

## 비전기반 암 로봇의 퍼지제어기 설계

신 화 영, 김 영 중, 임 묘 택

고려대학교 전기공학과

전화 : 02-3290-3804 / 핸드폰 : 016-9353-5490

### Design of Fuzzy Controller for Vision-based Arm Robot

Hwa-Young Shin, Young-Joong Kim, Myo-Taeg Lim

Dept. of Electrical Engineering, Korea University

E-mail : syoung.kyjoong, mlim@elec.korea.ac.kr

#### Abstract

In this paper, fuzzy logic controllers are designed for compensation of distance errors. Because we can't know information of the depth in a mono camera, these errors are occurred. Also, they are increased as a target object is to keep away from a center of image. Therefore, the errors for each position of joints of an arm robot should be modeled, but accurate models can't be obtained because of no information of the depth, uncertain feature points of image, parameter uncertainties, and illumination. Hence, fuzzy logic controllers for each error are designed for compensation. This paper consists of color image processing, error modeling, and the controller design. Experimental results are given to verify the effectiveness of our proposed method.

#### I. 서론

퍼지이론은 현상의 불확실한 상태를 그대로 표현해 주는 방법으로서 1965년 미국 버클리 대학의 Zadeh에 의해서 처음 소개 되었다[1][3]. 1974년 Mamdani 는 steam engine의 pilot plant 에 최초로 퍼지이론을 적용하여 퍼지제어의 가능성을 실증하였다. 그 이후 일본의 Sugeno 는 퍼지 적분(Fuzzy Integral), 퍼지 모델

링(Fuzzy Modeling)등을 제안하여 퍼지제어 시스템의 체계적인 구축에 많은 영향을 미쳤다. 퍼지이론은 애매하게 표현된 자료를 우리에게 유용한 자료로 만들기 위하여, 퍼지집합(fuzzy set), 퍼지논리(fuzzy logic), 퍼지숫자(fuzzy number) 등의 개념을 포함하고 있으며 수학적인 계산 방법도 잘 개발되어 있다[1][2][3]. 본 논문에서는 비주얼 서보잉 시스템[6][7]에 퍼지제어기를 설계하여 암 로봇의 각 관절의 위치제어를 수행한다. 이 시스템의 구성은 영상처리, 퍼지제어기, 로봇시스템 부분으로 구성된다. 전체적인 흐름은 영상처리 부분에서 비전정보를 받아 로봇시스템에 보내어 작업을 수행하게 되는데 이때 발생하는 오차들을 보정하기 위해 퍼지제어기[4]를 설계한다. 실험은 퍼지제어기를 적용했을 때와 적용하지 않았을 때 로봇의 추적궤도를 X축과 Y축으로 나누어 비교분석 하였다.

#### II. 시스템의 구성

그림1은 전체시스템의 블록다이어그램을 나타내고, 영상처리, 퍼지제어, 그리고 로봇 시스템으로 구성된다. 로봇 시스템은 로봇 몸체, 로봇 컨트롤러, 그리고 컴퓨터로 구성된다. 로봇은 컴퓨터와 연결되어 RS-232 시리얼 통신으로 데이터를 받는다. 그러나 로봇의 위치에 대한 데이터 값을 피드백으로 받을 수 없기 때문에 비전 피드백을 이용하여 로봇의 위치와 방향을 알아낸다. 따라서 로봇의 정면에 색깔을 부착하여 로봇이 현

재 어느 위치에 있는지 또는 어느 방향으로 움직였는지를 알 수 있다. 로봇은 좌우로의 회전, 상하, 잠기 동작을 수행할 수 있다. 카메라가 로봇의 정면에 설치되어 있을 때 영상이미지의 중심영역에서는 실제 보이는 부분과 영상에서 보이는 부분의 차이가 거의 없지만 중심에서 멀어질수록 실제 보이는 부분과 영상에서 보이는 부분의 차이가 비선형적으로 커짐을 알 수 있다. 따라서 이러한 오차를 줄이기 위해 퍼지제어기를 설계하여 오차를 보정해준다.

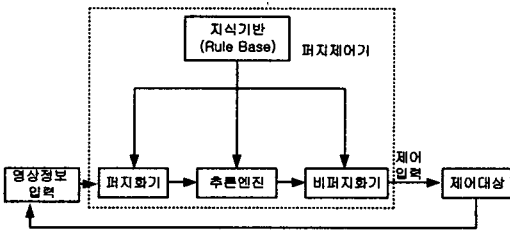


그림1. 전체 시스템의 블록다이어그램

### 2.1 영상처리 시스템의 개요

영상처리 시스템은 영상을 획득하고 처리하여 암 로봇의 정면에 부착된 마크의 색상과 물체의 색상을 인식한 후 처리된 데이터를 프로그램으로 전송하는 역할을 수행한다. 먼저, 획득된 영상을 RGB성분으로 나누어 실험에 사용되는 색상 값의 범위를 찾는다. 색상 값의 범위는 처음에 넓은 범위에서 시작하여 차츰차츰 범위를 좁혀감으로써 다른 색상이 추출되지 않을 때까지 수행한다. 다음으로 카메라의 해상도나 조명의 불균일성 등에 의해, 잡음이 있을 수 있다. 따라서 어느 화소 수 미만인 연결 성분들을 제거하여 우리가 추출하려는 영역만 남을 수 있도록 한다. 이것을 크기 필터링이라고 한다. 크기 필터링을 수행한 후 각각의 색상영역의 중심위치를 구한다.

```

for (int j=0; j<Height; j++)
{
    for (int i=0; i<Width; i++)
    {
        if (Color(i, j) == feature color)
        {
            start_i += i;
            start_j += j;
            A++;
        }
    }
}
x' = start_i / A,    y' = start_j / A
    
```

그림2. 색상영역의 중심위치 구하는 과정

위의 그림2는 색상영역의 중심위치를 구하는 프로그램 과정을 나타낸다. 여기에서  $x'$ 과  $y'$ 은 물체의 중심위치 값을 나타낸다. 각각의 색상영역의 중심 위치를 구했다면, 이 색상들 사이의 거리나 위치관계를 알 수 있으므로, 로봇의 위치와 방향을 계산할 수 있다.

### 2.2 로봇 시스템의 구조

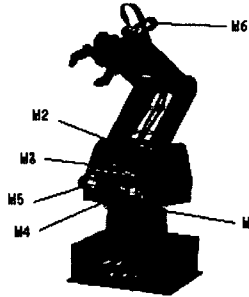


그림3. 다관절 로봇(Arm-7220C-4)

로봇 시스템은 로봇 몸체, 로봇 컨트롤러로 구성되어 있다. 그림3은 로봇의 몸체를 나타내고 있는데, 허리(M1), 어깨(M2), 엘보우(M3), 피치(M4), 롤(M5)의 5관절과 그립퍼(M6)로 6개의 DC 모터로부터 동력을 전달받으며, 특히 엘보우, 피치, 롤은 모터의 기어로부터 타 임벨트를 통해 동력을 전달받게 된다.

## III. 퍼지 논리 제어기

### 3.1 로봇의 물체추적에 대한 오차

로봇의 중심이 영상 이미지의 중앙(320, 240)에 있다고 가정한다. 그림4는 로봇의 곡선궤적에 대한 오차를 나타내고 있는데 숫자 1은 영상정보에 의한 로봇의 위치이고, 숫자 2는 실제 로봇이 정지해야할 위치이다. 숫자 3은 물체의 위치를 나타낸다. 이 그림에서 볼 때 로봇은 숫자 2에서 정지했을 때 물체와 일직선이 되어 다른 동작을 수행할 수가 있다. 그러나 로봇은 영상정보에 의해서 숫자 1에서 정지하게 된다. 따라서 숫자 1과 2사이만큼의 거리오차가 발생하게 된다. 이 오차는 영상의 중심(320, 240)에서 멀어질수록 비선형적으로 증가한다.

$$e_i = 160 - 160 \left( \sin \frac{\pi}{640} i \right) \quad (3.1)$$

위의 식을 일반화하면 (3.2) 식이 되고,  $\epsilon_{max}$ 는 오차

의 최대 값을 나타낸다.

$$e_i = \epsilon_{\max} - \epsilon_{\max} \left( \sin \frac{\pi}{640} i \right) \quad (3.2)$$

$$e_j = F - F \left( \sin \frac{\pi}{640} i \right) \quad (3.3)$$

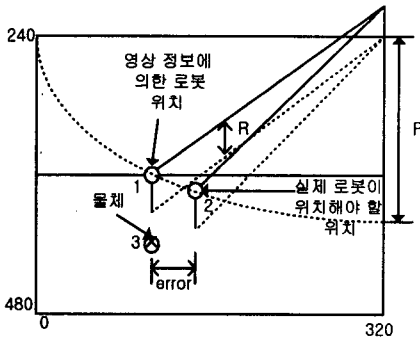


그림4. 로봇의 픽셀레제에 대한 오차

### 3.2 퍼지 논리 제어기 설계

제안된 퍼지제어기는 식 (3.4), (3.5)와 같다.

$$u_i = f_i(e_i, e_d) \quad (3.4)$$

$$u_j = f_j(e_j, e_d) \quad (3.5)$$

여기서  $e_i, e_j$  은 I축과 J축에서의 오차이고,  $e_d$  는 로봇과 물체사이의 높이에 대한 오차이다.  $u_i, u_j$ 는 I축과 J축에서의 제어기의 출력, 그리고  $f_i, f_j$ 는 I축과 J축에서의 퍼지제어기를 나타낸다.

퍼지제어기의 설계에서 제어규칙의 결정은 가장 중요하다. 제어규칙의 형태는 다음과 같다.

*If  $e$  is  $A_1$  and  $e_d$  is  $A_2$  then  $u$  is  $B$*

여기서  $A_1, A_2, B$  는 각각 I축 오차, 로봇과 물체 사이의 높이오차, 제어기 출력의 멤버쉽 함수이다. 위의 퍼지규칙에 의한 출력의 추론법은 Mamdani의 최대-최소법을 사용하였고[5], 식 (3.6), (3.7)과 같이 표현된다.

$$u_B(u) = \max[\min(w, u_B(u))] \quad (3.6)$$

$$w = \min(u_{A_1}(e), u_{A_2}(e_d)) \quad (3.7)$$

추론된 퍼지값을 실제 제어입력으로 바꾸는 방법에는 여러 가지가 있지만 본 논문에서는 무게 중심법을 사용하였다[5].

표1은 I축의 오차  $e_i$  를 10개의 크기로 나누고 비선형적인 오차를 보정하기 위해 가중치를 두어 퍼지 제

어규칙을 작성한다. 표2는  $e_j$  와  $e_d$  를 사용하여 J축에 대해서 퍼지 제어규칙을 작성하는데,  $e_j$  와  $e_d$  는 거의 근사적으로 선형이라고 할 수 있으므로 가중치를 적용하지 않고 퍼지제어 규칙을 작성한다.

가중치	0	0.16	0.31	0.45	0.59	0.71	0.81	0.9	0.95	0.99	1
축 높이	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0.16	0.62	1.35	2.36	3.55	4.86	6.3	7.6	8.91	10
2	0	0.32	1.24	2.7	4.72	7.1	9.72	12.6	15.2	17.8	20
3	0	0.48	1.86	4.05	7.08	10.6	14.5	18.9	22.8	26.7	30
4	0	0.64	2.48	5.4	9.44	14.2	19.4	25.2	30.4	35.6	40
5	0	0.8	3.1	6.75	11.8	17.7	24.3	31.5	38	44.5	50
6	0	0.96	3.72	8.1	14.1	21.3	29.1	37.8	45.6	53.4	60
7	0	1.12	4.34	9.45	16.5	24.8	34.0	44.1	53.2	62.3	70
8	0	1.28	4.96	10.8	18.8	28.4	38.8	50.4	60.8	71.2	80
9	0	1.44	5.58	12.1	21.2	31.9	43.7	56.7	68.4	80.1	90
10	0	1.6	6.2	13.5	23.6	35.5	48.5	63	76	89	100

표1. I 축에 대한 퍼지 제어규칙

축 높이	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10
2	0	2	4	6	8	10	12	14	16	18	20
3	0	3	6	9	12	15	18	21	24	27	30
4	0	4	8	12	16	20	24	28	32	36	40
5	0	5	10	15	20	25	30	35	40	45	50
6	0	6	12	18	24	30	36	42	48	54	60
7	0	7	14	21	28	35	42	49	56	63	70
8	0	8	16	24	32	40	48	56	64	72	80
9	0	9	18	27	36	45	54	63	72	81	90
10	0	10	20	30	40	50	60	70	80	90	100

표2. J 축에 대한 퍼지 제어규칙

## IV. 실험 결과

그림5와 그림6은 퍼지제어기를 사용하였을 때와 사용하지 않았을 때 로봇의 추적과정을 X축과 Y축으로 구분하여 보여준다. 그림5의 X축 좌표표를 보면 로봇과 물체사이의 거리가 먼 거리일 경우 단지 4회만에 물체의 X축 픽셀에 도달하지만 그림6의 X축 좌표표에서는 10회의 입력을 가해야 도달함을 알 수 있다. 또한 그림5의 X좌표의 중간거리와 짧은 거리는 각각 4회와 2회의 입력을 필요로 한다. 그러나 그림6의 X좌표의 중간거리와 짧은 거리는 각각 7회와 3회의 입력을 필요로 한다. 이 같은 데이터에서 볼 때 먼 거리에서 퍼지

제어기의 효율성이 뛰어난을 알 수 있다. 퍼지제어기를 사용한 Y축 좌표는 각각 5회, 3회 그리고 2회만에 목적지에 도달하지만 퍼지제어기를 사용하지 않은 경우는 각각 6회, 6회 그리고 3회만에 도달함을 알 수 있다.

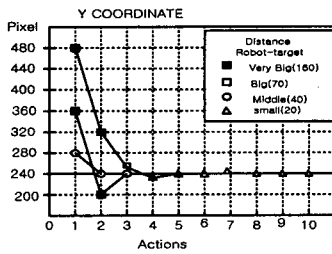
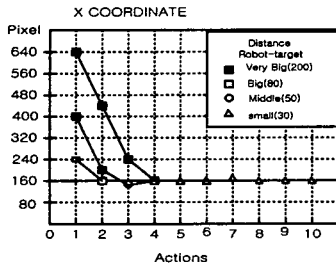


그림5. 퍼지 제어기를 적용한 경우

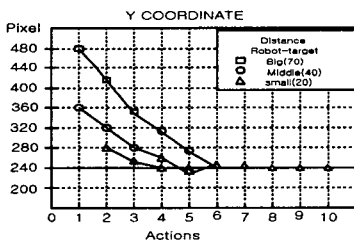
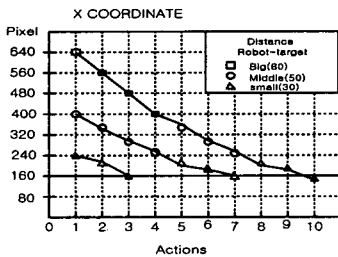


그림6. 퍼지 제어기를 적용하지 않은 경우

## V. 결론

본 논문에서는 비전기반 암 로봇이 물체를 추적하는데

발생하는 거리오차를 줄이기 위해 퍼지논리 제어기를 설계했다. 거리에 대한 오차는 영상 이미지의 중심에서 멀어질수록 비선형적으로 증가함을 알 수 있었다. 또한 로봇과 물체의 위치를 알려주는 색상영역의 중심점이 자주 변함으로 퍼지제어기를 사용하여 변화된 중심점의 위치를 보정해주었다. 이러한 결과에 대한 데이터는 퍼지제어기를 사용하여 오차를 보정해준 경우와 퍼지제어기를 사용하지 않고 실험한 경우를 비교함으로써 알 수 있다. 또한 퍼지제어기를 사용한 경우에서도 소속함수를 다르게 정함으로써 실험에 대한 결과가 다르게 나올 수 있다. 본 논문에서 사용한 로봇은 모델링 하기가 힘든 시스템이고, 로봇의 위치에 대한 피드백 정보도 받을 수 없다. 따라서 비전을 적용하여 로봇과 물체의 위치정보를 알 수 있었다. 그러나 비전 정보만으로 해결 할 수 없는 부분들이 발생했고 이러한 부분들에 퍼지제어기를 사용하여 해결했다는 점에서 퍼지제어기를 적절하게 사용했다고 보여진다.

## 참고문헌

- [1] 채석, 오영석, 퍼지 이론과 제어, 청문각, 2000.
- [2] 김도현, 퍼지 시스템, 에드텍, 1994.
- [3] 박종진, 최규석, 퍼지 제어 시스템, 교우사, 2001.
- [4] Wuwei Chen, James K. Mills, Jiixin Chu, Dong Sun. "A Fuzzy Compensator for Uncertainty of Industrial Robots", Proc. IEEE Int. Conf. on Robotics and Automation, May, 2001.
- [5] E. H. Mamdani, S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller", International Journal of Man-Machine Studies, Vol. 7, No 1, pp. 1-13, 1975.
- [6] Seth A. Hutchinson, Gregory D. Hager, Peter I. Corke, "A Tutorial on Visual Servo Control.", IEEE Trans. on Robotics and Automation. Vol. 12. No. 5. October 1996.
- [7] P. K. Allen, B. Yoshimi, and A. Timcenko. "Real-time Visual Servoing." in Proc. IEEE Int. Conf. Robotics Automat., Apr. 1991. pp. 851-856.