

# 퍼지제어와 성능함수 최적화를 이용한 여유자유도 로봇 팔의 장애물 우회 알고리즘

이병룡\*, 황재석\*\*, 박찬호\*\*, 양순용\*, 안경관\*

## An Obstacle-Avoidance Algorithm for a Redundant Robot Arm Using Fuzzy Control and Performance-Function Optimization

Byung-Ryong Lee\*, Jae-Suk Hwang\*\*, Chan-Ho Park\*\*, Soon-Yong Yang\*, and Kyung-Kwan Ahn\*

### ABSTRACT

In this paper, a motion control algorithm is developed using a fuzzy control and the optimization of performance function, which makes a robot arm avoid an unexpected obstacle when the end-effector of the robot arm is moving to the goal position. During the motion, if there exists no obstacle, the end-effector of the robot arm moves along the pre-defined path. But if there exists an obstacle and close to the robot arm, the fuzzy motion controller is activated to adjust the path of the end-effector of the robot arm. Then, the robot arm takes the optimal posture for collision avoidance with the obstacle. To show the feasibility of the developed algorithm, numerical simulations are carried out with changing both the positions and sizes of obstacles. It was concluded that the proposed algorithm gives a good performance for obstacle avoidance.

**Key Words :** Redundant Robot(여유자유도 로봇), Fuzzy Control(퍼지 제어기), Collision Avoidance(장애물 회피), Posture Optimization(자세 최적화)

### 1. 서론

현재 산업현장에서 사용되고 있는 로봇의 대부분은 용접작업, 도장작업 및 단순한 집기 및 놓기(pick and place) 작업에 이용되고 있다. 좀더 복잡한 작업환경이나 유연성(flexibility)이 필요한 작업환경에서 로봇을 사용하려면 예측할 수 없는 장애물에 대해서도 우회 성능을 갖는 것이 필요하다. 이를 위해 여러 가지 센서들을 조합하여 로봇의 움직임을 지능화하는 방법들이 최근에 많이 연구되고 있다.

로봇이 작업 영역을 이동할 때 발생하는 장애

물을 우회하는데 사용되는 경로 계획법은 크게 2가지로 구분할 수 있다. 하나는 상태 공간법(configuration space method)<sup>(1)(2)</sup>이며, 다른 하나는 인공전위장법(artificial potential field method)<sup>(3)(4)</sup>이다. 상태공간법은 로봇의 관절 운동이 n-차원의 상태공간에서 꺾은 운동으로 변환되며, 장애물 또한 상태공간에서 장애물 영역으로 치환된다. 따라서 장애물과 충돌이 발생하지 않는 안전한 경로를 구할 수 있게 되며, 이 경로로부터 각 관절의 회전각을 구하는 방법이다. 이 방법은 계산 시간이 많이 들어 적용에 한계가 있다. 인공전위장법은 인공적인 전위장 함수를 이용하는 방법인데 장애물

† 2001년 9월 11일 접수

\* 울산대학교 기계공학과

\*\* 울산대학교 대학원 기계공학과

이 있는 위치는 전위장값을 높게 할당하며, 로봇이 도달해야 하는 목표 위치는 전위장값을 최소값으로 할당해서 로봇이 목표위치에 도달하게 하는 방법이다. 그러나 이 방법은 이동용 로봇에서는 쉽게 적용이 가능하나 관절형 로봇에 적용하기는 매우 어렵다. 또한, 로봇이 목표를 찾아가는 도중에 지역최소점(local minimum)<sup>(5)</sup>에 빠지는 경우가 발생하기도 한다.

최근에는 퍼지 및 신경회로망 기법이 로봇의 궤적 추적 또는 장애물 우회를 위한 경로 계획에 대해 많이 적용되고 있다. 퍼지 및 신경회로망 기법은 앞에서 언급한 상태공간법이나 인공지능방법에 비하여 시스템을 해석하기 위한 모델링이 간단하며, 경로 탐색 알고리즘을 작성하는데 있어서 복잡한 기구학 및 동역학적 계산식을 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 장애물 우회하여 도달할 수 있도록 하였다. 그러나, 이 경우 돌발적으로 발생할 수 있는 미지의 장애물들과의 충돌을 예방할 수 없는 문제점이 있다. Bagchi<sup>(9)</sup>와 Shin<sup>(10)</sup> 등은 퍼지논리를 이용하여 여유자유도를 가진 로봇을 임의의 시작 위치에서 목표 위치까지 이동하면서 이동 중에 발생하는 장애물과의 충돌을 예방할 수 있음을 보였다. 그러나, 이들 논문에서는 로봇 팔의 끝점의 위치가 다음 위치로 이동할 시점에서 장애물과 로봇 팔과의 충돌 가능성을 최소화하기 위한 최적화(optimization) 문제를 고려하고 있지 않거나 유연하지 못한 최적화 함수를 이용하였기 때문에 장애물의 위치에 따라 충돌이 발생하는 부분이 존재할 가능성을 배제할 수 없다.

본 논문에서는 퍼지제어와 로봇팔의 자세최적화 함수를 이용하여 돌발적으로 발생하는 장애물을 우회할 수 있는 여유자유도 로봇팔의 장애물 우회 알고리즘을 개발하였다. 로봇의 이동 중에 장애물이 존재하는 경우 장애물의 크기 및 위치정보가 로봇제어 시스템에 전달되면, 퍼지 운동제어

기(fuzzy motion controller)는 로봇 팔 끝점의 이동 방향을 결정해 준다. 그 다음 자세최적화 알고리즘을 이용하여 장애물과의 충돌위험을 최소화하고 부드러운 움직임을 유지할 수 있는 최적화자세를 결정하게 된다. 이와 같은 과정을 반복하여 로봇은 목표위치에 도달하게 된다. 제안된 퍼지 운동제어기와 자세최적화 알고리즘의 타당성을 보여주기 위하여 장애물의 위치 및 크기를 바꾸어 가면서 다양한 전산모의실험을 수행하였다.

## 2. 여유자유도 로봇팔의 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. 예를 들어 로봇 팔의 작동 중에 소요되는 에너지를 최소화시킨다든지, 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.

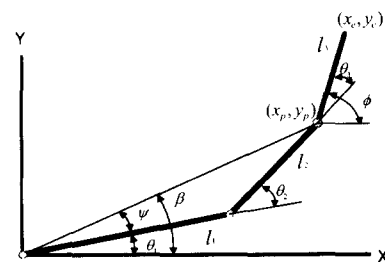


Fig. 1 Schematic Diagram of 3-Axis Redundant Robot Arm

### 2.1 로봇팔의 역기구학 (Inverse Kinematics)

여유자유도를 갖는 로봇 팔의 모델은 2 차원 (X,Y) 좌표계에서 움직이는 3 자유도 수평관절 로

봇으로 구성할 수 있다. Fig.1 에서 로봇 팔의 끝점의 좌표  $(x_e, y_e)$  와 로봇 팔의 3 번째 링크가 (X,Y) 좌표와 이루는 각도를  $\phi$  라고 하면, 로봇 팔의 3 번째 링크의 원점의 좌표  $(x_p, y_p)$  의 위치는 식 (1)과 같이 구해진다.

$$\begin{aligned} x_p &= x_e - l_3 \cos \phi \\ y_p &= y_e - l_3 \sin \phi \end{aligned} \quad (1)$$

삼각함수의 공식으로부터

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

위식을 정리하면

$$\cos \theta_2 = \frac{x_p^2 + y_p^2 - l_1^2 - l_2^2}{2l_1l_2} \quad (3)$$

$$\sin \theta_2 = \sqrt{1 - \cos^2 \theta_2}$$

식(3)으로부터 2 번째 관절각  $\theta_2$  는 식 (4)와 같이 구해진다.

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

Fig.1 에서

$$\beta = a \tan 2(x_p, y_p) \quad (5)$$

$$\psi = \cos^{-1} \left[ \frac{x_p^2 + y_p^2 + l_1^2 - l_2^2}{2l_1 \sqrt{x_p^2 + y_p^2}} \right]$$

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

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

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

## 2.2 퍼지(Fuzzy) 운동제어기

로봇의 이동 중에 돌발적인 장애물이 존재하는 경우, 장애물의 크기 및 위치정보가 로봇제어 시스템에 전달되면, 퍼지 운동제어기(fuzzy motion controller) 는 로봇 팔 끝점의 이동방향을 결정해 준다. Fig.2 는 로봇이 움직일 때 돌발적으로 발생한 장애물이 로봇 팔의 끝점의 진행방향 가까이 놓여져 있는 모습을 보여주고 있다. 퍼지제어를 이용하기 위하여 필요한 입력변수는 Gap 과 Length 로 선정하였다. Gap 은 로봇 팔의 끝점(end-

effector)이 진행방향의 선상으로부터 장애물의 법선방향으로 떨어진 거리로 정의되고, Length 는 장애물이 로봇 팔의 끝점에서 로봇 팔의 진행방향으로 떨어져 있는 거리로 정의된다.

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

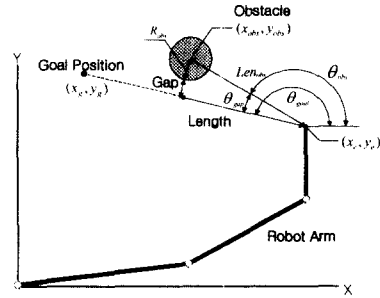


Fig. 2 Parametric Relations between a Robot Arm and an Obstacle

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

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

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

$$\theta_{gap} = (\theta_{goal} - \theta_{obs}) \operatorname{sgn}(x_{goal} - x_{start}) \quad (10)$$

퍼지제어기의 입력변수 Gap 과 Length 는 다음과 같은 식으로 표시된다.

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

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

그리고, 퍼지제어에 사용된 출력변수는  $\Delta \theta$  (로봇 팔의 끝점의 이동방향 수정각)로 선정하였다. 입력변수의 퍼지 세트 Gap 은 NB(Negative Big), NS(Negative Small), ZE(Zero), PS(Positive Small), PB(Positive Big)으로 선정하였고, 또 다른 입력 변수인 Length 는 ZE(Zero), PS(Positive Small), PM(Positive Medium), PB(Positive Big)으로 선정하였다. 출력변수에 대한 퍼지세트  $\Delta \theta$  는 ZE(Zero),

PS(Positive Small), PM(Positive Medium), PB(Positive Big), PVB(Positive Very Big)으로 선정하였다. 사용된 퍼지 소속함수(fuzzy membership function)는 Fig.3 에 나타나 있다.

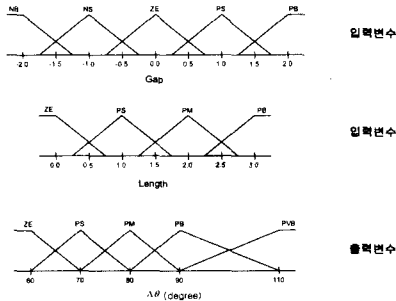


Fig. 3 Membership Function of Input and Output Variables

입력 및 출력변수의 퍼지세트에 대한 퍼지 제어 규칙인 **FAM(Fuzzy Associative Memory)**은 Table1 에 나타나 있다. FAM 이란 입력변수와 출력변수의 퍼지 세트간의 논리적으로 타당한 IF~THEN 규칙을 함축적으로 나열한 것이라고 볼 수 있다. 예를 들어 FAM 의 좌측상단의 블록을 달리 표현하면 다음과 같은 논리규칙으로 표시된다.

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

그리고, 퍼지제어기는 FAM 을 이용하여 임의의 입력변수가 입력 되었을 때, 비퍼지화(defuzzification method) 방법<sup>(11)</sup>을 이용하여 명확한 출력 값을 제공하게 된다.

Table 1 Fuzzy Associative Memory (FAM)

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

### 2.3 여유자유도 로봇팔의 자세최적화 알고리즘

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

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

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 역기구학 식은 식 (13)과 같이 구해진다.

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

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

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

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 설명해 주는 식이다. 식 (13)의 내부운동 중에서 로봇 팔이 원하는 성능함수(performance criterion)를 최대화 시켜 주도록 벡터의 값을 결정하여야 한다. 성능함수  $P(\Theta)$  라고 했을 때,  $P(\Theta)$  를 최대화하기 위한 벡터  $K$  의 관계는 식 (15)와 같이 구해진다.<sup>(12)</sup>

$$K = \frac{\partial P}{\partial \Theta} k \tag{15}$$

식(15)에서  $\partial P / \partial \Theta = [\partial P / \partial \theta_1, \partial P / \partial \theta_2, \dots, \partial P / \partial \theta_n]^T$ , 그리고  $k$  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 이동 중에 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.4 에서 보는 바와 같이 장애물로부터 가능한 멀리 떨어져 있는 자세를 취하도록 성능함수를 설정하였다.

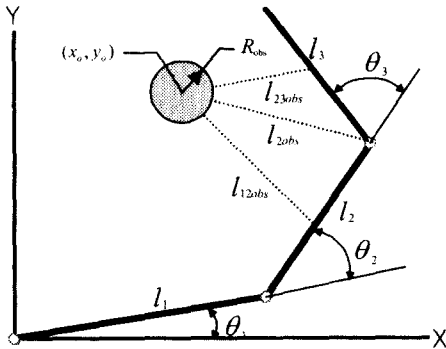


Fig. 4 Configuration of a Robot Arm and an Obstacle for Performance Criterion

$$P(\Theta) = C[C_{12}(l_{12obs} + \beta_1 l_{2obs}) + C_2 l_{2obs} + C_{23}(l_{23obs} + \beta_2 l_{2obs}) + C_{theta} \beta_{theta} \theta_2 + C_{det} \beta_{det} \sqrt{\det(J^T)}] \quad (16)$$

식 (16)에서 로봇팔과 장애물간의 길이 변수는 다음과 같다.

$$l_{2obs} = \left[ (l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) - x_o)^2 + (l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) - y_o)^2 \right]^{1/2} - R_{obs}$$

$$l_{12obs} = \left[ \left( l_1 \cos \theta_1 + \frac{l_2 \cos(\theta_1 + \theta_2)}{2} - O_x \right)^2 + \left( l_1 \sin \theta_1 + \frac{l_2 \sin(\theta_1 + \theta_2)}{2} - O_y \right)^2 \right]^{1/2} - R_{obs}$$

$$l_{23obs} = \left[ \left( l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + \frac{l_3 \cos(\theta_1 + \theta_2 + \theta_3)}{2} - O_x \right)^2 + \left( l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) + \frac{l_3 \sin(\theta_1 + \theta_2 + \theta_3)}{2} - O_y \right)^2 \right]^{1/2} - R_{obs}$$

그리고, 식 (16)의 오른쪽 4 번째 항은 2 번째 관절각  $\theta_2$  가 0 도 이하로 되는 경우, 즉 하관절(lower elbow) 에서 상관절(upper elbow)의 형태로 로봇팔이 바뀌는 것을 방지하기 위하여 첨가하였고, 5 번째 항은 로봇 팔의 동작 중에 부드러운 운동, 즉 조작성(manipulability)를 높이기 위하여 첨가되었다.

위 식에서  $x_o, y_o$ 는 각각 장애물 중심의 x 및 y 성분이다. c 는 임의의 상수이며, 나머지 계수  $C_{12}, C_2, C_{23}, C_{theta}, C_{det}$  의 값은 로봇팔의 이동 중에 로봇팔과 장애물의 상대위치와 자세에 따라 바뀌게 된다. 이렇게 함으로써, 로봇팔의 어느 부분이 장애물에 부딪히거나, 로봇의 2 번째 관절의 각도가 최소 규정각보다 작아지거나 최대 규정각보다 커지는 경우를 방지할 수 있게 된다. 위의 계수들을 조정하는 규칙은 Table2 에 나타나 있다. Table2 에서 보는 바와 같이 장애물과 로봇팔의 링크가 적절하게 떨어져 있게 되면 성능함수 각 요소의 계수들은 같은 가중치를 갖게 된다. 그러나, 임의의 링크 부분이 장애물과 근접하게 되면 해당되는 요소에 대한 계수 가중치를  $\zeta$  배 만큼 증가하게 된다. 표의 마지막 규칙은 로봇 팔이 장애물과의 충돌이 임박하는 경우 혹은 로봇의 2 번째 관절의 각도가 비정상적으로 증가 혹은 감소하는 경우 로봇은 비상정지를 하게 된다.

Table 2 The Coefficient-Selection Rules of the Performance Function

	$C_{12}$	$C_2$	$C_{23}$	$C_{theta}$	$C_{det}$
$l_{12obs} < R_{obs} \text{ or } l_{23obs} < R_{obs} \text{ or } l_{2obs} < R_{obs}$	1	1	1	1	1
$R_{obs} < l_{12obs} \text{ or } R_{obs} < l_{23obs} \text{ or } R_{obs} < l_{2obs}$	$\theta_2 = \min\{\theta_{min}, \theta_{max}\}$	$C_{12} \times \zeta$	1	1	1
	$\theta_2 = \min\{\theta_{min}, \theta_{max}\}$	1	$C_{23} \times \zeta$	1	1
	$\theta_2 = \min\{\theta_{min}, \theta_{max}\}$	1	1	$C_{theta} \times \zeta$	1
$\theta_2 < \theta_{min}$	1	1	1	$C_{theta} \times \zeta$	1
$\theta_2 > \theta_{max}$	1	1	1	$C_{theta} \times \zeta$	1
$\theta_2 < \theta_{min} \text{ or } \theta_2 > \theta_{max} \text{ or } l_{12obs} < R_{obs} \text{ or } l_{23obs} < R_{obs} \text{ or } l_{2obs} < R_{obs}$	Emergency Stop				

### 2.4 경로제어 알고리즘

Fig.5 는 여유자유도를 갖는 로봇 팔의 경로제어 알고리즘에 대한 제어 흐름도이다. Fig.5 에서 로봇 팔이 계획된 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)  $P(\Theta)$  를 최대화 시키는 자세를 구하게 된다. 그리고, 로봇과 장애물과의 관계로부터 성능함수의 계수들을 새롭게 조정 한 후 전체과정을 반복하면서 목표위치에 도달하게 된다.

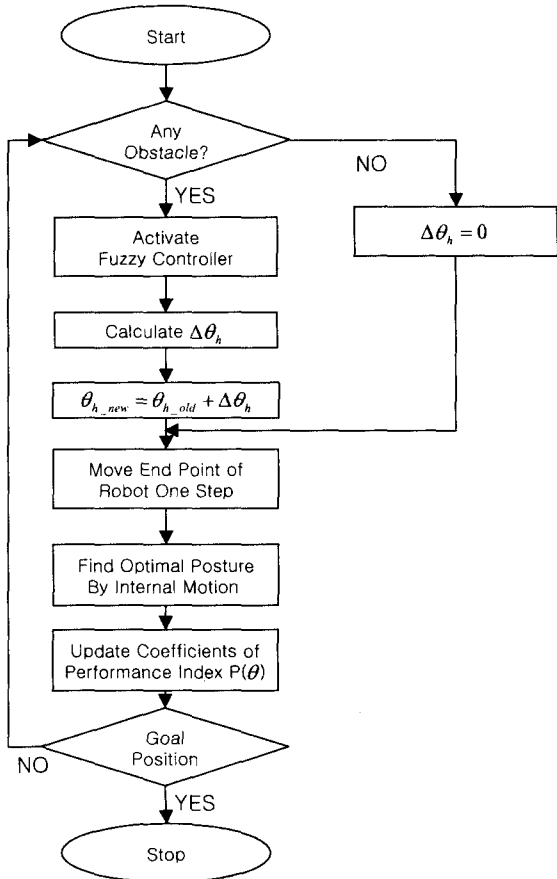


Fig. 5 Flowchart of the Motion Control Algorithm

### 3. 전산모의 실험

제안된 장애물우회 알고리즘의 성능을 확인하기 위하여 몇 가지 전산 모의실험을 수행하였다. 모의실험을 위하여 사용된 3 관절 로봇 팔의 크기는  $l_1 = 60$ ,  $l_2 = 40$ ,  $l_3 = 25$  이다. 식 16 에서

사용된 계수들은  $C=0.01$ ,  $\beta_{12} = 0.5$ ,  $\beta_{23} = 0.5$ ,  $\beta_{theta} = \pi/180$ ,  $\beta_{det} = 0.01$  과 같이 선정되었다. 그리고 Table2 의 계수들은  $\theta_{min} = 30^\circ$ ,  $\theta_{max} = 150^\circ$ ,  $\theta_{cri\_min} = 5^\circ$ ,  $\theta_{cri\_max} = 175^\circ$ ,  $\lambda = 2$ ,  $\omega = 0.5$ ,  $\zeta = 2$  와 같이 선정되었다. 모의실험을 하기 위한 입력변수는 초기위치에서의 로봇 팔의 끝점  $x, y$  좌표, 초기위치에서 로봇 팔의 3 번째 링크의 각도, 장애물  $x, y$  좌표와 반경, 그리고 목표 위치에서 로봇 팔의 끝점  $x, y$  좌표이다. 장애물의 형상은 원의 형태로 가정하였으며 장애물의 크기와 위치는 원의 반경과 중심의 위치로 표현하였다. 그리고 장애물의 위치와 로봇 팔의 끝점의 간격이 장애물 직경의 5 배 이내로 가까워졌을 때 퍼지제어기가 작동하여 로봇 팔의 끝점의 진행 방향을 수정하도록 하였고, 이때부터 퍼지제어와 동시에 로봇 팔의 최적화 자세를 찾는 과정을 수행하였다

Fig.6 에서 보는 바와 같이 장애물이 로봇 팔의 끝점으로부터 비교적 멀리 떨어져 있는 경우 로봇 팔의 끝점은 직선적인 경로를 따라 목표위치에 도달하게 된다. Fig.7 과 같이 장애물이 로봇 팔의 끝점 부근에 존재하면, 퍼지 운동제어기가 작동하여 로봇 팔의 끝점의 이동방향은 수정되어 장애물을 우회하게 된다. Fig.7 에서 보는 바와 같이 로봇 팔 끝점의 시작위치는 (110,0) 이고, 로봇 팔의 3 번째 링크가 x 축과 이루는 각도는  $30^\circ$ , 로봇팔 끝점의 목표위치는 (0,110), 그리고 장애물의 위치는 (60,50)이고 장애물의 반경은 5 의 값을 가진다. Fig.8 과 Fig.9 의 경우는 장애물의 위치가 초기위치 및 목표위치 부근에 존재하는 경우이며, 장애물우회 작업이 까다로운 경우라고 할 수 있는데, 이 경우도 장애물우회 작업이 성공적으로 수행됨을 볼 수 있다. Fig.10 은 로봇 팔의 시작 및 목표 위치가 Fig.7 의 경우와 반대로 된 경우의 결과이다. Fig.11 은 장애물의 크기가 다른 경우의 2 배로 증가하고, 시작 및 목표위치도 약간 변경된 경우의 결과이다. Fig.12 와 Fig.13 은 장애물이 시작위치에서 임의의 방향으로 이동하는 경우에 로봇팔의 장애물우회 성능을 나타내고 있다. 이상의 전산모의 실험에서 알 수 있듯이, 본 연구에서 제안된 장애물우회 알고리즘은 다양한 장애물의 위치, 크기 및 로봇 팔의 위치에 상관 없이 우수하게 장애물우회 성능을 나타냄을 알 수 있다.

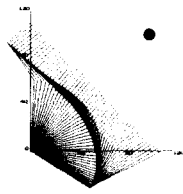


Fig. 6 Simulation when the obstacle is far from the robot arm: Start Position: (110,0,30), Goal Position: (0,110), Obstacle: (100,100,5)

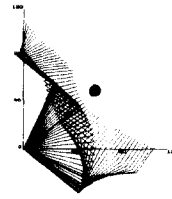


Fig. 10 Simulation when the goal and start positions are reversed: Start Position: (0,110, 90), Goal Position: (110,0), Obstacle: (60,50,5)

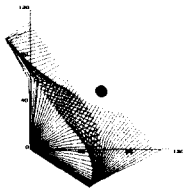


Fig. 7 Simulation when the obstacle is in the middle of workspace: Start Position: (110,0,30), Goal Position: (0,110), Obstacle: (60,50,5)

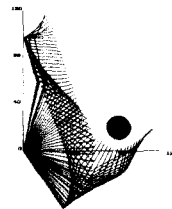


Fig. 11 Simulation when the obstacle is bigger twice than before: Start Position: (110,20, 45), Goal Position: (20,120), Obstacle: (70,20,10)

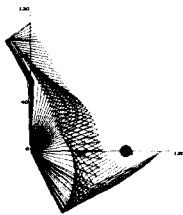


Fig. 8 Simulation when the obstacle is close to start position: Start Position: (110,0,30), Goal Position: (0,110), Obstacle: (80,0,5)

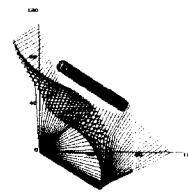


Fig. 12 Simulation when the obstacle is moving : Start Position: (110,0, 30), Goal Position: (0,110), Initial Obstacle: (70,40,5)

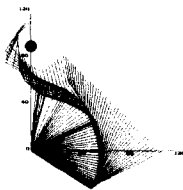


Fig. 9 Simulation when the obstacle is close to the goal position: Start Position: (110,0,30), Goal Position: (0,110), Obstacle: (0,90,5)

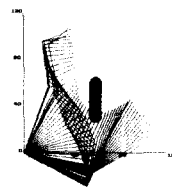


Fig. 13 Simulation when the obstacle is moving : Start Position: (110,20, 45), Goal Position: (40,110), Initial Obstacle: (60,60,5)

#### 4. 결론

본 논문에서는 여유자유도를 갖는 수평 다관절 로봇이 초기 위치에서 목표 위치까지 정해진 궤적을 따라 이동할 때 돌발적으로 발생할 수 있는 장애물을 퍼지제어와 자세 최적화 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 위치에도달하게 되지만, 장애물이 충돌위험 영역으로 접근하게 되면 퍼지제어시스템이 작동하게 되어 로봇의 끝점의 이동 방향을 수정하게 하여 주어진 궤적을 이탈하게 하고 장애물을 우회하여 목표 위치에 도달하게 한다. 특히, 기존의 논문과는 달리 본 논문에서는 자세최적화를 위한 성능함수의 계수를 매 샘플링마다 조정할 수 있도록 하여, 계수가 고정된 성능함수를 사용하는 경우보다 장애물과의 충돌위험성을 더욱 줄일 수 있게 하였다.

개발된 운동제어 알고리즘을 성능을 확인하기 위하여 다양한 모의실험을 수행하였다. 장애물의 크기와 위치를 변경하여 로봇이 장애물을 성공적으로 우회하는지 조사하였으며, 또한 이동장애물에 대해서도 적용하여 만족스러운 결과를 얻었다. 본 논문에서는 단순히 모의실험만으로 이루어졌으나 비전 센서(vision sensor)나 거리 센서 등을 이용하여 장애물의 크기나 위치를 실시간적으로 계산하게 되면 실제 작업에도 쉽게 적용이 가능하다고 본다.

#### 후 기

본 연구는 과학재단 지원 울산대학교 지역협력센터(RRC)지원으로 수행되었으며, 관계자 여러분께 감사 드립니다.

#### 참고문헌

1. T.Lozano-Perez, "Spatial Planning: a Configuration Space Approach," IEEE TRans. Comp, pp. 108-120, 1983.
2. T.Lozano-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. J. of Robotic Research, Vol. 5, No. 1, pp. 90-98, 1986.
4. J.Barraguang and J.C.Latombe, "Robot Motion Planning: A Distributed Representation Approach," Int. J. of Robotics Research, Vol. 10, No. 6, pp. 628-649, 1991.
5. 이병룡, "포텐셜함수를 이용한 자율주행 로봇들 간의 충돌예방을 위한 주행제어 알고리즘의 개발," 한국정밀공학회, 제 15 권, 제 6 호, pp. 107-115, 1998.
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 Planner: 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.
10. 신현배, 이병룡, "다관절 지능 로봇시스템을 위한 장애물 우회 연구," 한국정밀공학회, 제 15 권 8 호, pp. 165-173, 1998.
11. L.Wang, "Adaptive Fuzzy Systems and Control," Prentice Hall.
12. Yoshikawa T, "Analysis and Control of Robot Manipulator with Redundancy," Robotics Research: 1<sup>st</sup> Int. Sym, MIT Press, pp. 735-747, 1984.