

LSE 학습법을 이용한 퍼지제어기 설계와 도립진자의 제어

김 근기*, 유 창완, 임 화영
 광운대학교 제어계측공학과

Using Least-Square Learning Method design Fuzzy Controller and control Inverted Pendulum

Kim Kuen-Ki*, Ryu Chang-Wan, Yim Wha-Yeong
 Dep. of Control & Instrumentation Eng. KwangWoon Univ.

Abstract - Design of Fuzzy cotroller consists of intuition of human expert, and any other information about how to control system, they translated into a set of rules. If the rules adequately control the system, the design work is done well. If the rules are inadequate, the designer must modify the rules. Through this procedure, the system can be controlled. In this paper, we designed simply a fuzzy controller based on human knowledge, but it has errors showing some vibrations. So we updated the optimal parameters of fuzzy controller using Recursive least square algorithm.

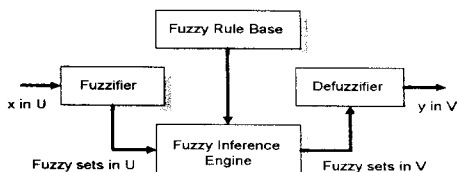
1. 서 론

기존의 퍼지 제어기는 일반적으로 모델이 복잡하거나 비선형등으로 복잡한 시스템의 경험이나 직관등에 의해 규칙을 언어 변수로 표현할 수 있으며 제어 대상 시스템의 수학적인 모델링이 필요하지 않는다는 이유로 간단히 적용할 수 있는 장점이 있다. 반면에 비선형성이 강한 도립진자를 제어하는데 시간이 너무 오래 걸린다는 단점이 있다. 그래서 본 논문에서는 이 퍼지 제어기의 결론부 파라미터를 LSE(최소자승법)를 통하여 최적화 시켰다. LSE(Least Squares Estimator)는 시스템 출력값 오차의 자승을 계산하여 가장 적절한 출력값, 즉, 가장 이상적인 제어 입력을 만들어 내는 학습법이다. 이 최적화를 통해 보다 쉽고 빠르게 도립진자를 제어하는 제어기를 설계할 수 있었다.

2. 본 론

2.1 퍼지 시스템

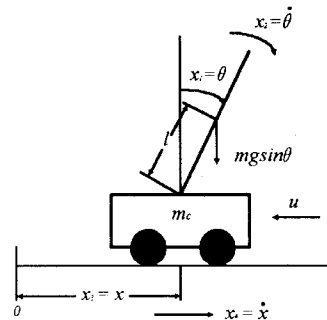
퍼지 시스템은 <그림 1>에서 보는 바와 같이 정량적인 입력을 퍼지화하는 퍼지화기(Fuzzifier), 퍼지값을 추론하는 추론 엔진(Inference Engine), 추론 과정에서 이용되는 규칙 기반(Rule Base), 규칙 기반을 구성하는 IF-THEN 규칙, 퍼지값을 다시 정량적인 값으로 바꾸어주는 비퍼지화기(Defuzzifier)로 나누어진다. 퍼지화기, 추론 엔진 비퍼지화기는 여러 가지가 제시되어 있지만 Singleton 퍼지화기, Product 추론 엔진, Center Average 비퍼지화기가 보편적이다.



<그림 1.> 퍼지 시스템

2.2 도립진자 시스템

본 논문의 실험대상인 도립 진자의 제어목적은 진자를 도립된 상태로 원하는 위치에 놓이도록 하는 것이다. 도립 진자 제어 시스템은 비선형 시스템의 전형적인 예로서 제어 이론의 성능을 입증하는 플랫폼으로 많이 이용되고 있다.



<그림 2.> 도립 진자 시스템

시스템의 상태 방정식은 다음과 같다.

$$\dot{x}_1 = x_2 \tag{1}$$

$$\dot{x}_2 = \frac{g \sin x_1 - \frac{m l x_2^2 \sin x_1 \cos x_1}{m_c + m} - \frac{\cos x_1}{m_c + m}}{l \left(\frac{4}{3} - \frac{m \cos^2 x_1}{m_c + m} \right)} - \frac{\cos x_1}{m_c + m} \cdot u \tag{2}$$

$$\dot{x}_3 = x_4 \tag{3}$$

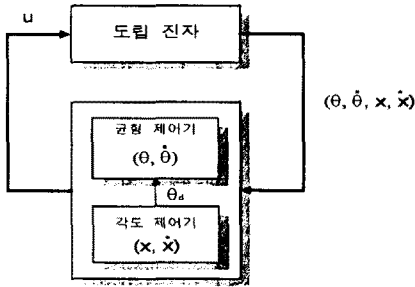
$$\dot{x}_4 = \frac{m l x_2^2 \sin x_1 - m l x_2 \cos x_1 + \frac{1}{m_c + m}}{m_c + m} \cdot u \tag{4}$$

x_1	진자의 각도	θ degree
x_2	진자의 각속도	$\dot{\theta}$ degree/sec
y_1	수레의 거리	x cm
y_2	수레의 속도	\dot{x} cm/sec
u	제어 입력	Newton
m	진자의 질량	0.1 kg
g	중력 가속도	9.81 m/s ²
m_c	수레의 질량	1 kg
l	진자 길이의 반	0.5 m

<표 1.> 상태 변수와 시스템 파라미터

2.3 퍼지 제어기 설계

본 논문에서 제안한 퍼지 제어기는 균형 제어기 (Balancing Controller)와 각도 제어기(Angle Controller)로 구성되어 있다. 각도 제어기는 수레의 위치와 속도를 입력으로 받아 수레를 원하는 위치로 움직이도록 하는 기준 각도(θ_d)를 출력한다. 균형 제어기는 현재의 각도에서 기준 각도를 뺀 차이와 각속도를 입력으로 받아 진자의 각도 오차를 0으로 만드는 출력 u 를 만든다. 전체 블록도는 다음과 같다.



<그림 3.> 퍼지 제어 시스템

2.3.1 IF THEN 규칙 베이스

퍼지 제어 시스템을 구성하기 위해서는 우선 상태변수와 출력 변수를 정해야 하는데 상태 변수로는 진자의 각도(θ)와 각속도($\dot{\theta}$), 수레의 원점으로부터의 거리(x)와 속도(\dot{x})로 정하였고, 출력 변수는 프로세스로 들어가는 입력 u 로 정하였다. 정의된 입력 변수는 샘플링 시간마다 퍼지 규칙에 의하여 제어기의 출력을 결정하게 된다. 균형 제어기와 각도 제어기에 대한 IF-THEN 규칙은 다음과 같다.

IF x_1 is A_1' and x_2 is A_2' , THEN u is \bar{y}_1'

IF x_3 is C_1' and x_4 is C_2' , THEN θ_d is \bar{y}_2'

$A_1', A_2', \bar{y}_1', C_1', C_2', \bar{y}_2'$ 는 각각에 대한 퍼지 집합이다. 규칙을 나타내면 다음과 같다.

$\theta, \dot{\theta}$	NE	PO	NE	ZE
	ZE	NE	ZE	PO
	PO	ZE	PO	NE
		NE	ZE	PO
		$\theta, \dot{\theta}$		

<표 2.> IF-THEN 규칙 테이블

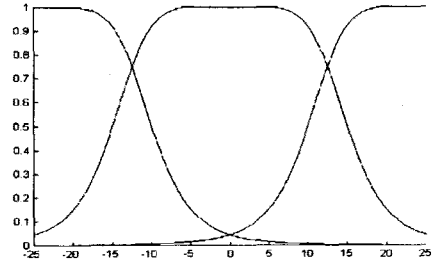
2.3.2 소속 함수 (Membership Function)

소속 함수는 여러 함수가 이용되고 있다. 삼각 함수, 사다리꼴 함수, 가우스 함수, 종형 함수 등이 있으나 제어 대상이 비선형성이 강하므로 이를 보완할 수 있는 비선형 함수인 종형 함수로 정하였다. 종형 함수는 3가지 파라미터(a, b, c)를 포함하며 이 파라미터값에 따라 다른

형태의 모양이 된다.

$$\mu(x) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (5)$$

각각의 상태 변수에 대한 조건부 소속 함수는 3부분으로 나누어 <그림 4.>와 같이 정의하였다.



<그림 4.> 종(bell)형 소속 함수

2.4 LSE Algorithm

본 논문에서는 앞에서 설계한 퍼지 제어기의 결론부 파라미터 \bar{y}' 을 LSE로 학습시켜서 보다 나은 성능을 연구하였다.

제어 입력 u 로부터 결론부 파라미터 \bar{y}' 을 다음과 같이 학습시켰다.

$$J_p = \sum_{j=1}^p [f(x_0^j) - y_0^j]^2 \quad (6)$$

$$f(x) = \frac{\sum_{l=1}^N \Lambda \sum_{l'=1}^N \bar{y}'^{l \wedge l'} \left[\prod_{i=1}^4 \mu_{A_i'}(x, i) \right]}{\sum_{l=1}^N \Lambda \sum_{l'=1}^N \left[\prod_{i=1}^4 \mu_{A_i'}(x, i) \right]} \quad (7)$$

$$\theta = (\bar{y}'^{1 \wedge 1}, K, \bar{y}'^{N1 \wedge 1}, \bar{y}'^{12 \wedge 1}, K, \bar{y}'^{N12 \wedge 1}, K, \bar{y}'^{1N2 \wedge N}, K, \bar{y}'^{N1N2 \wedge N})^T \quad (8)$$

연쇄법칙으로부터 오차를 찾아내어 다음의 순환식으로 퍼지 제어기를 학습시켜 최적화된 결론부 파라미터를 찾아서 오버슈트와 정정시간이 줄어들 수 있었다.

$$\frac{\partial E}{\partial (\bar{y}')} = \frac{\partial E}{\partial x_1} \frac{\partial x_1}{\partial u(k)} \frac{\partial u(k)}{\partial (\bar{y}')} \quad (9)$$

$$\theta(p) = \theta(p-1) + K(p) [y_0^p - b^T(x_0^p) \theta(p-1)] \quad (10)$$

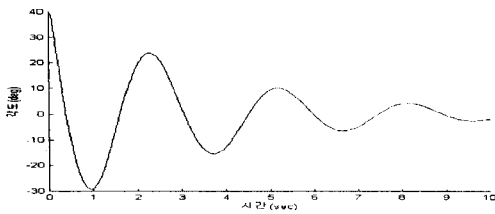
$$K(p) = P(p-1) b(x_0^p) [b^T(x_0^p) P(p-1) b(x_0^p) + 1]^{-1} \quad (11)$$

$$P(p) = P(p-1) - P(p-1) b(x_0^p) [b^T(x_0^p) P(p-1) b(x_0^p) + 1]^{-1} b^T(x_0^p) P(p-1) \quad (12)$$

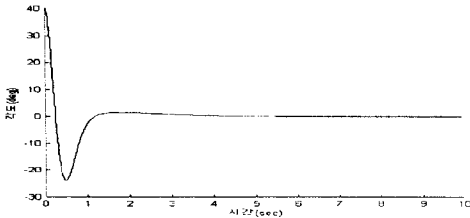
2.5 시뮬레이션 결과

시뮬레이션 결과는 다음과 같다.

그림 5와 그림 6은 [40°, 0 cm]의 초기값에서 실행한 각도에 대한 결과이다.

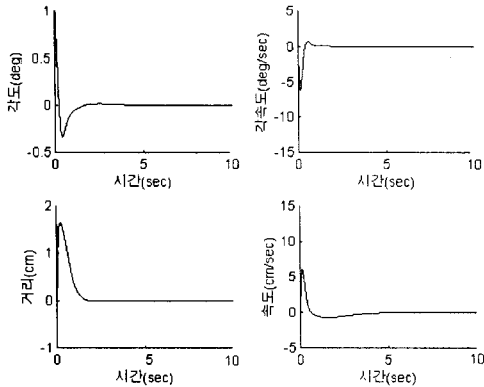


〈그림 5〉 학습전

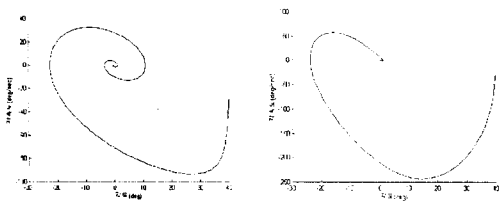


〈그림 6〉 학습후

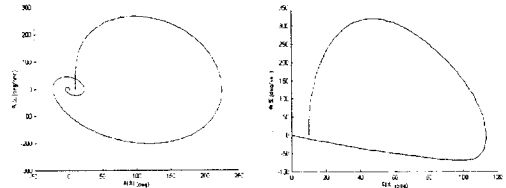
위의 그림에서 볼 수 있듯이 학습후의 settling time과 overshoot가 현저히 줄어들 것을 확인할 수가 있다. 다음에 전체 시뮬레이션을 나타내었다.



〈그림 7〉 시뮬레이션 결과 (각도, 각속도, 거리, 속도)



〈그림 8〉 각도와 각속도 - 학습전(左), 학습후(右)



〈그림 9〉 거리와 속도 - 학습전(左), 학습후(右)

구분	오버슈트[°]	정정시간[sec]
학습전	15	12
학습후	7	1.8
비교	8°차이	10.2초 차이

〈표〉 각도에 대한 학습전후 비교(초기값[10°, 10])

구분	오버슈트[cm]	정정시간[sec]
학습전	25	7
학습후	11	1.2
비교	14cm차이	5.8초 차이

〈표〉 거리에 대한 학습전후 비교(초기값[10°, 10])

3. 결 론

기존의 퍼지 제어기는 조건부 소속함수와 결론부 파라미터를 설정하는 것이 쉽지가 않다. 설정을 하여도 실험 과정에서 확인한 것처럼 많은 진동이 있고 오버슈트도 컸다. 그래서 본 논문에서는 LSE 학습법을 적용하여 결론부 파라미터 y_j 값을 학습시켜서 보다 빠르게 독립진자를 세우는데 성공하여 퍼지 제어를 보다 안정화시켰다. 다음에는 조건부 소속함수도 학습하도록 연구하여 더 좋은 성능을 얻도록 연구할 것이다.

(참 고 문 헌)

- [1] Li-Xin Wang, "A Course in Fuzzy and Control", Prentice-Hall, 1997.
- [2] Lefteri H. Tsoukalas, Robert E. Uhrig, "Fuzzy and Neural Approaches in Engineering", John Wiley & Sons INC., 1996
- [3] Hagan, Demuth, Beale, "Neural Network Design", PWS, 1995
- [4] D. Driankov, H. Hellendoorn, M. Reinfrank, "An Introduction to Fuzzy Control", Springer, 1993
- [5] M. Beale, H. Demuth, "Fuzzy Systems Toolbox for Use with MATLAB", PWS, 1994
- [6] J.-S.R. Jang, C.-T. Sun, E. Mizutani, "Neuro-Fuzzy and Soft Computing", Prentice-Hall, 1997