공학 연구논문집 제28권 제2호 pp. 177~191, 울산대학교.
Journal of Engineering Research Vol. 28, No. 2, pp. 177~191, 1997
University of Ulsan.

여유자유도를 갖는 로봇팔의 장애물 우회* 알고리즘 개발

이병룡 · 신현배
기계공학과

<요 약>

본 논문에서는 로봇팔이 임의의 시작위치에서 목표위치까지 움직일 때 돌발적으로 발생하는 장애물을 퍼지제어를 이용하여 장애물을 실시간적으로 우회할 수 있는 운동제어 알고리즘을 개발하였다. 로봇팔의 이동중에 장애물이 존재하지 않으면 로봇팔의 끝점은 미리 규정된 경로를 따라 움직이며, 만약에 장애물이 일정 영역내에서 감지되면 퍼지제어기가 작동하여 규정된 경로를 이탈하여 충돌을 방지하면서 목표점까지 새로운 경로를 발생하게 된다. 본 논문에서는 개발된 알고리즘의 유용성을 입증하기 위하여 여유자유도로봇과 비여유자유도로봇에 개발된 알고리즘을 적용하여 전산모의실험을 하였다.

Development of a Collision Avoidance Algorithm for Redundant Robot Arms

Byung-Ryong Lee · Hyun-Bae Shin
Dept. of Mechanical Engineering

<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 goal posture. During the motion, if there exists no obstacles the robot arm moves along the pre-defined path. But if some obstacles are recognized

* 본 연구는 울산대학교 학술연구조성비에 의하여 연구되었음.

and close to the robot arm, fuzzy controller is activated to adjust the path of the robot arm. To show the feasibility of the developed algorithm, a simulation is carried out. In the simulation, non-redundant and redundant planar robot arms are considered for the collision avoidance test, and it was proved that the developed algorithm gives good collision-avoiding performance both the stationary obstacles and moving obstacles.

Key Words: Redundant Robot, Fuzzy Logic, Collision Avoidance, Posture Optimization

1. 서 론

로봇이 작업 영역에서 이동 중일 때 발생되는 장애물을 우회하는데 사용되는 경로계획법은 크게 2가지로 구분할 수 있는데, 한 방법은 상태공간(configuration space)법이며 다른 방법은 인공전위장(artificial potential field)법이다. 상태공간법[1][2]은 로봇팔의 관절(joint)의 운동이 n-차원(n-dimension)의 상태공간에서의 점의 운동으로 변환이 되며, 장애물 또한 상태공간에서 장애물영역으로 치환(mapping)된다. 따라서, 상태공간으로부터 로봇과 장애물사이의 충돌이 발생하지 않는 안전한 경로를 구할 수 있게 되며, 이 경로로부터 각 관절의 회전각을 산출하는 방법이다. 이 방법은 계산 시간이 많이 들며, 장애물의 위치를 사전에 알고 있어야 하는 문제점을 안고 있다. 인공전위장법[3][4]은 인공적인 전위장함수를 이용하는 방식인데 장애물이 있는 위치는 전위장값을 높게 할당하며, 로봇이 도달해야 하는 목표 위치는 전위장값을 최소 값으로 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 지역최소점(local minima)[5]에 빠지는 경우가 발생할 수도 있는 문제점을 안고 있다.

최근에는 퍼지 및 신경회로망 기법이 로봇의 궤적추적 또는 장애물우회를 위한 경로 계획에 많이 적용되고 있다. 퍼지 및 신경회로망 기법은 앞에서 언급한 상태공간법이나 인공전위장법에 비하여 시스템을 해석하기 위한 모델링이 간단해 지며, 경로탐색 알고리즘을 작성하는데 있어서 복잡한 기구학 및 동역학적 계산식을 고려할 필요가 없으며 단지 경험을 가진 교사(teacher)의 경험에 의해 학습을 시킬 수 있다는 점이 장점이다. Raju[6], Nedungadi[7] 등은 퍼지규칙을 이용하여 관절형 로봇을 역기구학을 계산하지 않고 임의의 초기위치에서 목표위치까지의 경로를 단순히 추종하도록 하는 연구를 수행하였다. Chen[8] 등은 로봇의 작업영역에서 장애물들이 존재할 때 장애물 때문에 로봇이 접근할 수 없는 영역을 신경회로망을 이용하여 학습을 시켜 로봇이 시작위치에서 목표위치까지 장애물을 우회하여 도달할 수 있도록 하였다. 그러나, 이 경우 돌발적으로 발생할 수 있는 미지의 장애물들과의 충돌을 예방할 수 없는 문제점이 있다. Bagachi[9] 등은 퍼지로직을 이용하여 여유자유도로봇(redundant robot)을 임의의 시작위치에서 목표위치까지 이동하면서 이동중에 발생하는 장애물과의 충돌을 예방할 수 있음을 보였다. 그러나, 이 논문에서는 로봇팔의 끝점(end-effector)의 위치가 다음위치로 이동할 시점에서 장애물과 로봇팔과의 충돌가능성을 최소화하기 위한 최적화(optimization) 문제는 다루지 않았다.

본 논문에서는 로봇의 이동 중에 일정한 영역내에서 장애물이 감지되면 퍼지로직을 이

용하여 실시간 적으로 장애물을 우회하도록 하며, 또 우회하는 도중에 로봇과 장애물과의 충돌 위험을 최소화하는 최적자세(optimal posture)를 유지하면서 목표 위치까지 로봇을 이동시킬 수 있는 최적화경로계획 알고리즘을 개발하였다. 그리고, 개발된 알고리즘을 비여유자유도 (non-redundant)로봇과 여유자유도(redundant)로봇에 적용하여 개발된 알고리즘이 돌발적으로 발생하는 미지의 정지장애물 및 운동장애물을 우회하여 목표위치까지 도달할 수 있음을 전산모의 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. 여유자유도 로봇 시스템에서는 로봇팔의 끝위치가 주어질 때 끝위치를 만족하는 로봇팔의 자세는 무수히 많이 존재하게 된다. 따라서, 이러한 여유자유도를 이용하여 원하는 로봇의 작동요소를 최적화할 수 있다는 장점이 있다. 예를 들어 로봇팔이 소모하는 에너지를 최소화시키거나, 특이점(singular point)을 회피하거나, 원하는 궤적을 따라서 속도 및 힘의 전달율(transmission ratio)을 최대화 할 수 있는 등의 작업을 수행할 수 있다. 본 연구에서는 여유자유도를 가진 로봇이 움직일 때 돌발적으로 발생하는 장애물을 최적의 자세를 유지하면서 우회할 수 있는 장애물우회 알고리즘을 개발하고자 한다.

2.1 퍼지로직을 이용한 장애물 우회 계획

퍼지로직은 입력변수와 출력변수로 구성이 되며 각 변수는 여러개의 퍼지 집합(fuzzy set)으로 표현된다. 입력변수와 출력변수의 퍼지집합으로부터 IF-THEN 규칙을 만들어 내며 이 규칙을 이용하여 제어출력값을 유도하게 된다. 이러한 규칙이 만들어진 후에 임의의 입력변수값이 들어오면 퍼지로직은 입력변수값에 대응하는 출력값을 제공하게 된다. 본 연구에서는 퍼지로직을 이용하여 로봇이 장애물을 실시간(real-time)적으로 우회할 수 있는 퍼지 알고리즘을 유도하고자 한다. Fig.1은 로봇이 움직일 때 돌발적으로 발생한 장애물이 로봇팔 끝점의 진행방향 가까이 놓여져 있는 모습을 보여주고 있다. 퍼지로직을 이용하기 위하여 필요한 입력변수는 Gap과 Length로 선정하였다. Gap은 로봇팔의 끝점(end effector)이 진행하고 있는 방향의 선상으로부터 장애물의 법선방향으로 떨어진 거리로 정의되고, Length는 장애물이 로봇팔의 끝점에서 로봇팔의 진행방향으로 떨어져 있는 거리로 정의된다. Fig.1에서 θ_{goal}, θ_{obs} 그리고 Len_{obs} 는 아래와 같이 표시된다.

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

$$\theta_{obs} = \tan^{-1}\left(\frac{y_{obs} - y}{x_{obs} - x}\right) \tag{2}$$

$$Len_{obs} = \sqrt{(x_{obs} - x)^2 + (y_{obs} - y)^2} \tag{3}$$

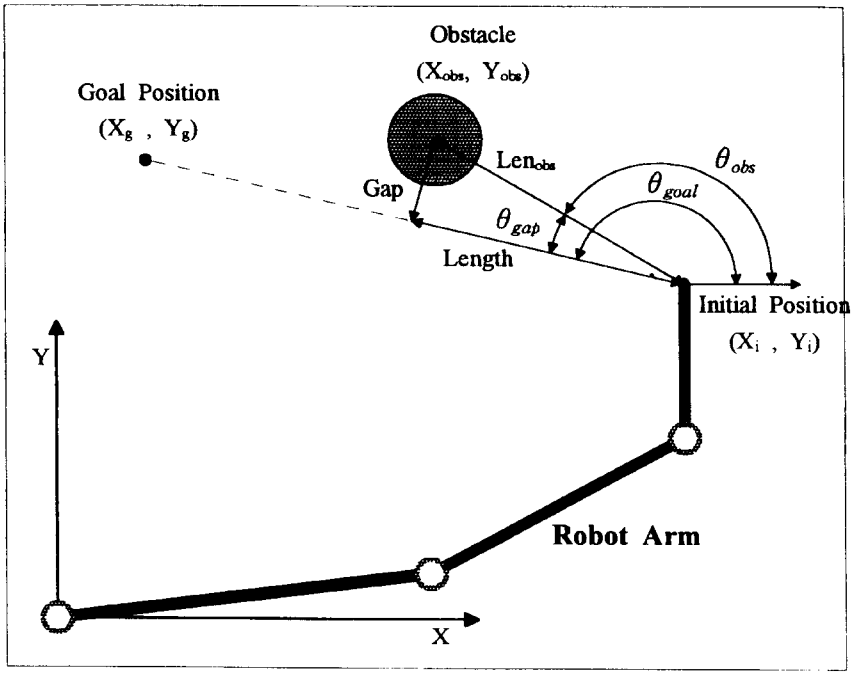

Fig.1 Configuration of Robot Arm and Obstacle

θ_{gap} 을 다음과 같이 정의하면

$$\theta_{gap} = \theta_{goal} - \theta_{obs} \tag{4}$$

퍼지제어기의 입력변수 Gap과 Length는 다음과 같은 식으로 표시된다.

$$Gap = Len_{obs} \cdot \sin\theta_{gap} \tag{5}$$

$$Length = Len_{obs} \cdot \cos\theta_{gap} \tag{6}$$

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

입력 및 출력변수에 대한 퍼지구성함수(fuzzy membership function)와 퍼지 세트는 Fig.2에 나타나 있다. Fig.2에서 입력변수 Gap은 NB(Negative Big), NS(Negative Small), ZE(Zero), PS(Positive Small) 그리고 PB(Positive Big)으로 5개의 퍼지세트로 구성되어 있으며, 입력변수 Length는 NE(Negative), ZE(Zero), PS(Positive Small) 그리고 PB(Positive Big)로 4개의 퍼지세트로 구성되어 있다. 마찬가지로 출력변수 $\Delta\theta$ 는 ZE(Zero), PS(Positive Small), PM(positive Medium), PB(Positive Big) 그리고 PVB(Positive Very Big)로 5개의 퍼지세트로 구성되어 있다. 입력 및 출력변수의 퍼지세트에 대한 퍼지제어규칙인 FAM(Fuzzy Associative Memory)은 Fig.3에 나타나 있다. FAM이란 입력변수와

출력변수의 퍼지세트간의 논리적으로 타당한 IF-THEN 규칙을 함축적으로 나열한 것이라고 볼 수 있다. 예를들어 FAM의 좌측상단의 블록을 달리 표현하면 다음과 같은 논리규칙으로 표시된다.

IF LENGTH is NE and GAP is NB, THEN $\Delta\theta$ is PVB.

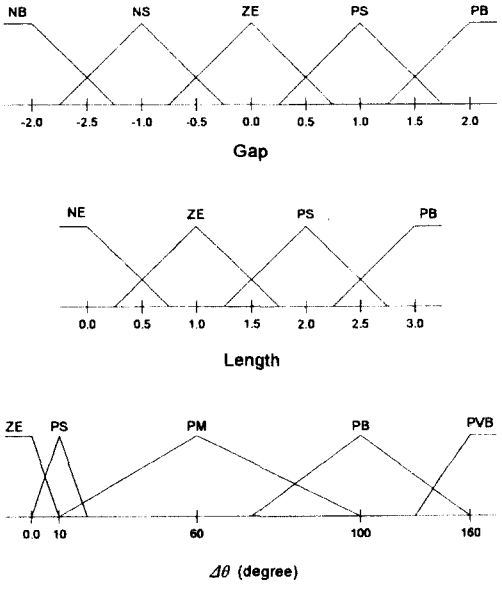

Fig.2 Fuzzy Membership Functions of Input and Output Variables

Gap \ Length	NE	ZE	PS	PB
NB	PVB	PVB	PVB	PB
NS	PVB	PVB	PB	PM
ZE	PVB	PB	PM	PS
PS	PM	PM	PS	ZE
PB	PS	PS	ZE	ZE

Fig.3 Fuzzy Associative Memory of Input and Output Variables

그리고, 퍼지제어기는 FAM을 이용하여 임의의 입력변수가 입력되었을 때 비퍼지화(defuzzification method) 방법을 이용하여 명확한 출력값을 제공하게 된다. 비퍼지화방법에는 여러 가지 방법이 있으나 본 논문에서는 일반적으로 많이 쓰이고 있는 무게중심(center of area)[10]법을 사용하였다.

2.2 자세 최적화 (posture optimization)

여유자유도를 갖는 로봇팔의 속도는 높은 차원의 관절공간(joint space)에서 낮은 차원의 작업공간(task space)으로 식(7)과 같이 치환(mapping)이 된다.

$$\dot{X} = J(\Theta)\dot{\Theta} \tag{7}$$

여기서, $X = [x_1, x_2, \ldots, x_m]^T$ 는 작업좌표 벡터이고, $\Theta = [\theta_1, \theta_2, \ldots, \theta_n]^T$ 는 관절좌표 벡터이며, J는 $m \times n$ Jacobian matrix이다. 식(7)에 대한 역기구학식은 식(8)과 같이 구해진다.

$$\dot{\Theta} = J^{+}\dot{X} + (I - J^{+}J)K \tag{8}$$

식(8)에서 J^{+} 는 Jacobian matrix의 pseudo-inverse이며 아래의 식과 같이 정의된다.

$$J^{+} = J^{T}(JJ^{T})^{-1} \tag{9}$$

그리고, 식(8)에서 K 는 임의의 벡터이다. 식(8)에서 $(I - J^{+}J)K$ 값은 로봇팔의 내부운동 (internal motion)을 설명해 주는 항목이다. 즉, 로봇팔의 끝점이 고정되어 있다 하더라도 여유자유도를 갖는 로봇팔은 무수히 많은 자세를 가지게 되는데, 바로 이 현상을 설명해 주는 식이다. 식(8)의 내부운동 중에서 로봇팔이 원하는 성능함수(performance criterion)를 최대화시켜 주도록 벡터 K 의 값을 결정하여야 한다. 성능함수를 $P(\Theta)$ 라고 했을 때, $P(\Theta)$ 를 최대화하기 위한

벡터 K 의 관계는 식(10)와 같이 구해진다.

$$K = \frac{\partial P}{\partial \Theta}x \tag{10}$$

식(10)에서 $\partial P/\partial \Theta = [\partial P/\partial \theta_1, \partial P/\partial \theta_2, \ldots, \partial P/\partial \theta_n]^T$ 그리고 x 는 임의의 양의 상수이다. 따라서, 식(10)의 관계가 유지되게 하는 내부운동(internal motion)은 항상 $P(\Theta)$ 를 증가시키는 방향으로 움직인다[11]는 것을 의미한다.

2.3 경로제어 알고리즘

일반적으로 여유자유도를 갖는 로봇팔의 경우 경로제어 알고리즘의 전체적인 구성은 비

여유자유도의 경로제어 알고리즘과 거의 같지만, 매 샘플링마다 성능함수(performance criterion) 를 최대화시켜주는 자세를 찾기 위한 내부운동이 추가된다는 점이 다르다. Fig.4는 여유자유도를 갖는 로봇팔의 경로제어 알고리즘에 대한 제어흐름도이다. Fig.4에서 로봇팔이 계획된 경로를 따라 움직일 때 규정된 반경 이내에 장애물이 감지되지 않으면 로봇팔은 계획된 경로를 수정하지 않고 목표 위치에 도달하게 되지만, 장애물이 감지되면 퍼지 콘트롤라(fuzzy controller)가 작동이 되어 계획된 경로에서 방향 수정에 필요한 수정각 $\Delta\theta$ 를 계산하게 된다. 따라서, 로봇팔 끝점의 진행 방향이 결정되며 이 방향으로 한 스텝만큼 움직이게 하는 정보를 주게 된다. 그러나, 로봇팔 끝점에서 여유자유도 로봇은 수많은 자세(posture)가 존재하므로 주어진 목적에 맞는 성능함수(performance criterion)를 최대화시키는 자세를 구하게 된다.

본 연구에서는 로봇팔의 이동중에 발생될 수 있는 장애물을 충돌하지 않고 안전하게 우회하여 목적지에 도달하는 것이 중요한 요소이므로, 장애물이 존재하고 로봇 끝점의 위치가 결정된 경우에 로봇팔의 각각의 링크(link)가 Fig.5에서 보는 바와 같이 장애물로부터 가능한 멀리 떨어져 있는 자세를 취하는 것이 가장 이상적인 자세가 된다.

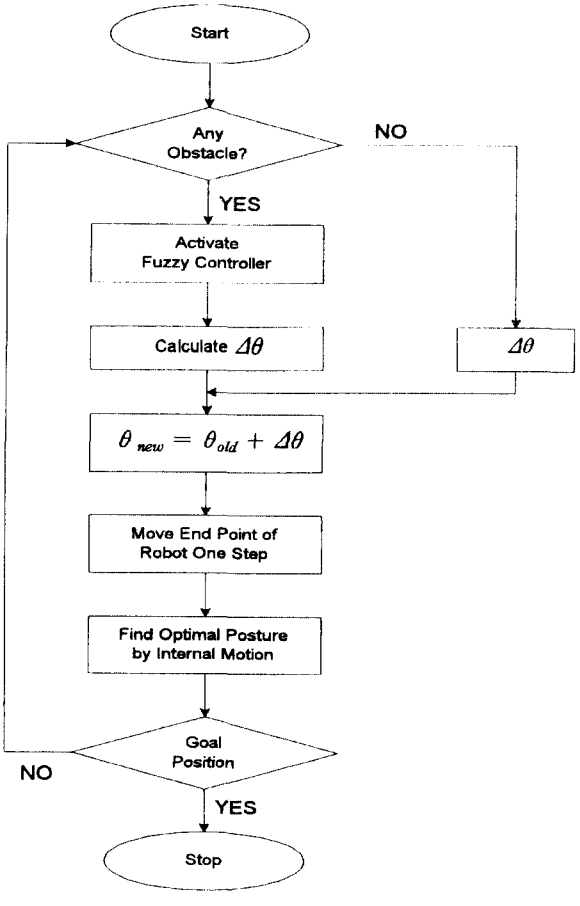

Fig.4 Path Control Algorithm of Redundant Robot

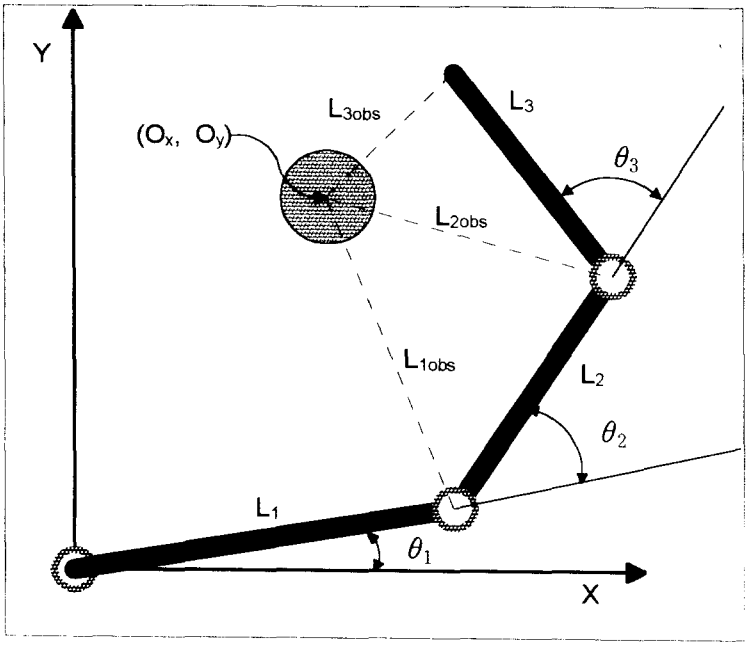

Fig.5 Parameters of Performance Criterion

따라서, 성능함수(performance criterion) 식은 식(11)과 같이 구성이 된다.

$$P(\Theta) = L_{1obs} + L_{2obs} + L_{3obs} \tag{11}$$

식 (11)에서

$$L_{1obs} = [(L_1 \cos\theta_1 - O_x)^2 + (L_1 \sin\theta_1 - O_y)^2]^{1/2} \tag{12}$$

$$L_{2obs} = [(L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) - O_x)^2 + (L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) - O_y)^2]^{1/2} \tag{13}$$

$$L_{2obs} = [(L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) - O_x)^2 + (L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) - O_y)^2]^{1/2} \tag{14}$$

그리고, 위 식들에서 O_x, O_y 는 각각 장애물 중심의 x성분 및 y성분이다.

3. 전산 모의 실험

3.1 여유자유도 로봇의 내부운동의 최적화

내부운동중의 최적화 자세를 찾는 과정을 보여주기 위하여 Fig.6과 같이 장애물과 로봇팔이 근접하여 있는 경우를 고려하였다. 그림에서, 로봇의 끝점 위치는 (6.0,8.0)이고, 로봇팔의 치수는 L1=8.0, L2=4.0, L3=3.0이다. 장애물 중심의 위치는 (5.0,7.0)이며 장애물의 반경은 0.5로 설정하였다. Fig.6과 같은 초기 자세에서 최적화 알고리즘 식(10)과 성능함수인 식(11)을 이용하여 내부운동중에 성능함수를 최대화시키는 자세를 Fig.7과 같이 구하였다. Fig.7에서 알 수 있듯이 스텝 수가 증가할수록 로봇의 자세는 장애물로 가능한 한 멀어지려고 하는 것을 알 수가 있다.

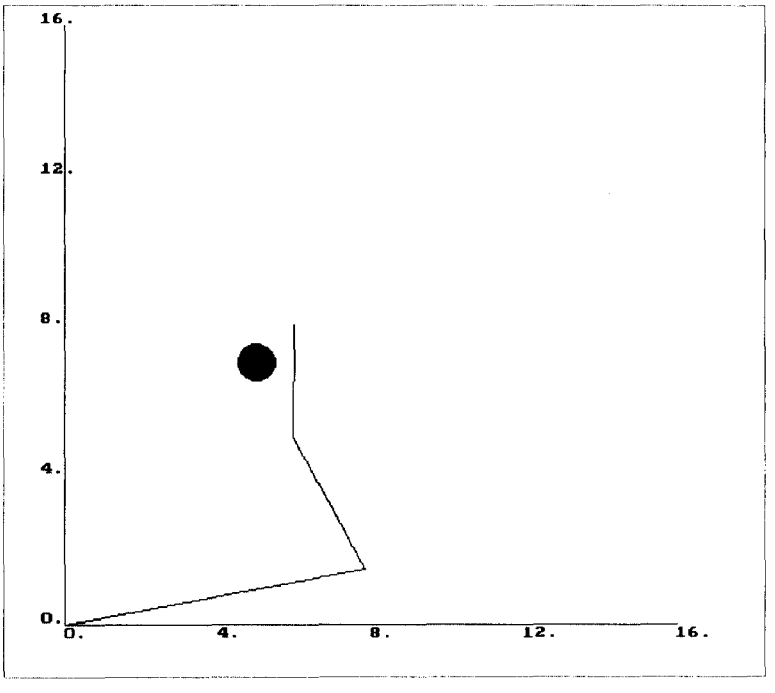

Fig.6 Initial Posture of Internal Motion Optimization

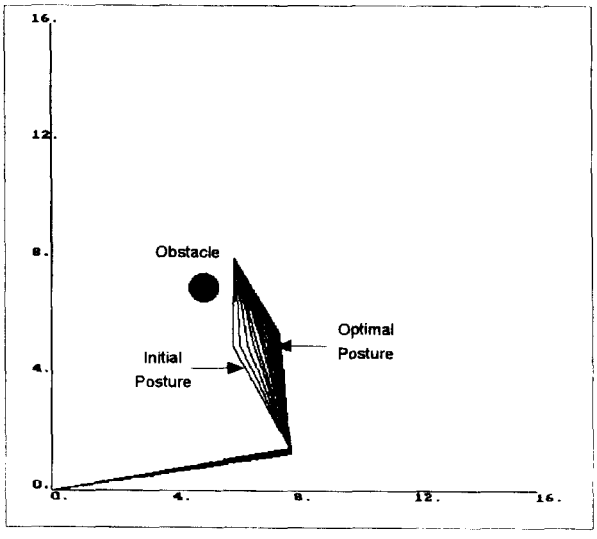

Fig.7 Result of Internal Motion Optimization

3.2 여유자유도 로봇의 장애물 우회

모의실험을 위하여 사용된 3관절 여유자유도 로봇팔의 크기는 L1=8.0, L2=4.0, L3=3.0으로 선정하였으며, 여유자유도 로봇팔과 성능을 비교하기 위하여 2관절 비여유자유도 로봇팔의 크기는 L1=10.0, L2=5.0으로 선정하였다. 두 경우에 있어서 로봇팔 전체의 길이는 같도록 하였고 시작 위치에서의 로봇팔의 끝점 좌표는 $(e_x, e_y)=(12.0, 8.0)$이며, 3관절 로봇팔의 경우 3번째 링크가 좌표축과 이루는 각도는 60o이다. 그리고 목표 위치에서의 끝점 좌표는 $(e_x, e_y)=(2.0,11.0)$으로 선정하였다. 장애물의 형상은 원(circle)의 형태로 가정하였으며 장애물의 크기와 위치는 장애물의 반지름과 장애물 중심의 위치에 의해서 표현하였다. 로봇팔이 이동중에 장애물의 위치와 로봇팔 끝점의 간격이 장애물 직경의 5배 이내로 가까워 졌을 때 퍼지 콘트롤러가 작동하여 로봇팔 끝점의 진행 방향을 수정하도록 하였다. 그리고, 장애물은 로봇팔의 첫 번째 링크의 회전반경 밖에 존재한다고 가정하였다. 시뮬레이션 결과는 Fig.8에서 Fig.12에 걸쳐 나타나 있다. Fig.8과 Fig.9는 장애물의 중심위치가 (10.0, 7.0)이며 장애물의 반경이 1.0인 경우 각각 2관절 로봇팔과 3관절 로봇팔에 대한 장애물 우회 결과를 보여준다. 이 경우 2관절 및 3관절 로봇 모두 장애물을 성공적으로 우회하여 목표점에 도달하는 것을 알 수 있다. Fig.10은 장애물 중심의 위치가 (6.0, 8.0)이며 장애물의 반경이 0.5인 경우의 2관절 로봇의 결과이다. 그림에서 보는 바와 같이 로봇의 끝점은 장애물을 우회하였지만 두 번째 관절이 장애물에 충돌하는 것을 볼 수 있다. 그러나, 여유자유도를 가지고 있는 3관절 로봇은 장애물 중심의 위치가 (6.0, 8.0)에 위치한 경우에도 로봇팔이 장애물을 성공적으로 우회하였다. 이 경우에 대한 결과는 Fig.11에 나타나 있다. 그림에서 보는 바와 같이 3관절 로봇팔의 경우는 로봇이 장애물을 우회하는 동안 로봇팔이 장애물과 가능한 한 멀어지려고 하는 최적화 자세(optimal posture)를 유지하고 있다. 따라서, 여유자유도 로봇팔의 경우가 비여유자유도 로봇팔보다 안전하고 유연한 작

업을 수행할 수 있음을 알 수 있다. 위의 모의실험에서는 장애물이 정지하고 있는 경우에 대한 적용 예이다. 그러나, 본 연구에서 개발된 알고리즘은 장애물이 움직이고 있는 경우에도 바로 적용이 가능하다. Fig.12는 이동장애물에 대한 장애물 우회 결과이다. 장애물이 위치 (8.0, 11.0)에서 로봇의 끝점의 이동속도와 비슷한 속도로 아랫방향으로 움직이고 있으며 위치 (8.0, 8.0) 부근에서 로봇팔의 끝점과 만나게 되지만 로봇이 성공적으로 우회하는 것을 볼 수 있다.

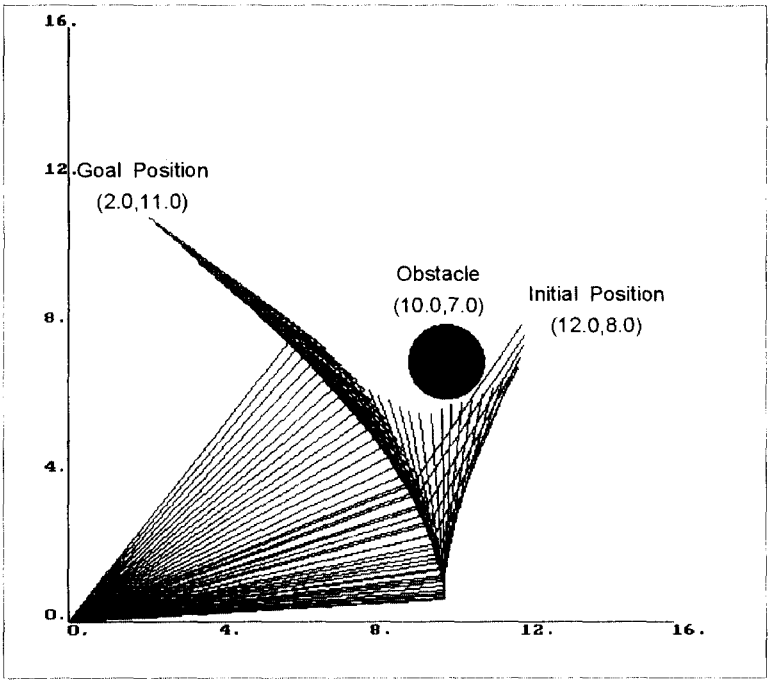

Fig.8 Collision Avoidance Path of 2-Link Robot Arm
: (x_obs,y_obs,r_obs)=(10.0, 7.0, 1.0)

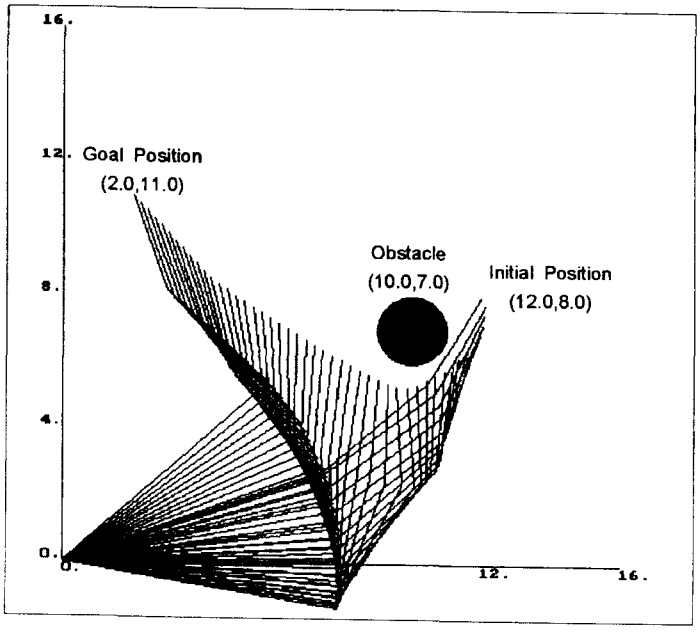

Fig.9 Collision Avoidance Path of 3-Link Robot Arm
: (x_obs,y_obs,r_obs)=(10.0, 7.0, 1.0)

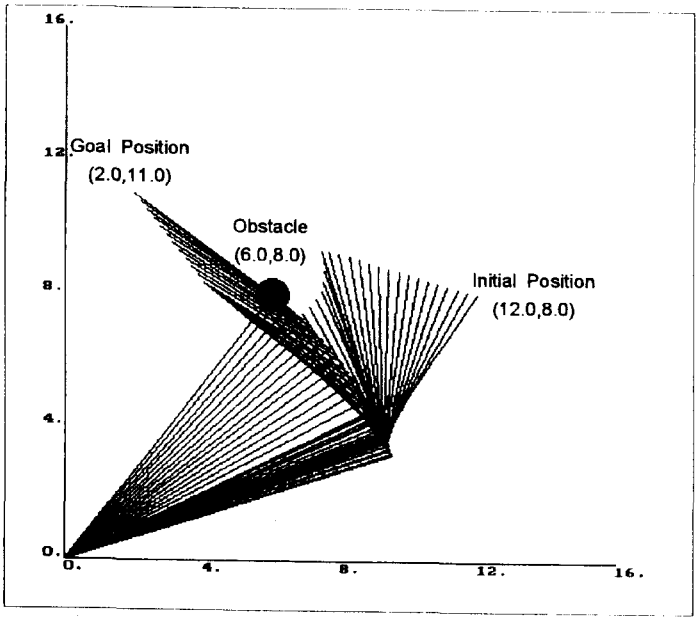

Fig.10 Collision Avoidance Path of 2-Link Robot Arm
: (x_obs,y_obs,r_obs)=(6.0, 8.0, 0.5)

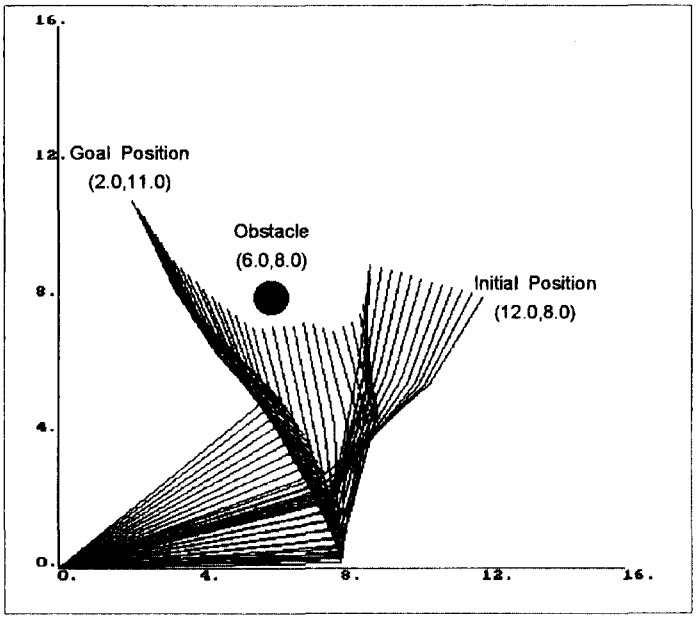

Fig.11 Collision Avoidance Path of 3-Link Robot Arm
: (x_obs,y_obs,r_obs)=(6.0, 8.0, 0.5)

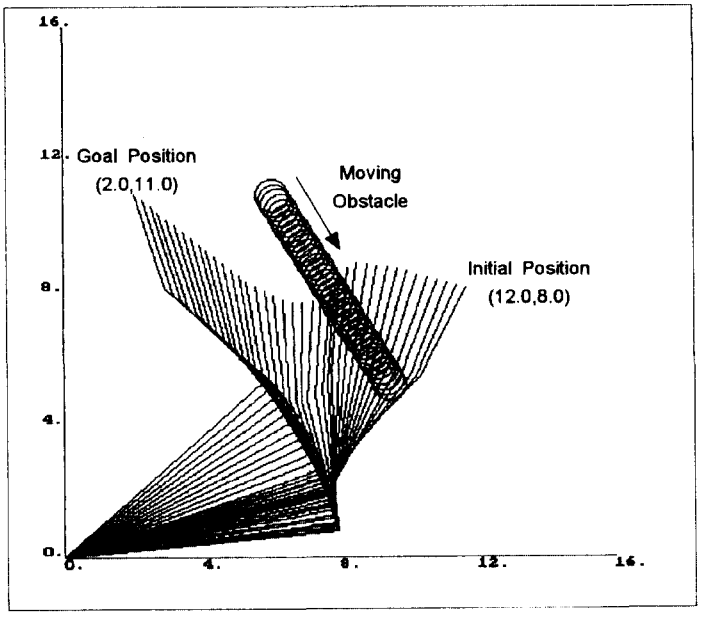

Fig.12 Collision Avoidance Path of 3-Link Robot Arm for Moving Obstacle

4. 결 론

본 논문에서는 수평다관절 로봇의 초기 위치에서 목표 위치까지 로봇이 이동할 때 우연히 발생할 수 있는 장애물을 퍼지제어 기법을 이용하여 장애물을 우회할 수 있는 운동제어 알고리즘을 개발하였다. 작업영역에서 로봇이 움직일 때 어떠한 장애물도 감지되지 않으면 로봇은 미리 주어진 경로를 따라 궤적을 수정할 필요 없이 목표 위치에 도달하게 되지만, 장애물이 충돌위험 영역으로 접근하게 되면 퍼지제어시스템이 작동하게 되어 로봇의 끝점의 이동 방향을 수정하게 하여 주어진 궤적을 이탈하게 하고 장애물을 우회하여 목표점에 도달하게 한다. 개발된 운동제어 알고리즘은 수평면을 움직이는 2축 로봇과 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 자세(posture)가 생기므로 최적화 알고리즘을 이용하여 최적의 자세를 찾도록 하였다. 3축 로봇은 2축 로봇보다 계산시간이 많이 걸리는 단점이 있지만 같은 조건, 즉 같은 출발위치와 같은 목표위치 그리고 로봇팔의 전체길이를 같게 하였을 때, 시뮬레이션 결과에서 알 수 있듯이 3축 로봇 경우는 장애물을 우회하는 순간에 장애물과 로봇 사이의 거리를 충분히 유지하는 자세를 취할 수 있으므로, 2축 로봇으로는 불가능한 장애물 우회작업도 3축로봇의 경우에서는 원활한 장애물 우회작업을 수행할 수 있음을 알 수 있다. 그리고, 성능함수(performance criterion)를 목적에 맞게 변경함으로서 다양한 형태의 이동 자세를 얻을 수 있는 장점이 있다. 본 논문에서는 장애물의 크기 및 위치를 알고리즘의 실행 전에 입력변수로 사용하였으나, vision sensor나 거리 sensor 등을 이용하여 장애물의 크기나 위치를 실시간(real-time)적으로 계산이 가능하므로 실제 작업에 쉽게 적용이 가능하다고 본다. 따라서, 향후의 연구에서는 이러한 위치 및 크기에 대한 정보를 실시간적으로 획득하여 실제 작업에서 장애물 우회에 대한 연구를 수행하고자 한다.

참 고 문 헌

1. Lozaano-Perez,T., "Spatial Planning: a Configuration Space Approach", IEEE TRans. Comp, pp.108-120,1983.
2. Lozaano-Perez,T., "A Simple Motion-Planning Algorithm for General Robot Manipulators," IEEE Journal of Robotics and Automation, Vol.3, No.3, pp.224-238, 1987.
3. Khatib,O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", Int. J. of Robotic Research, Vol,5, No.1, pp.90-98, 1986
4. Barraguang,J. and Latombe,J.C., "Robot Motion Planning: A Distributed Representation Approach," Int. Journal of Robotics Research, Vol.10, No.6, pp.628-649, 1991.
5. Luenberg,D.G., "Linear and Nonlinear Programming," Addison Wesley, pp.168-169, 1984.

6. Raju,G.V.S. and Zhou,J., "Fuzzy Rule Based Approach for Robot Motion Control", IEEE Int. Conf. on Fuzzy Systems, pp.1349-1356, 1992.
7. Nedungadi,A., " A Fuzzy Robot Controller-Hardware Implementation," IEEE Int. Conf. on Fuzzy Systems, pp.1325-1331, 1992.
8. Chen,N. and Chung,H., "Robot Path Planner: A Neural Networks Approach, " IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vol.1, pp.548-553, 1992.
9. Bagachi,A. and Hatwal,H., "A Solution Strategy for Collision Avoidance of Multiple Bodies Moving on a Plane Using Fuzzy Logic," Proc. Int. Sym. Intelligent Robotics, Jan., 1991.
10. Wang,L.X., "Adaptive Fuzzy Systems and Control," Prentice-Hall, pp.22-23, 1994.
11. Yoshikawa,T., "Analysis and Control of Robot Manipulator with Redundancy," In Robotics Research:the first Int. Sym.,eds.Brady,M. and Paul,R., MIT Press, pp.735-747, 1984.