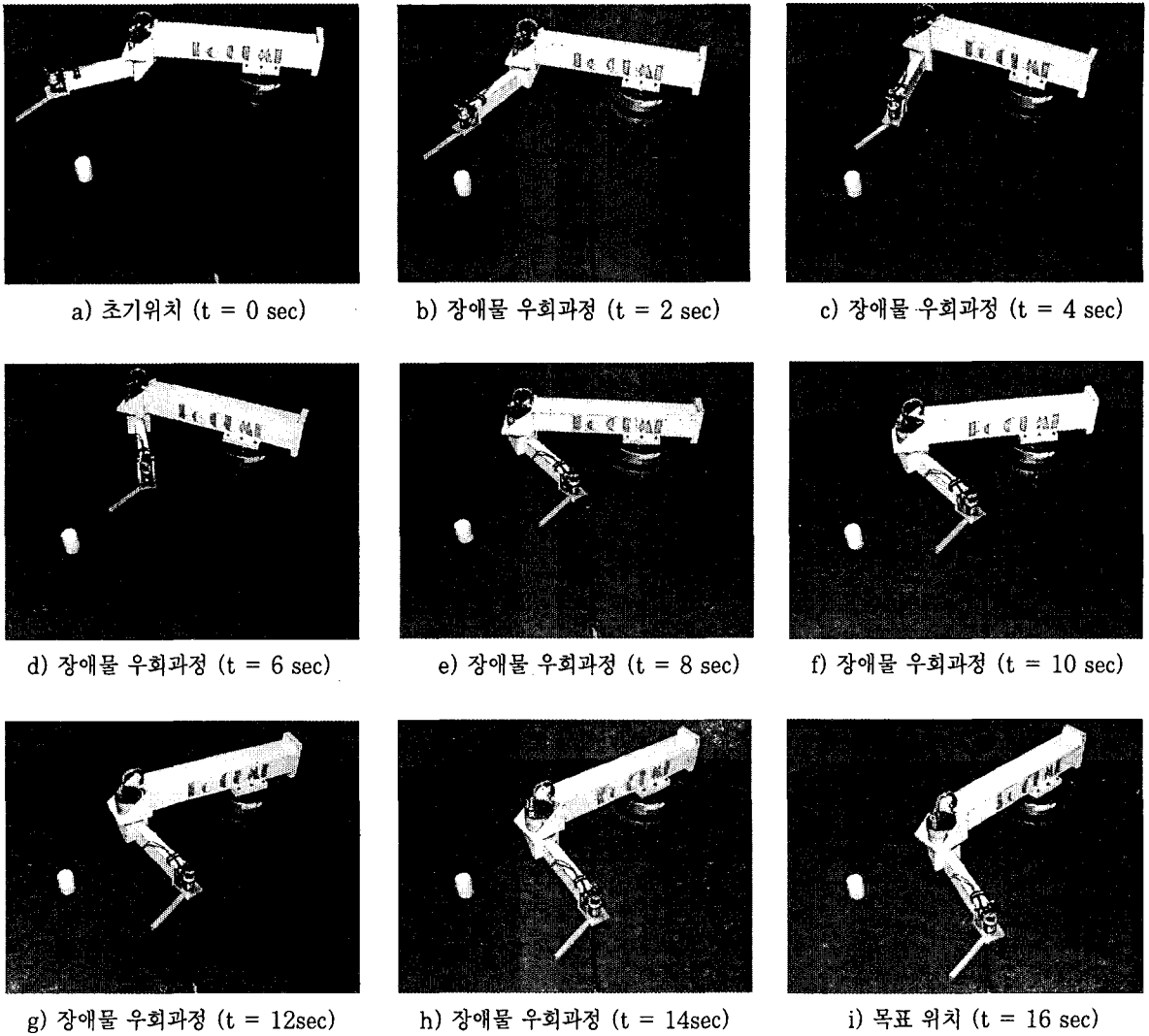


Fig. 8 Collision Avoidance Experiment for 3-Axis Robot Arm

Fig.8은 장애물의 중심과 반경이 (80, 50, 6)이고,  $\phi = 50^\circ$ 인 경우의 실험 결과이다. 모의 실험과 마찬가지로 장애물을 성공적으로 우회함을 알 수 있다.

위의 실험에서는 장애물이 정지하고 있는 경우에 대한 적용 예이다. 그러나, 본 연구에서 개발된 알고리즘은 장애물이 움직이고 있는 경우에도 바로 적용이 가능하다. Fig.10은 장애물이 움직이고 있는 경우의 모의실험 예이다. Fig.10에서 보듯이 로봇은 장애물을 성공적으로 우회하여 목표위치에 도달하였다.

#### 4. 결 론

본 논문에서는 수평 다관절 로봇이 초기 위치에서 목표 위치까지 이동할 때 우연히 발생할 수 있는 장애물을 퍼지 제어 기법을 이용하여 우회할 수 있는 운동 제어 알고리즘을 개발하였다. 작업영역에서 로봇이 움직일 때 어떠한 장애물도 감지되지 않으면 로봇은 미리 주어진 경로를 따라 계획을 수정할 필요 없이 목표 위치에 도달하게 되지만, 장애물이 충돌위험 영역으로 접근하게 되면 퍼지 제어

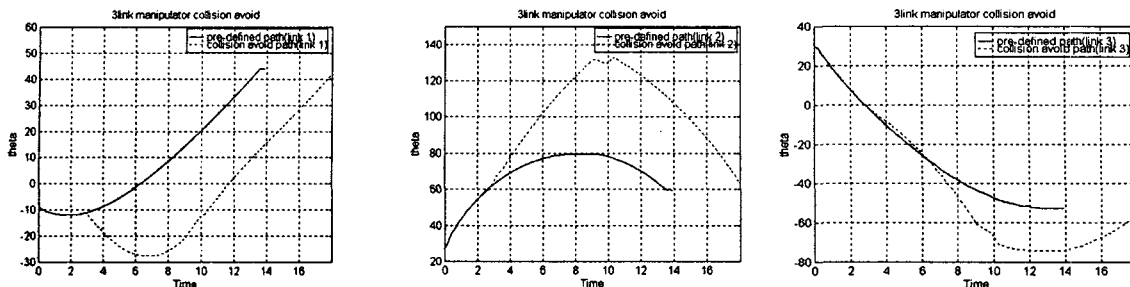


Fig. 9 Comparison of Collision Avoidance Trajectory and Scheduled Trajectory

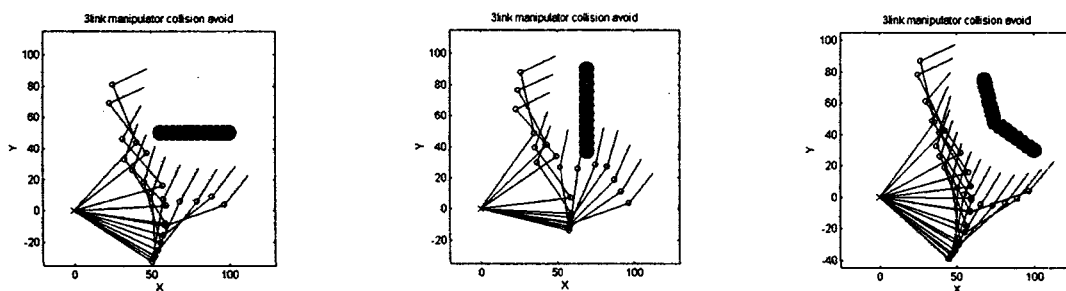


Fig. 10 Collision Avoidance Simulation for a Moving Obstacle

시스템이 작동하게 되어 로봇의 끝점의 이동 방향을 수정하게 하여 주어진 궤적을 이탈하게 하고 장애물을 우회하여 목표 위치에 도달하게 한다.

개발된 운동제어 알고리즘을 수평면을 움직이는 3축 로봇에 대하여 적용하여 보았다. 장애물의 크기와 위치를 변경하여 로봇이 장애물을 성공적으로 우회하는지 조사하였으며 만족스러운 결과를 얻었다. 3축 로봇은 여유자유도를 갖는 로봇이므로 주어진 로봇의 끝점에 대하여 많은 자세가 생기는데, 최적화 알고리즘을 이용하여 최적의 자세를 찾도록 하였다. 여유자유도를 갖는 로봇은 최적화 과정에서 계산 시간이 많이 걸리는 단점이 있지만 실험 결과에서 알 수 있듯이 장애물을 우회하는 순간에 장애물과 로봇 사이의 거리를 충분히 유지하는 자세를 취할 수 있으므로, 원활한 장애물 우회작업을 수행할 수 있음을 알 수 있다. 그리고, 성능함수를 목적에 맞게 변경함으로써 다양한 형태의 이동 자세를 얻을 수 있는 장점이 있다. 본 논문에서는 장애물의 크기 및 위치를 알고리즘의 실행 전에 임의로 선정하였으나, 비전 센서(vision sensor)나 거리 센서 등을 이용하여 장애물의 크기나 위치를 실시간적으로 계산이 가능하므로 실제 작업에 쉽게 적용이 가능하다고 본다.

### 참고 문헌

1. T.Lozaano-Perez, "Spatial Planning: A Configuration Space Approach," IEEE Trans. Comp., pp. 108-120, 1983.
2. T.Lozaano-Perez, "A Simple Motion-Planning Algorithm for General Robot Manipulators," IEEE Journal of Robotics and Automation, Vol. 3, No. 3, pp. 224-238, 1987.
3. O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," Int. Journal of Robotics Research, Vol. 5, No. 1, pp. 90-98, 1986.
4. J.Barraguang and J. C. Latombe, "Robot Motion Planning: A Distributed Representation Approach," Int. Journal of Robotics Research, Vol. 10, No. 6, pp. 628-649, 1991.
5. D. G. Luenberg, "Linear and Nonlinear Programming," Addison Wesley, pp. 168-169.



- 1984.
6. G. V. S. Raju and J. Zhou, "Fuzzy Rule Based Approach for Robot Motion Control," IEEE Int. Conf. on Fuzzy Systems, pp. 1349-1356, 1992.
  7. A. Nedungadi, "A Fuzzy Robot Controller-Hardware Implementation," IEEE Int. Conf. on Fuzzy Systems, pp. 1325-1331, 1992.
  8. N. Chen and H.Chung, "Robot Path Plan-  
ner: A Neural Networks Approach," IEEE/RSJ Int. conf. on Intelligent Robots and Systems, Vol. 1, pp. 548-553, 1992.
  9. A. Bagachi and H.Hatwal, "A Solution Strategy for Collision Avoidance of Multiple Bodies Moving on a Plane Using Fuzzy Logic," Proc. Int. Sys. Intelligent Robotics, Jan. 1991.