

# 유연 로봇 매니플레이터의 자동 구축 퍼지 적응 제어기 설계 Design of an Automatic Constructed Fuzzy Adaptive Controller (ACFAC) for the Flexible Manipulator

이 기 성 · 조 현 철 \*

Keeseong Lee, Hyun-Chul Cho \*

홍익대학교 전자전기공학부, 경북전문대학 전자과 \*

## 요 약

유연 로봇 매니플레이터의 위치 제어 알고리즘에 대한 연구를 하였다. 제안하는 알고리즘은 신경회로망의 학습 알고리즘에 근거한 자동 구축 퍼지 적응 제어기(ACFAC: Automatic Constructed Fuzzy Adaptive Controller)에 기본으로 한다. 제안하는 시스템은 비지도 경쟁 학습 알고리즘을 사용하여 입력 변수의 멤버십 함수와 지도 Outstar 학습 알고리즘을 사용하여 출력 정보를 학습시킨다. ACFAC는 유연 로봇 매니플레이터의 동역학 모델을 필요로 하지 않는다. ACFAC는 유연 로봇 매니플레이터의 끝점이 원하는 궤적을 따라가도록 설계되었다. 이 제어기의 입력은 위치 오차, 위치 오차의 미분 값과 오차의 variation에 의해 결정된다. ACFAC의 우수성을 보여주기 위해서 PID 제어나 신경회로망 알고리즘을 사용한 결과와 비교를 하였다.

## ABSTRACT

A position control algorithm of a flexible manipulator is studied. The proposed algorithm is based on an ACFAC(Automatic Constructed Fuzzy Adaptive Controller) system based on the neural network learning algorithms. The proposed system learns membership functions for input variables using unsupervised competitive learning algorithm and output information using supervised outstar learning algorithm. ACFAC does not need a dynamic modeling of the flexible manipulator. An ACFAC is designed that the end point of the flexible manipulator tracks the desired trajectory. The control input to the process is determined by error, velocity and variation of error. Simulation and experiment results show a robustness of ACFAC compared with the PID control and neural network algorithms.

## 1. 서 론

자율 주행 운반체의 경우 현재는 단순한 운반 수단으로만 사용되고 있지만 인간의 일을 대신하기 위해서는 자율 주행 운반체에 Manipulator를 장착해야 한다. 기존의 Manipulator를 장착하는 경우 견고한 재료로 제작되어 매우 무거워서 자율 주행 운반체의 운반 효율을 떨어뜨리게 한다. 이러한 단

점을 극복하기 위해서 사용이 되는 Flexible Manipulator는 기존의 Manipulator보다 가볍고, 안전하며 고속으로 사용할 수 있는 장점을 가지고 있다. 그러나 Flexible Manipulator는 기존의 Manipulator에 비해서 운동 방정식이 복잡하기 때문에 제어하기가 힘들다는 단점을 가지고 있다.

Flexible Manipulator의 제어를 위해 많은 노력이 이루어져 왔는데 제어기를 설계하기 위해서는 먼

\* 본 연구는 한국과학재단 핵심전문 연구비(951-0914-122-2) 지원으로 수행되었으며 지원에 감사를 드립니다

저 정확한 모델 식을 찾는 노력이 선행되었다. Finite Element Method를 이용해서 모델 식을 구한 방법 [2, 12] 과 Euler-Lagrange Equation 을 이용해서 구하는 방법 [13, 14, 15, 16, 17] 등이 있다. 한편 Flexible Manipulator의 제어를 위해서 많은 제어 알고리즘이 제안되었는데 Computed Torque Method [18], Optimal Control Theory [14, 16, 19], MRAC (Model Reference Adaptive Control) [20, 21], Self-tuning Control [4] 등이 있는데 이 방법들은 제어기 설계를 위해서 많은 계산 시간을 필요로 하는 단점이 있다. 또한 가장 널리 사용이 되고 있는 PID (Proportional, Integral, Differential) 제어기의 경우 각각의 제어 성분에 대한 이득 값이 고정되어 있기 때문에 비선형적인 특성에 적절히 대응하지 못하는 단점을 가진다. 이러한 단점을 극복하기 위해서는 정확한 동역학 운동 방정식을 구하고 이를 이용해서 비선형 제어기를 설계하여야 한다. 하지만 현실적으로는 정확한 운동 방정식을 구하기 어려울 뿐만 아니라 비선형 제어기의 경우 연산 시간 문제로 실시간 제어를 구축하기가 어렵다. 비선형적인 환경에 적절하게 대응하기 위해 인간의 학습 능력과 언어 처리 능력 등을 모방하는 새로운 구조와 알고리즘을 갖는 새로운 시스템의 개발이 필요하다. 오늘날 이러한 시스템을 구현하기 위하여 가장 많이 사용하는 기법이 신경회로망 [11] 과 퍼지이론 [1, 7, 8, 9, 10] 이다.

본 논문에서는 Flexible Manipulator의 끝점의 제어를 위해서 자동 구축 퍼지 제어기 [6,10] 를 이용하여 설계하였고, 이 제어기의 성능을 PID 제어, 신경회로망 제어와 끝점 부하를 바꾸어 가면서 시뮬레이션과 실험을 통해 Flexible Manipulator 위치 제어에 있어서 자동 구축 퍼지 제어의 방법이 강인함을 보여준다.

## 2. 유연 로봇 매니플레이터

Flexible Beam의 위치를 표현하기 위해서는 그림 1 을 이용하여 표현한다. Euler-Bernoulli 방정식으로 알려진 식은 다음과 같이 나타낼 수 있다 [3].

$$EI \frac{\partial^4 y(x,t)}{\partial t^4} + \rho \frac{\partial^2 y(x,t)}{\partial x^2} = 0 \quad (1)$$

여기서  $y(x, t)$  : Transverse Displacement  
 $x$  : 거리,  $t$  : 시간  
 $E$  : 탄성 계수,  $I$  : 관성 모멘트

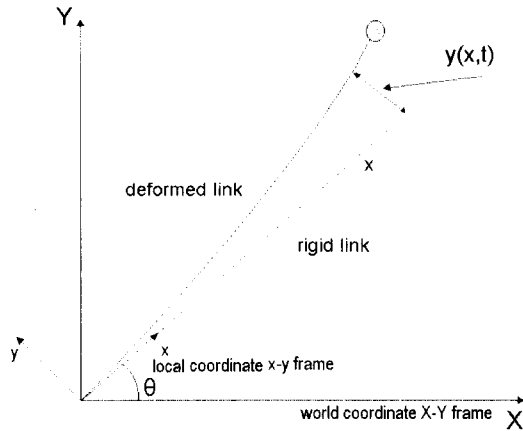


그림 1. Flexible Manipulator

### 2.1 Flexible Manipulator의 위치 방정식

Flexible Manipulator의 위치를 나타내기 위해서는 지역 좌표계(x-y)와 전역 좌표계(X-Y)를 정의해야 한다.  $y(x, t)$ 는 시간  $t$ , 거리  $x$ (지역 좌표계)일 때 유연한 Manipulator와 유연하지 않은 Manipulator와의 변위를 말하고,  $\theta$ 는 유연하지 않은 Manipulator의 초기 출발점과의 각도를 말한다.  $y(x, t)$ 은 시간  $t$ 와 변위  $x$ 에 의존하는 값이다.  $y(x, t)$ 는 변수 분리법에 의해 거리에 의존하는  $\phi_i(x)$ 와 시간에 의존하는  $\xi_i(t)$ 로 나타냄으로써 해를 구할 수 있다. 따라서  $y(x, t)$ 는 급수 형태로 다음과 같이 표현할 수 있다.

$$y(x, t) = \sum_{i=1}^{\infty} \phi_i(x) \xi_i(t) \quad (2)$$

많은 과거의 연구 사례를 통해서 식(2)에서 유한한 수의 항을 이용해서도 Flexible Manipulator의 동작을 표현할 수 있음을 알 수 있다. 본 논문에서는 4개의 기저 함수(Basis Function)를 이용해서 Flexible Manipulator의 횡단 변위를 표현하였다. 좌표 중심에서의 횡단 변위는 무시할 정도로 영향을 미치지 않으므로 본 논문에서는 생략하였다. 그러므로, 지역 좌표계에서 횡단 변위는 다음 식과 같이 된다.

$$y(x, t) = \sum_{i=1}^4 \phi_i(x) \xi_i(t) \quad (3)$$

여기서  $\phi_i(x)$ ,  $i = 1, 2, 3, 4$ 는 다음과 같이 경계 조건에 의해서 구해진다.

$$\begin{aligned} \phi_1(x) &= 1 - \frac{3}{L^2}x^2 - \frac{2}{L^3}x^3 & \phi_2(x) &= x - \frac{2}{L}x^2 + \frac{1}{L^2}x^3 \\ \phi_3(x) &= \frac{3}{L^2}x^2 - \frac{2}{L^3}x^3 & \phi_4(x) &= -\frac{1}{L}x^2 + \frac{1}{L^2}x^3 \end{aligned} \quad (4)$$

그림 1 과 식 (3)에 의해 Flexible Manipulator의 전역 좌표계에서의 위치를 알 수 있다.

$$\begin{aligned} P_x &= x \cos\theta - y(x, t) \sin\theta \\ P_y &= y \sin\theta - y(x, t) \cos\theta \end{aligned} \quad (5)$$

### 2.2 Flexible Manipulator 의 운동 방정

Flexible Manipulator 의 동력학 운동 방정식은 위치에너지와 운동에너지에 의한 Lagrange 방정식에 의해 유도할 수 있다. Lagrange 함수 T는 위치에너지와 운동에너지의 차로 표현한다.

$$T = V - P \quad (6)$$

여기서 V와 P는 각각 위치에너지와 운동에너지이다. Flexible Manipulator의 운동 모델 방정식은 Lagrange 방정식에 의해 구할 수 있다.

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} = \tau_i \quad (7)$$

여기서  $q_i$  와  $\tau_i$  는 각각  $i$  번째 독립변수와 외부 Torque이다.

위의 식 (7)에 의해서 Flexible Manipulator의 동력학 방정식을 다음과 같이 구할 수 있다[4].

$$\begin{bmatrix} \frac{1}{3} \rho L^3 + \xi^T M \xi & H \\ H & M \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\xi} \end{bmatrix} = \begin{bmatrix} \tau - 2\theta^T \xi^T M \xi \\ \theta^T M \xi - K \xi \end{bmatrix} \quad (8)$$

여기서

$$M = \rho \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{13L}{35} & -\frac{11L^2}{210} \\ 0 & 0 & -\frac{11L^2}{210} & \frac{L^3}{105} \end{bmatrix}, H = \begin{bmatrix} 0 \\ 0 \\ \frac{7L^2}{20} \\ \frac{L^3}{20} \end{bmatrix}, K = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \quad (9)$$

### 3. 자동 구축 퍼지 적응 제어기

#### (ACFAC : Automatic Constructed Fuzzy Adaptive Controller)

유연 로봇 매니퓰레이터의 위치 제어를 수행하기 위해 ACFAC 사용하였고 이때 제어기의 구조는 그림 2와 같다. ACFAC의 입력 값으로는 에러, 에러의 변화량과 끝점 속도를 사용하였고 출력은 진 동을 최소화하며 기준 궤적을 추종하게 하는 Torque 값이다. Torque 값은 유연 로봇 매니퓰레이터의 운동 방정식에 인가되며 끝점의 위치가 Feedback 되어 Servo 제어를 수행한다.

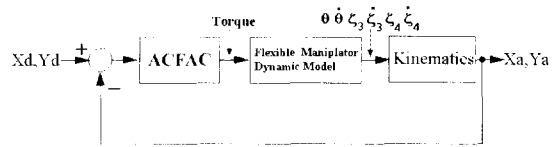


그림 2. 퍼지 제어기 구조

#### 3.1 신경회로망 학습 알고리즘에 기반한 퍼지 적응 시스템

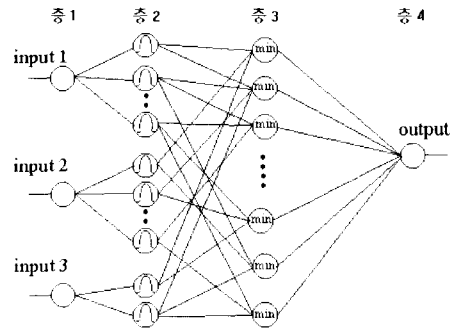


그림 3. 자동 구축 퍼지 제어기 구조

그림 3 은 제안된 신경회로망의 학습 알고리즘에 기반한 퍼지 적응 시스템을 보여 주고 있다. 각 층들이 수행하는 기능은 다음과 같다.

층 1 : 모든 노드들은 단지 입력 값을 다음 층으로 전달해 주는 역할만을 한다.

층 2 : 각 노드는 하나의 멤버십 함수로서 기능한다. 이것을 수식의 형태로 표현하면 다음과 같다.

$$f_{ij} = e^{-\frac{(x - m_{ij})^2}{2\sigma_{ij}^2}} \quad (10)$$

여기서  $m_{ij}$  와  $\sigma_{ij}$ 는 각각  $i$  번째 입력 변수인  $x_i$  의  $j$  번째 퍼지 함수의 중심 값과 분산이다.

층 3 : 이 층의 노드들은 T-norm 연산자의 기능을 수행한다. T-norm 연산자는 여러 종류가 있으나, 제안된 시스템에서는 Minimum 연산자를 사용하였다. 이 연산은 다른 것들에 비해 하드웨어 구현이 쉽다.

$$T(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x)) \quad (11)$$

T : T-norm 연산자

$\mu_A(x)$ 와  $\mu_B(x)$  : 퍼지 집합 A와 B 에서  $x$  가 갖는 멤버십 값들이다.

층 4 : 이 층의 노드들은 비퍼지화 기능을 수행한다. 비퍼지화 방법으로는 Yager의 레벨 집합 방법 [5] (Level-Set Method)을 사용한다. 이 방법은 무게중심 법에 비해 계산이 간단할 뿐 아니라 성능 면에서도 뒤지지 않는다는 것이 실험에 의해 증명되었다.

$$Z_c = \frac{\sum_{i=1}^r m_i \tau_i}{\sum_{i=1}^r \tau_i} \quad (12)$$

식에서  $Z_0$ 는 비퍼지화된 출력을,  $r$ 은 규칙의 수를, 그리고  $\tau_i$  는  $i$  번째 규칙의 Firing Strength를 의미한다. 출력부의 퍼지 집합이 대칭인 특징을 갖는다면,  $m_i$  는 멤버십 함수의 중심 값이 된다.

### 3.2 학습 알고리즘

두 번째 층에 있는 멤버십 함수들의 중심 값들을 결정하기 위해 경쟁 학습 알고리즘을 사용한다. 간단히 살펴보면 다음과 같다.

$$m_{i^*}(t+1) = m_{i^*}(t) + \alpha(t) [x(t) - m_{i^*}(t)], \quad i = i^* \\ m_i(t+1) = m_i(t), \quad i \neq i^* \quad (13)$$

$i^*$  : 승자 뉴런

$\alpha(t)$  : 학습률(Learning Rate)

각 멤버십 함수의 분산 값들을 N-Nearest-Neighbors 방법에 의해 결정한다. 최소화해야 할

목적 함수는 다음과 같다.

$$E = \frac{1}{2} \sum_{i=1}^N \left[ \sum_{j \in \text{Nearest}} \left( \frac{m_i - m_j}{\sigma_i} \right)^2 - \gamma \right]^2 \quad (14)$$

$\gamma$  : overlap 파라미터

가장 가까운 성분만을 고려하는 First-Nearest-Neighbor 방법을 사용하면  $\sigma_i$  는 다음과 같다.

$$\sigma_i = \frac{|m_i - m_{\text{closest}}|}{\gamma} \quad (15)$$

두 번째 층에 있는 모든 멤버십 함수의 파라미터들을 결정하고 나면 규칙 베이스의 입력부는 완성된 셈이다. 이제는 각 규칙의 입력부에 상응하는 출력부의 멤버십 함수를 결정해 주면 된다. 이 과정은 Grossberg의 Outstar 학습 알고리즘을 사용하여 쉽게 해결할 수 있다. Outstar 학습 알고리즘을 간단히 살펴보면 다음과 같다.

$$m_i^*(t+1) = m_i^*(t) + \alpha(t) [y(t) - m_i^*(t)], \quad i^* = \arg \max_{1 \leq i \leq n} u_i(t) \\ m_i^*(t+1) = m_i(t), \quad i \neq i^* \quad (16)$$

여기서  $y(t)$ 는  $u_i(t)$ 에 상응하는 출력을 의미하고  $n$  은 입력 벡터의 차원을 의미한다. 위와 같은 방법으로 퍼지 함수의 중심 값과 분산을 결정한다.

## 4. Flexible Manipulator 시뮬레이션과 실험 결과

본 논문에서 제안된 자동 구축 퍼지 제어기의 유용성을 증명하기 위해 Flexible Manipulator를 이용하여 시뮬레이션과 실험을 수행하였다. 시뮬레이션을 위해서 IBM 호환 컴퓨터를 사용하였고, 사용한 언어는 과학 계산 프로그래밍 언어인 MATLAB을 이용하였다.

### 4.1 시뮬레이션

Flexible Manipulator의 위치 제어를 위해 자동 구축 퍼지 제어기를 설계하고 시뮬레이션을 수행한 뒤 PID 제어기, 신경회로망 제어기와 성능을 비교하였다. 시뮬레이션에 사용한 경로는 반지름이 2m인 원모양의 경로이다. 끝점의 속도는 최대 속

도를 1m/s에서 3m/s로 늘려 가면서 실험하였다. Flexible Manipulator의 링크 재료는 알루미늄을 사용하였고, 길이는 2m이다. 끝점 무게 역시 0.3Kg에서 0.8Kg까지 가변 하면서 시뮬레이션하였다. 입출력 퍼지 함수 학습에 사용한 데이터는 퍼지 제어기를 이용해서 얻었다. 속도 1m/s 와 3m/s에서 각각 0.3Kg, 0.5Kg, 0.8Kg 의 부하를 가지고 시뮬레이션하여서 4428개의 입출력 데이터를 얻었다. 에러와 에러의 퍼지 함수를 적절하게 선정하기 위해 초기에는 10개로 학습하고 학습 후의 중심 값과 중심 값의 거리가 전체 범위의 5%미만의 성분이 있는 경우에는 멤버십 함수의 개수를 하나씩 줄여 나면서 성능을 검증하였다. 이에 따른 결과로 에러의 퍼지 함수는 6개, 에러 미분의 경우에는 7개, 속도의 퍼지 함수는 2개를 사용했다. PID제어기의 이득 성분의 값은 한계감도법을 사용하였고, 신경회로망 제어를 위해 학습 데이터로 자동 구축 퍼지 제어를 위해 만든 입출력 데이터를 사용했고, 학습과 제어에는 Back Propagation Network (BPN)를 이용했다. BPN의 입력은 에러와 에러의 미분 값을 사용했고 출력은 Torque 이다. Hidden Layer 의 Node 수는 10개이고, 10000회 Epoch에서의 이득 값을 사용했다. 속도에 따라 출력의 노드를 분리해서 Network을 구성했다. 학습 및 제어에 사용된 BPN을 그림 4에서 보여준다.

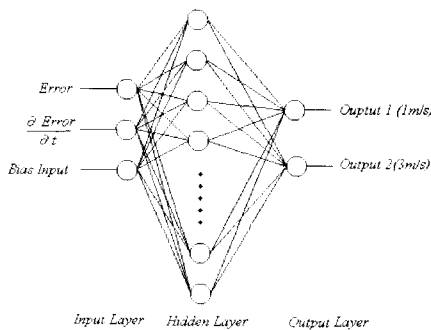


그림 4. 학습 및 제어에 사용한 BPN의 구조

시뮬레이션에 사용한 Flexible Manipulator의 동역학 운동 모델식은 식(8)을 사용하였다. 자동 구축 퍼지 제어기의 성능을 PID 제어기와 비교하기 위해 x축, y축의 위치 값과 위치 오차를 구하였다.

속도 1m/s, 3m/s 에서 0.3Kg, 0.5Kg, 0.8Kg 의 부하를 가지고 6가지 환경에서 성능을 비교했다. 속도가 1m/s 인 경우에는 원 궤적을 한 바퀴 도는 기준 궤적을 사용했고, 3m/s인 경우에는 두 바퀴를 회전하는 것을 기준 궤적으로 사용했다.

#### 4.2 시뮬레이션 결과

먼저 기준 궤적과 시뮬레이션 조건은 다음과 같다.

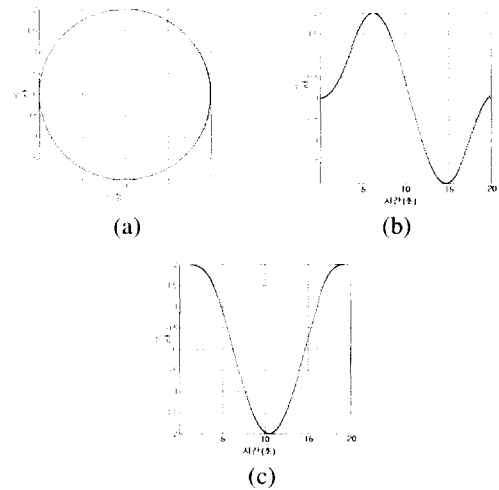


그림 5. (a) 기준 궤적, (b) x축 기준 궤적, (c) y축 기준 궤적 (X축, Y축 값 : meter)

- 시뮬레이션 조건

$$E : 7.17e-10 \text{ N/m}^2$$

$$I : 3.758e-10 \text{ m}^4$$

$$\rho : 7.706e-2 \text{ Kg/m}$$

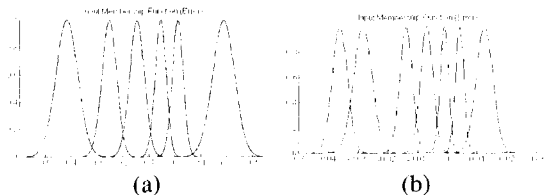
끝점 무게 : 0.3 Kg ~ 0.5 Kg

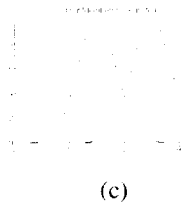
길이 : 2 m

재질 : 알루미늄

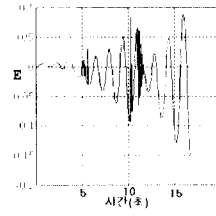
Sampling Time : 12.4 msec

학습에 의해 자동 구축된 입력의 멤버십 함수의 형태는 다음과 같다.





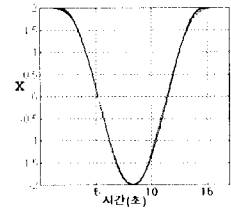
(c)



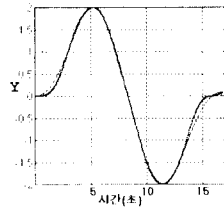
(c)

그림 6. (a) 에러의 퍼지 함수, (b) 에러 미분의 퍼지 함수, (c) 속도의 퍼지 함수

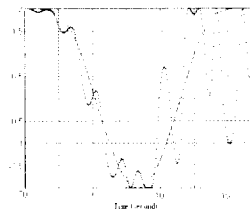
그림 8. 속도 1m/s, 0.3Kg 부하, 자동 구속 퍼지 제어 (단위: meter) (a) x축 추종 결과, (b) y축 추종 결과, (c) 추종 에러



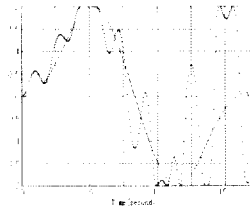
(a)



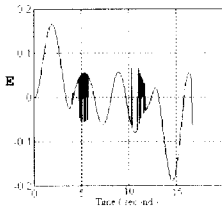
(b)



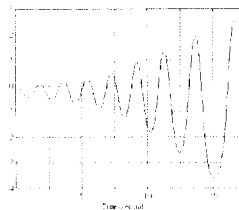
(a)



(b)



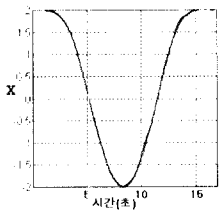
(c)



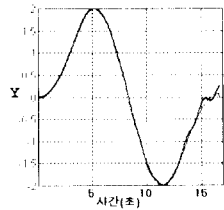
(c)

그림 7. 속도 1m/s, 0.3Kg 부하, PID 제어 (단위: meter) (a) x축 추종 결과, (b) y축 추종 결과, (c) 추종 에러

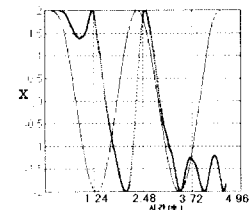
그림 9. 속도 1m/s, 0.3Kg 부하, 신경회로망 제어 (단위: meter) (a) x축 추종 결과, (b) y축 추종 결과, (c) 추종 에러



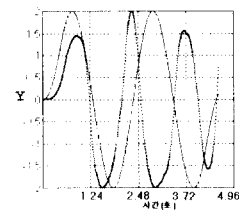
(a)



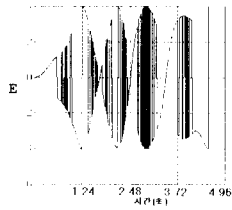
(b)



(a)

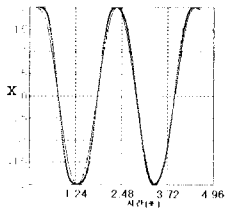


(b)

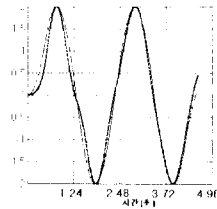


(c)

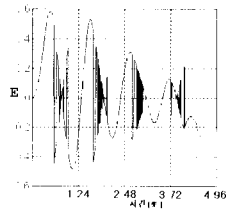
그림 10. 속도 3m/s, 0.3Kg 부하, PID 제어 (단위: meter) (a) x축 추종 결과, (b) y축 추종 결과, (c) 추종 에러



(a)

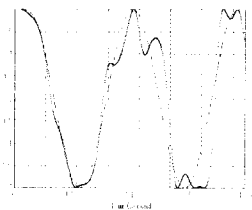


(b)

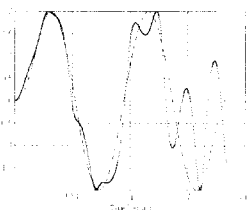


(c)

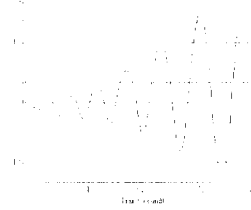
그림 11. 속도 3m/s, 0.3Kg 부하, 자동 구축 퍼지 제어 (단위: meter) (a) x축 추종 결과, (b) y축 추종 결과, (c) 추종 에러



(a)

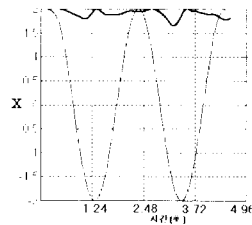


(b)

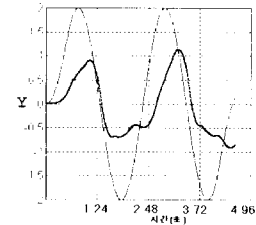


(c)

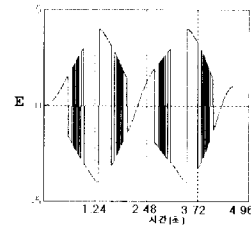
그림 12. 속도 3m/s, 0.3Kg 부하, 신경회로망 제어 (단위: meter) (a) x축 추종 결과, (b) y축 추종 결과, (c) 추종 에러



(a)

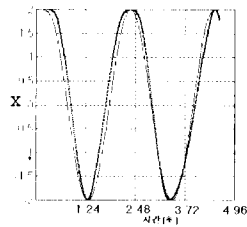


(b)

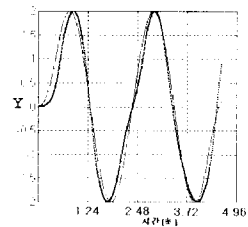


(c)

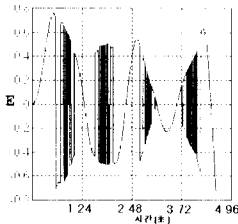
그림 13. 속도 3m/s, 0.8Kg 부하, PID 제어 (단위: meter) (a) x축 추종 결과, (b) y축 추종 결과, (c) 추종 에러



(a)

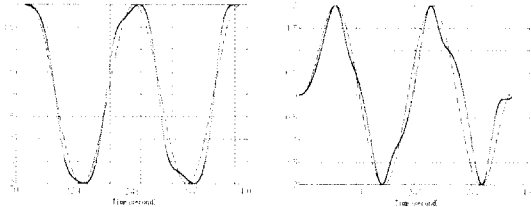


(b)



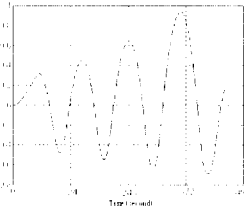
(c)

그림 14. 속도 3m/s, 0.8Kg 부하, 자동 구축 퍼지 제어 (단위: meter) (a) x축 추종 결과, (b) y축 추종 결과, (c) 추종 에러



(a)

(b)



(c)

그림 15. 속도 3m/s, 0.8Kg 부하, 신경회로망 제어 (단위: meter) (a) x축 추종 결과, (b) y축 추종 결과, (c) 추종 에러

### 4.3 실험 결과

시뮬레이션을 통해 자동 구축 퍼지 적응 제어기 성능이 이득 값이 고정되어 있는 PID 제어기와 신경회로망을 이용한 제어기에 비해 우수함을 보여주었다. 다음은 실제의 Flexible Manipulator를 이용한 실험에서도 자동 구축 퍼지 적응 제어기의 성능이 시뮬레이션 결과와 동일하게 우수함을 보이고자 한다. 그림 16은 Flexible Manipulator 실험의 전

체 블록도를 보여준다.

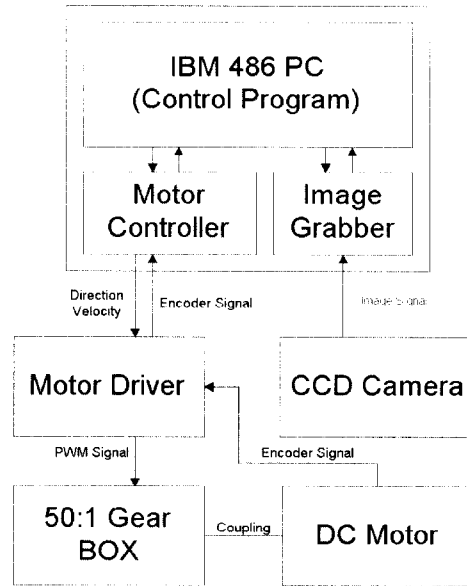


그림 16. Flexible Manipulator 제어기 전체 블록도

실험실의 크기와 카메라의 View Angle의 제약 때문에 시뮬레이션과 다르게 1.5m 인 알루미늄 Beam 을 사용하였고 기준 궤적은 원이 아닌 원 궤적 중 1/4을 왕복 운동하는 궤적으로 실험했다 (그림 17). NTSC 신호는 1초에 30 Frame의 화상을 Image Grabber로 전달하지만 본 논문에서는 샘플링 시간을 줄이기 위해 한 Frame을 홀수 Field와 짝수 Field로 나누고 각각을 따로따로 입력 신호로 사용하였다. 그리고 Frame 출력과 Frame 출력 사이의 출력값은 Interpolation에 의하여 샘플링 신호의 1/4 시간 (8.34msec) 만으로 가능하도록 하였다. 여기서 끝점을 찾기 위한 시간과 자동 구축 퍼지 제어기에서 제어 입력을 계산하는데 필요한 시간을 더해서 Sampling Time을 12.7 msec로 정하였다. Flexible Manipulator의 끝점 속도는 0.5m/s, 1m/s로 실험하였고, Flexible Manipulator의 끝점 부하는 무부하 시와 0.2Kg 부하를 이용해서 실험했다.

자동 구축 퍼지 제어기의 퍼지 함수 구축을 위한 학습 데이터는 퍼지 제어기를 이용해서 추출하였다. 0.5m/s, 1m/s 속도에서 무부하와 0.2Kg 부하



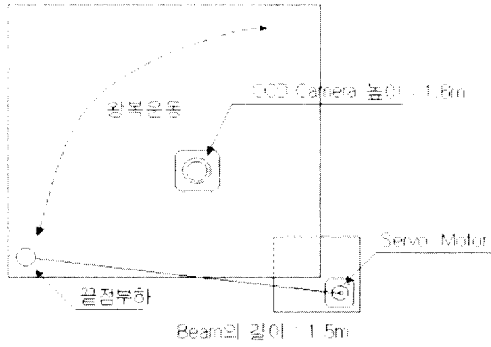


그림 17. 실험 장소의 구조 (천장에서부터의 View)

시 5번 실험을 하여 20,000개의 입출력 데이터를 획득하고 이를 이용해서 입출력 퍼지 함수를 구축했다. 입력의 퍼지 함수는 시뮬레이션의 경우와 동일하게 10개의 퍼지 함수부터 차례로 줄여 나가서 속도 2개, 에러 6개, 에러의 미분값 5개 퍼지 함수를 사용했다. 출력의 경우에는 PWM 레벨을 퍼지 함수로 사용했다. 실험 Data에 의한 학습에 의해 자동 구축된 입력의 멤버십 함수의 형태는 다음과 같다.

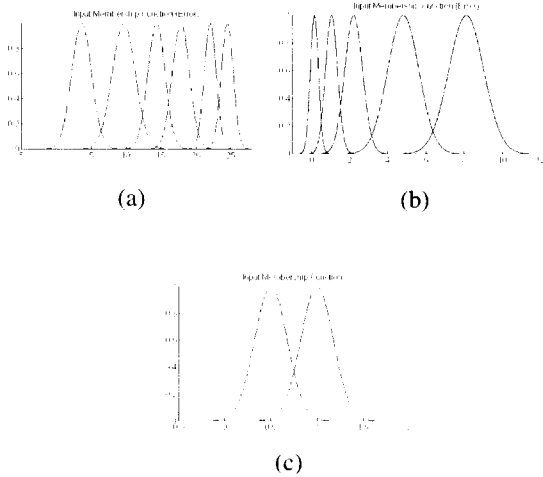


그림 18. (a) 에러의 퍼지 함수, (b) 에러 미분의 퍼지 함수, (c) 속도의 퍼지 함수

다음은 PID 제어와 자동 구축 퍼지 제어기를 이용한 결과들이다.

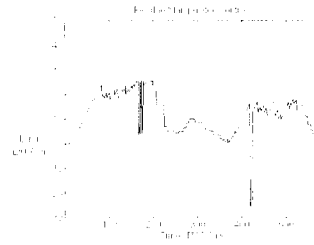


그림 19. PID 제어기를 이용한 실험 결과 (위치 오차) (1m/s, 무부하)

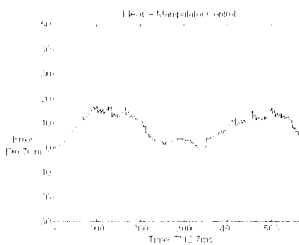


그림 20. 자동 구축 퍼지 제어기를 이용한 실험 결과 (위치 오차) (1m/s, 무부하)

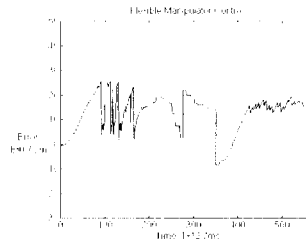


그림 21. PID 제어기를 이용한 실험 결과 (위치 오차) (1m/s, 0.2Kg 부하)

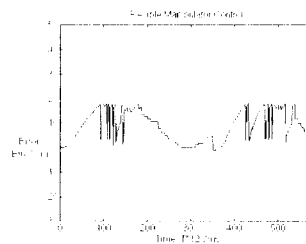


그림 22. 자동 구축 제어기를 이용한 실험 결과 (위치 오차) (1m/s, 0.2Kg 부하)

## 5. 결론 및 검토

본 논문의 목적은 연산 시간이 짧고 성능이 우수한 Flexible Manipulator 제어기 설계를 위해 정확한 모델식 없이 제어가 가능한 퍼지 계열의 제어기를 제안하였다. 시뮬레이션과 실험에 의해서 속도가 낮고 끝점의 부하가 적은 경우 PID제어와 자동 구축 퍼지 제어기는 성능의 차이를 보이지 않았지만, 속도가 빠르고 부하가 큰 경우에는 퍼지 제어기의 경우 모델이 변하더라도 에러에 따라 동적으로 대처하는 능력이 있기 때문에 비교적 끝점 부하가 큰 경우에도 기준 궤적을 추종함을 볼 수 있었다. 그러나, 퍼지 제어의 경우도 모든 면에서 우수한 것은 아니고 규칙을 추출하기가 쉽지 아니하고 우수한 성능을 보이기 위해서는 많은 실험 결과가 필요하다. 이런 경우 부하에 따라 퍼지 함수를 결정하기가 매우 힘든 문제이다. 실제로 본 논문에서도 고전적인 퍼지 제어기를 이용해서 성능 향상을 보고자 많은 노력을 하였지만 퍼지 함수의 조종이 쉽지 않았다. 그러나, 제안된 자동 구축 퍼지 제어기는 입출력 데이터로부터 적절한 퍼지 함수를 찾았고, 그 결과 기준 궤적을 우수하게 추종하는 것을 보여주었다. 또, 같은 지능 제어 계열의 신경회로망을 이용해서 시뮬레이션을 수행해보았다. 시뮬레이션에서 신경회로망 제어기는 무거운 부하에서는 어느 정도 기준 궤적을 추종하지만 가벼운 부하에서는 기준 궤적을 추종하지 못함을 알 수 있었다. 그러나, 제어하려고 하는 대상의 입출력 데이터를 뽑기 힘든 경우에는 적절하지 못하므로 입출력 데이터 없이 퍼지 함수를 자동 구축할 수 있는 방법을 찾아야 할 것이다.

실험에서는 Flexible Manipulator의 위치를 측정하는 센서로 CCD Camera를 사용했는데 여러 번의 실험 결과 몇 가지 부적합한 면이 나타났다. 무엇보다도 작업 공간이 카메라에 의해 매우 한정적이 된다는 점이다. 즉, 카메라 렌즈의 View Angle에 따라서 작업 공간의 크기가 한정될 뿐 아니라 빛의 양에도 매우 민감하다. 두 번째로, 데이터의 처리 속도 문제이다. NTSC 신호의 특성상 샘플링 시간을 줄이기가 어려웠다. 그러므로 실시간 제어에 어려움을 극복하기 위해서 향후에는 Strain Gauge를 이용한 위치 측정이 이루어져야 하겠다.

## 참고문헌

- [1] C.H. Lee, "Fuzzy Logic in Control Systems : Fuzzy Logic Controller Part I and Part II," IEEE Trans. on System, Man, and Cybernetics, Vol. 20, No. 2, pp. 404-435, March/April 1990.
- [2] A. Midha, A.G. Erdman, "Finite Element Approach to Mathematical Modeling of High Speed Elastic Linkages," J. of Mechanism and Machine Theory, pp. 603-618, Vol. 13, 1978.
- [3] H.I. Laursen, Structural Analysis, McGraw-Hill, New York, 1969.
- [4] A.J. Koivo, K. Lee, "Self-tuning Control of a Two-link Manipulator with a Flexible Forearm," Int. J. of Robotics Research, Vol. 11, No. 4, pp. 383-395, 1992.
- [5] M. Figueiredo, F. Gomide, A. Rocha and R. Yager, "Comparison of Yager's Level Set Methode for Fuzzy Logic Control with Mamdani's and Rasen's Method," IEEE Trans. Fuzzy System, Vol. 1, No. 2, May 1993.
- [6] 황인호, "신경회로망을 이용한 퍼지 시스템의 자동 구축", 홍익 대학교 석사 학위 논문, 1993.
- [7] 이기철, "멀티룰 베이스 구조의 퍼지 적응 제어기", 연세 대학교 석사 학위 논문, 1992.
- [8] 홍윤광, 조성원, 최경삼, "퍼지 교통 신호 제어에 관한 연구", 한국 퍼지 시스템 학회 춘계 학술 대회 논문집, 1994.
- [9] 이광형, 오길록 공저, "퍼지 이론 및 응용 1, 2", 홍릉 과학 출판사, 1992.
- [10] 이승준, 이기성, "유연 로봇 매니퓰레이터 퍼지 제어기 설계", 대한 전기 학회 하계 학술 대회, 1995.
- [11] Chin-Teng Lin and C. S. George Lee, "Neural-Network-Based Fuzzy Logic Control and Decision System," IEEE Trans. Computer, Vol. 40, No. 12, December, 1991.
- [12] P.B. Usoro, R. Nidira and S.S. Mahil, "A Finite Element/Lagrange Approach to Modeling Lightweight Flexible Manipulators," ASME J. of Dynamic Systems, Measurement, and Control, pp. 198-205, Vol. 108, 1986.
- [13] W.J. Book, "Recursive Lagrangian Dynamics of

Flexible Manipulator Arms." Int. J. of Robotics Research, pp. 87-101, Vol. 3, No. 3, Fall 1984.

[14] R.H. Cannon, E. Schmitz, "Initial Experiment on the End-point Control of a Flexible One-link Robot," Int. J. of Robotics Research, pp. 62-75, Vol. 3, No. 3, Fall 1984.

[15] S. Cetinkunt, B. Siciliano and W.J. Book, "Symbolic Modeling and Dynamic Analysis of Flexible Manipulator," 1986 IEEE Int. Conference on Robotics and Automation, 1986.

[16] G.G. Hasting, W.J. Book, "Experiment on the Control of Flexible Robot Arm," 1985 American Control Conference, pp. 728-729, Boston, MA, June 1985.

[17] K.H Low, "A Systematic Formulation of Dynamic

Equation for Robot Manipulators Elastic Links," J. of Robotics System, pp. 435-456, Vol. 4, No. 3, 1987.

[18] A.A. Goldenberg, F. Rakhsha, "Feedforward Control of a Single-link Flexible Robot," J. of Mechanism and Machine Theory, pp. 325-335, Vol. 21, No. 4, Pergamon Press, 1986.

[19] Y. Sakawa, F. Matsuo, S. Fukusima, "Modeling and Feedback Control of Flexible Arm," J. of Robotics Systems, pp. 453-472, Vol. 2, No. 4, 1985.

[20] D.R. Meldrum, M.J. Balas, "Application of Model Reference Adaptive Control to a Flexible Remote Manipulator Arm," Proc. 1986 ACC, 1986.

[21] B. Siciliano, W.J. Book, B-S Yuan, "Model Reference Control of an One Link Flexible Arm," Proc. 1986 IEEE CDC, Athens, Greece, December



**이기성 (Keeseong Lee)**

1980년 : 서울대학교 전자공학과 졸업 (공학사)  
 1983년 : 미국 Ohio 주립대학교 전기공학과 졸업(공학석사)  
 1989년 : 미국 Purdue 대학교 전기공학과 졸업(공학박사)  
 1991년-현재 : 홍익대 공대 전자전기공학부 조교수

주관심분야 : 퍼지시스템, 영상처리 및 인식, 로봇틱스 등.



**조현철 (Hyun-Chul Cho)**

1985년 : 동아대학교 전기공학과 졸업 (공학사)  
 1990년 : 동아대학교 전기공학과 졸업 (공학석사)  
 1997년 : 동아대학교 전기공학과 졸업 (공학박사)  
 1993년-현재 : 경북전문대학 전자과 조교수

주관심분야 : 신경회로망, 퍼지제어, 지능제어, 로봇틱스, 컴퓨터비전 등