

퍼지제어기의 멤버쉽함수의 튜닝 방법

(A Tuning Method for the Membership Functions of a Fuzzy Controller)

李志弘*, 蔡錫*, 吳映碩*

(Ji hong Lee, Seog Chae and Young seok Oh)

要約

퍼지제어기의 성능은 퍼지화 방법, 추론규칙, 비퍼지화 방법, 그리고 언어적으로 묘사되는 각 변수의 모양에 관계한다. 이 중에서 언어 변수와 제어규칙 이외의 요소는 보통 퍼지제어기의 설계자인 엔지니어에 의해 결정되나, 언어변수와 제어규칙은 대상 시스템에 대해서 잘 알고 있는 전문가에 의해서 제어기 설계자에게 전달된다. 이 전달과정에서 불분명한 부분이나 빠뜨린 부분에 대해서 언어변수의 모양에 제한하여 보완하는 방법을 제시한다. 일차적으로 퍼지제어기내의 모든 언어정보를 간단한 형태로 정의하고 이 언어 정보의 형태와 제어기 성능과의 연관관계를 구하여 시스템의 운전을 통하여 최초로 전달된 언어정보를 튜닝하는 방법을 제시한다. 제시된 방법은 type 1, 2 시스템에 적용되어 유용성을 보인다.

Abstract

It is known that the performance of a fuzzy controller is related with fuzzification method, inference rules, defuzzification method, and linguistic variables. Among these, generally, the linguistic variables and control rules are transferred to control engineers from an expert or experts of the controlled system, and other parts are designed by control engineers. However, there may be some missed infirmations or uncertainties in the transferred data. The purpose of the paper is to propose an algorithm to tune the membership functions of initially given fuzzy sets. To do so, a simple shape of the membership function is assumed for the fuzzy sets, and the relations between the shapes of the fuzzy sets and the performance of the control system is derived. According to the relations, the shape of the membership functions are modified during operation of the whole system. The proposed algorithm will be applied to two example plants, type 1 and type 0 systems.

1. 서론

*正會員, 金烏工科大学 電子制御工學科, 助教授
(Dept. of cont. Eng., Kumoh Nat'l Inst.
of Tech. Univ.)

接受日字: 1992年 7月 23日

Zadeh^[1,2]에 의해 퍼지이론이 제어에 도입되면서 퍼지제어에 관한 많은 이론적 진전이^[3,4,5] 있었으며 또한 많은 성공적인 응용사례가 보고되고있다. 이러한 퍼지제어기의 성능은 퍼지화 방법, 추론규칙, 비퍼

지화 방법. 그리고 언어적으로 묘사되는 각 변수의 모양에 관계한다. 이중에서 언어 변수와 제어규칙 이외의 요소는 보통 퍼지제어기의 설계자인 엔지니어에 의해 결정되나, 언어변수와 제어규칙은 대상 시스템에 대해서 잘 알고 있는 전문가에 의해서 제어기 설계자에게 전달된다. 이 전달과정은 사실 매우 복잡한 과정으로 제어기 설계에 대해서 잘 모르는 운전자가 얼마나 자신의 지식을 잘 표현해 주는가와 제어기 설계자가 얼마나 잘 그 지식들을 컴퓨터가 처리할 수 있는 형태로 변환하는가가 구성될 퍼지제어기의 성능과 밀접한 관계가있다. 물론 기타 퍼지화 방법, 추론 방법, 그리고 비퍼지화 방법 등에 의해서도 제어기의 성능이 좌우되겠지만, 이들에 대해 지금까지 제시된 여러 방법들이 이론적으로 성능비교가 되어있기보다는 경험적, 실험적 또는 직관적인 비교들만이 되어있는 실정이고 이부분은 숙련 운전자가 결정하기 보다는 제어기 설계자가 결정해야하는 부분으로 아직도 연구가 진행중인 상태에있다. 이러한 전달과정에는 필연적으로 불분명한 부분이나 빠뜨린 부분이 있기 마련인데, 이 부분의 보완을 위해서 많은 시행착오를 거쳐 최적의 언어변수의 묘사와 제어규칙을 찾게 된다. 이러한 불필요한 과정을 줄이기 위해서는 전달된 정보의 전부 또는 일부를 자동적으로 튜닝하는 방법이 필요하게 된다.

또한 전달된 운전자의 지식중에는 퍼지제어기에 맞추기 위한 난이도가 다를 수있다. 일단 제어규칙 자체는 숙련 운전자가 표현 하기에 그리 어려움이 없다고 예상할 수 있는데 그 이유는 그가 실제로 그러한 규칙을 마음에 두고 운전을 하기 때문이다. 그러나 그 규칙 안에 들어있는 여러 언어변수의 경우에는 상황이 좀 다르다. 예를 들어 "크다"라는 언어변수가 숙련 운전자에 의해 표현 되었을때 이 언어변수를 퍼지집합화하는 즉, 이 변수의 멤버쉽 함수를 결정하는 문제를 생각해보자. 일단 숙련운전자가 이러한 단어를 사용하면 제어기설계자는 "얼마정도가 '크다'인가?"하고 질문하게 된다. 그러면 숙련 운전자는 "5 정도면 '크다'고 볼 수있다"고 대답한다면 제어기 설계자는 '크다라는 언어변수의 멤버쉽함수의 중심값(center)을 결정하게 된다. 그리고 완전한 멤버쉽함수의 모양을 결정하기위해 "어느정도의 범위가 전문가가 '크다'에 속하는가?"라는 질문으로 이 퍼지집합의 support를 구하려하면 이에 대한 대답은 쉽지않으리라는 것을 예상할 수있다. 더구나 support내의 값에 대한 소속정도(degree of membership)을 알기 위해 추가적으로 "6 정도면 어느 정도 '크다'고 할 수 있는가?"하고 질문하면 이에 대해 숙련운전자가 숫자

로 대답하기는 더욱 어려울 것을 예상할 수 있다. 즉, 아무리 숙련된 운전자를 확보하고있다고 하더라도 그 지식을 퍼지제어기로 옮기는 과정에는 확정짓기 곤란한 부분이 존재한다는 것이다.

이러한 문제에 대해 일단 분명히 전달된 정보만으로 퍼지제어기를 구성한 후, 이 제어기가 숙련 운전자의 조작과 같아지도록 불분명한 부분을 확정지어 나가는 방법이 있으나 이 방법은 그 해가 유일하게 존재하지 않을 수도있다는 문제점과 교정에 사용된 데이터를 제공한 숙련조작자의 상태에 크게 좌우될 수 있다는 문제점이 지적될 수 있다.

본 논문에서는 시스템의 운전을 통하여 최초로 전달된 언어정보를 튜닝하는 방법을 제시한다. 일반적으로 퍼지변수의 멤버쉽함수는 임의의 모양을 가질 수있으나 이러한 임의의 모양을 알아낸다고 하는 것은 현재로는 그리 쉬운일이 아니다. 그러므로 본 논문에서는 각 멤버쉽함수의 중심값과 support만 고려한 삼각형꼴의 멤버쉽함수를 가정하고 제어대상 시스템의 운전을 통해 점차적으로 튜닝해가는 방법을 제안한다. 이를 위해 일차적으로 퍼지제어기내의 모든 언어정보를 간단한 삼각형꼴로 가정하고 이 각 삼각형의 세 꼭지점과 제어시스템의 성능과 연관시켜 제어 대상 시스템을 운전해가면서 제어기의 성능이 개선되는 방향으로 각 삼각형의 꼭지점의 위치를 변경시켜나가는 일종의 학습방법을 제시한다.

이어 2장에서는 퍼지제어기의 구조와 튜닝 방법에 대해 설명하고 3장에서는 시뮬레이션 결과와 검토를, 4장에서는 결론을 맺도록하겠다.

II. 멤버쉽함수의 튜닝

이 장에서는 사용될 퍼지제어기의 구조와 튜닝 알고리즘에 대해 설명한다.

A 모델

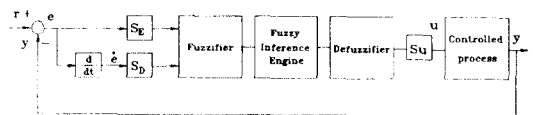


그림 1. 전형적인 퍼지제어기의 예

Fig. 1. A typical fuzzy feedback controller.

그림 1로 표현되는 전형적인 퍼지제어기는 크게 퍼지화부, 추론부, 그리고 비퍼지화부로 구성된다. 본 논문에서는 Mamdani의 직접추론법과 간단화된 무게중심법에 의한 비퍼지화 방법을 채택했다. [8] 즉 다

음과 같은 두개의 규칙이 주어져 있다고 가정하면

- $R_1 : \text{if } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } z \text{ is } C_1.$
- $R_2 : \text{if } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ then } z \text{ is } C_2.$

R_1 에 의해 rule feasibility w_1 과 자체 무게중심 u_1 이 구해지고 R_2 에 의해 같은 의미의 w_2 와 u_2 가 구해진다. 그러면 최종적으로 제어기의 출력은

$$u_0 = \frac{w_1 u_1 + w_2 u_2}{w_1 + w_2} \tag{1}$$

로 구해지며 N_U 개의 제어 규칙에 대해서는

$$u_0 = \frac{\sum_{k=1}^{N_U} w_k u_k}{\sum_{k=1}^{N_U} w_k} \tag{2}$$

로 확장된다.

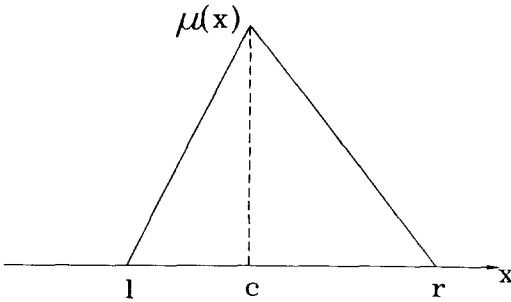


그림 2. 삼각형꼴의 멤버십함수

Fig. 2. A triangular-shaped membership function

퍼지제어기 내부의 퍼지집합은 높이가 1인 삼각형으로 가정한다. 그러면 각 퍼지집합은 세꼭지점의 x 좌표에 의해서 결정된다. 이러한 모양을 나타내는 세 변수 l(left), c(center), r(right)로 그림 2처럼 표시되고 이는 다음의 수식으로 정의한다.

$$\mu(x) = t(l, c, r, x) \tag{3}$$

B. 튜닝 알고리즘

이제 그림 1의 제어기에 대해 각 변수의 멤버십 함수를 다음으로 표시하고 각 멤버십함수의 계수 l, c, r 이 시스템성과 어떻게 연관되는지 살펴보자.

$$\mu_{E_i}(e) = t_E(l_{E_i}, c_{E_i}, r_{E_i}, e), \quad i = 1, \dots, N_E \tag{4}$$

$$\mu_{D_j}(\dot{e}) = t_D(l_{D_j}, c_{D_j}, r_{D_j}, \dot{e}), \quad j = 1, \dots, N_D \tag{5}$$

$$\mu_{U_k}(u) = t_U(l_{U_k}, c_{U_k}, r_{U_k}, u), \quad k = 1, \dots, N_U \tag{6}$$

우선 제어시스템의 제어성능을 매 순간

$$E(t) = \frac{1}{2}(r(t) - y(t))^2 \tag{7}$$

로한다. 이 성능지수에 대해 대표로 l_{E_i} 에 대해 그 영향을 살펴본다. 표기의 편의상 시간 t 를 생략하고 (7)을 l_{E_i} 로 편미분한다.

$$\frac{\partial E}{\partial l_{E_i}} = -(r-y) \frac{\partial y}{\partial l_{E_i}} = -(r-y) \frac{\partial y}{\partial u_0} \frac{\partial u_0}{\partial l_{E_i}} \tag{8}$$

(8)식의 마지막 항은 다음으로 전개되는데

$$\frac{\partial u_0}{\partial l_{E_i}} = \sum_{k=1}^{N_U} \frac{\partial u_0}{\partial u_k} \frac{\partial u_k}{\partial l_{E_i}} \tag{9}$$

여기서 k는 규칙의 순서를 의미하고 따라서 N_U 는 전체 규칙의 갯수를 의미한다. (2)식을 참고하면 (9)의 $\partial u_0 / \partial u_k$ 는

$$\frac{\partial u_0}{\partial u_k} = \frac{w_k}{\sum_{k=1}^{N_U} w_k} \tag{10}$$

가 되므로 결국 (9)식은

$$\frac{\partial u_0}{\partial l_{E_i}} = \frac{\sum_{k=1}^{N_U} w_k (\partial u_k / \partial l_{E_i})}{\sum_{k=1}^{N_U} w_k} \tag{11}$$

가 된다. 여기서 다음의 δ 를 정의하면

$$\delta = (r-y) \frac{\partial y}{\partial u_0} \tag{12}$$

(8)식은 다음으로 정리된다.

$$\frac{\partial E}{\partial l_{E_i}} = -\delta \frac{\sum_{k=1}^{N_U} w_k (\partial u_k / \partial l_{E_i})}{\sum_{k=1}^{N_U} w_k} \tag{13}$$

그리고 l_{E_i} 의 학습규칙은

$$\Delta l_{E_k} = \delta \frac{\sum_{k=1}^{N_k} w_k \frac{\partial u_k}{\partial l_{E_k}}}{\sum_{k=1}^{N_k} w_k} \quad (14)$$

를 정의하여

$$l_{E_k}(n+1) = l_{E_k}(n) + \eta_k \Delta l_{E_k} \quad (15)$$

로한다. 전체적으로 각 $l_k, c_k, r_k, i=1, \dots, N_k, c_0, r_0, j=1, \dots, N_0$ 와 $l_k, c_k, r_k, k=1, \dots, N_k$ 에 대해서도 같은 방법으로 조정규칙이 유도된다. 이렇게 유도된 조정규칙으로 제어대상 시스템을 운전하면서 퍼지제어기의 언어변수들의 모양을 변화시켜나가면서 (7)로 정의된 제어성능을 개선시켜나간다.

여기서 언급하고 넘어가야 할 부분은 $\partial y / \partial u_0$ 와 $\partial u_k / \partial l_k$ 의 계산에 관한 문제이다. 일반적으로 $\partial y / \partial u_0$ 는 시스템의 다이나믹스에 해당하는 항으로 시스템내의 모든 상태변수와 입력들의 함수이다. 그런데 시스템의 상태변수를 모두 측정가능하지 않을 수도 있고 계산자체를 학습과 병행할 수가없다. 그러므로 어떤 형태로든 근사화의 방법이 사용되어야하는데 그림 1의 제어기의 구조를 고려하여보면 y 와 \dot{y} 와 u 를 어느 일정한 간격으로 나누어서 디스크리타이즈시킨 점, 즉 각 (y, \dot{y}, u) 의 값에 대해 사전에 시스템을운전시킨 데이터로부터 테이블을 만들어놓고나서 학습과정에 측정된 (y, \dot{y}, u) 의 값에 대해 이 테이블로부터 interpolation하여 $\partial y / \partial u_0$ 의 값을 계산해내는 방법과 대상 시스템의 숙련조작자로부터의 경험적지식을 토대로 근사 모델을 구해 사용하는 방법등이 가능하다. 그러나 전자의 방법은 모든 시스템이 사전에 운전시켜보는 일이 가능하지는 않다는 점에서 약간의 약점이있다고 할 수있으나 본 논문에서는 사전 운전이 가능한 시스템을 가정하여 전자의 방법을 택하였다. 다음으로 $\partial u_k / \partial l_k$ 의 계산에 대해 생각해본다. 이항은 제어대상 시스템에 관계가없고 퍼지제어기에 관계되는 항이므로 채택된 퍼지화 방법, 추론 방법, 비퍼지화 방법등에 의해서실 복잡하기는하지만 경우에 따라 해석적인 계산이 가능하기는 하다.

그러나 그 개념에 비해 수식이 매우 복잡하고 간단히 근사화가 가능하므로 본 논문에서는 근사 계산 방법을 쓰기로 한다. 즉 현재의 l_{E_k} 값에 의해 u_k 의 값을 구하고 다른 계수들은 모두 고정시키고 l_{E_k} 를 Δl_{E_k} 만큼 변화시켜 u_k 를 구해 아까의 값과 비교하여 Δu_k 를 구하여 $\partial u_k / \partial l_k$ 를 $\Delta u_k / \Delta l_{E_k}$ 로 근사화시켜 사용할 수있다. 다른 계수들에 대해서도 같은 방법으로 근사

값을 구하여 대치할 수있다. 그런데 이상의 방법에 의해 각 멤버쉽함수를 조정하면 측정된 오차나 오차의 변화율이 갖는 전공간에 빈틈없이 퍼지집합을 배치하지 못할 수도있게 된다. 그래서 여기서는 하나의 퍼지집합의 최대 멤버쉽을 주는 값과 좌우의 두 개의 퍼지집합의 소속정도가 0이되는 값이 같도록 재조정하는 과정을 추가하여 어느 값에서도 두개의 퍼지집합의 영향을 받도록한다.

이상의 방법에 의해 (15)식과 같은 방법으로매 제어 순간 또는 일정한 간격으로 구해 각퍼지집합의 계수를 튜닝해나가게된다.

III. 적용 예

2장에서 제안된 방법을 type 1, type 0시스템에 적용하여 본다

A. type 1 시스템의 적용

type 1 시스템은 step 입력에 대해 정상 상태오차가 없다고 알려져있으며 여기서 적용된 시스템의 전달함수는 다음과 같다.

$$G(s) = \frac{1}{s(s+1)} \quad (16)$$

또한 매 순간 e 와 \dot{e} 를 측정하여 제어입력 u 를 계산하는 규칙 테이블이 표 1에 나타나있다. 또한 대집합을 $(-\infty, +\infty)$ 로 갖는 e, \dot{e} 의 초기 퍼지 집합은 그림 4-(a), (b)에 보였다.

표 1. 퍼지제어 규칙

Table 1. Rule table.

$\dot{e} \ e$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NB	NB	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

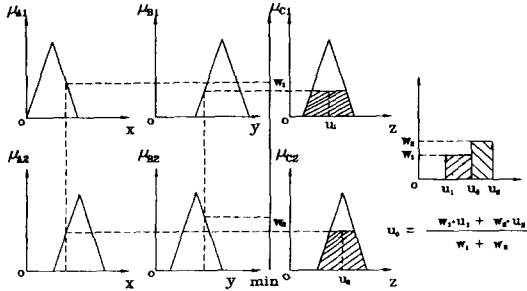
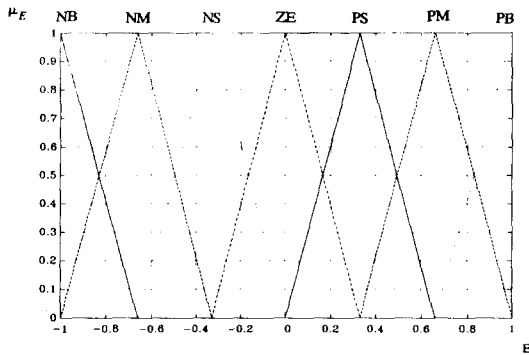


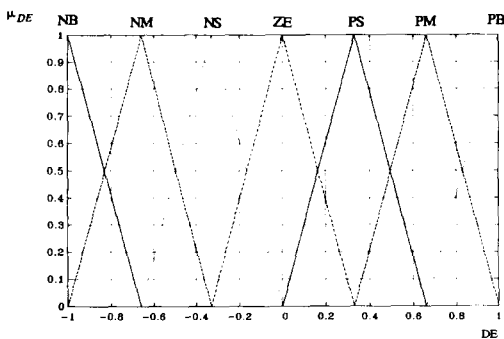
그림 3. 퍼지제어기에 의한 제어입력 계산의 도식적 해석

Fig. 3. Graphical interpretation of data processing of fuzzy controller.

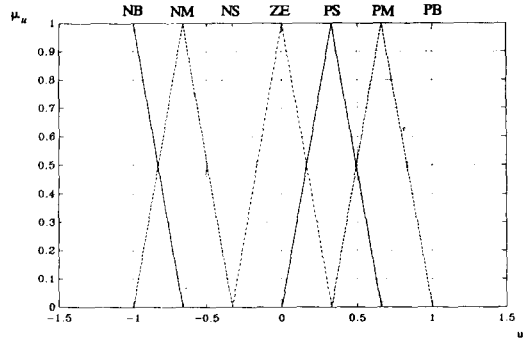
(-1.5, +1.5)를 갖는 u 의 초기 퍼지집합을 그림 4-(c)에 보인다. 또한 몇 번의 시운전을 통해 얻은 각 변수의 scaling factor는 S_E 는 1, S_D 는 0.5, S_U 는 10으로 이는 예를 들어 실제로 측정된 \dot{e} 의 값이 1.4이면 그림 4의 퍼지집합으로 환산될 때는 1.4×0.5 해서 0.7로 계산된다는 뜻이다.



(a)



(b)



(c)

그림 4. 학습전의 멤버십함수 (a)e, (b) \dot{e} , (c)u

Fig. 4. Membership function before learning for (a)e, (b) \dot{e} , (c)u.

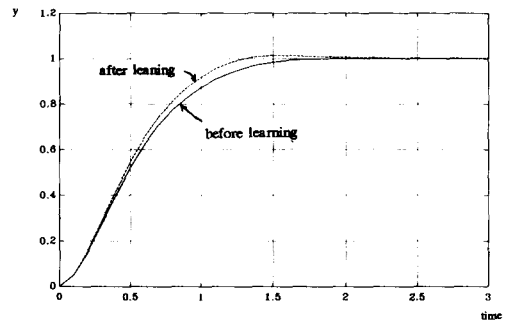


그림 5. 학습전후의 계단 입력에 대한 응답

Fig. 5. Response of (13) with step input before and after learning.

이렇게하면 제어 대상이 바뀌어도 정규화된 퍼지집합을 사용할 수 있어 편리하다. 그림 5의 초기모양으로 부터 2초 간격으로 기준입력을 -1 과 +1 사이의 값으로 랜덤하게 변화시키면서 학습을 시킨 결과 초기 퍼지 집합에 의한 step 입력의 응답과 비교하여 그림 5의 응답과 그림 6의 제어입력등을 얻었다. 학습과정의 상태는 그림 7에 나타나 있으며 결과로 구해진 퍼지 집합은 그림 8에 보였다. 사용한 학습율은 $(\eta_E, \eta_D, \eta_U) = (5.5, 0.1)$ 이었다.

B. type 0 시스템에의 적용

type 0 시스템은 step 입력에 대해 비레이득에 의해 정해지는 정상상태오차를 갖는다고 알려져있으며 여기서 적용된 시스템의 전달함수는 다음과 같다.

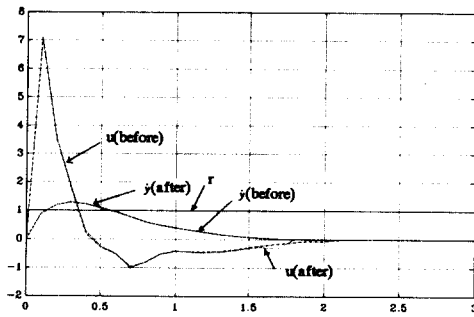


그림 6. 학습전후의 계단 입력에 대한 응답의 y, u
 Fig. 6. y and u of (13) for step input before and after learning.

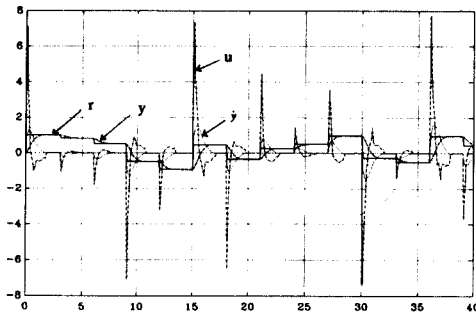
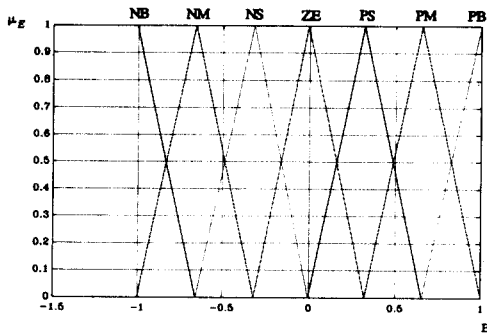
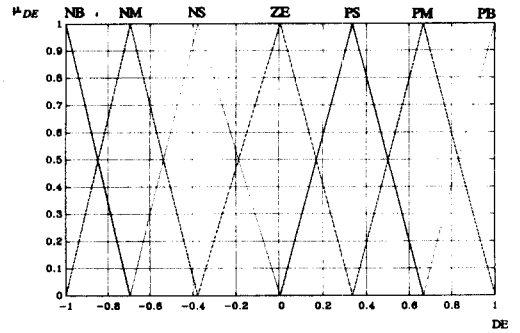


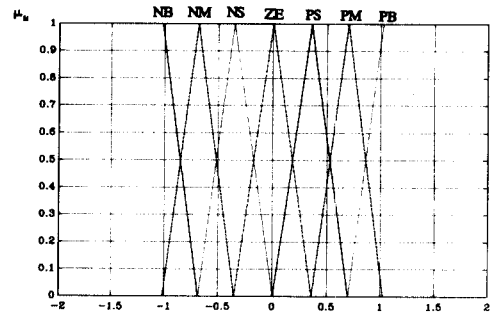
그림 7. 학습 과정
 Fig. 7. Learning process of (13).



(a)



(b)



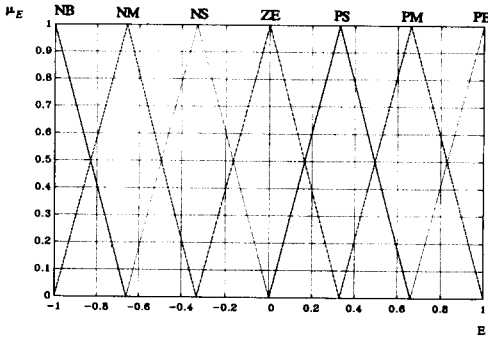
(c)

그림 8. 학습후의 멤버쉽함수 (a) e , (b) \dot{e} , (c) u
 Fig. 8. Membership function after learning for (a) e , (b) \dot{e} , and (c) u .

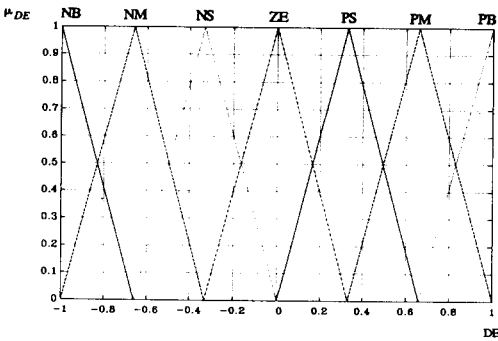
$$\Delta n_B(x') = \frac{\Delta n_B(x'_{PC}) \cdot \sinh\left(\frac{W' - x'_{PC}}{L_B}\right)}{\sinh\left(\frac{W' - x'_{PC}}{L_B}\right)} \quad (17)$$

또한 매 순간 e 와 \dot{e} 를 측정하여 제어입력 u 를 계산하는 규칙 테이블이 표 1과 같다. 또한 대집합을 $(-\infty, +\infty)$ 로 갖는 e, \dot{e} 의 초기 퍼지 집합은 그림 9(a, b)에 보였다. 그리고 대집합 $(-1.5, +1.5)$ 를 갖는 u 의 초기 퍼지 집합을 그림 9(c)에 보인다. 또한 몇 번의 시운전을 통해 얻은 각 변수의 scaling factor는 SE 는 1, SD 는 0.1, SU 는 2으로 이는 예를 들어 실제 측정된 \dot{e} 의 값이 1.4이면 그림 4의 퍼지 집합으로 환산될 때는 1.4×0.1 해서 0.14로 계산된다는 뜻이다. 그림 9의 초기모양으로 부터 2초 간격으로 기준입력을 -1 과 +1 사이의 값으로 랜덤하게 변화시키

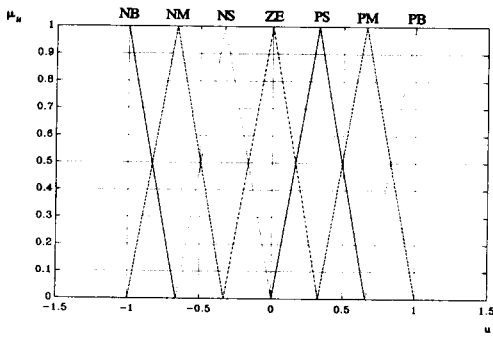
면서 학습을 시킨 결과 초기 퍼지 집합에 의한 step 입력의 응답과 비교하여 그림 10의 응답과 그림 11의 제어입력 등을 얻었다.



(a)



(b)



(c)

그림 9. 학습전의 멤버쉽함수 (a)e (b) \dot{e} , (c) u
Fig. 9. Membership function before learning for (a)e, (b) \dot{e} , (c) u.

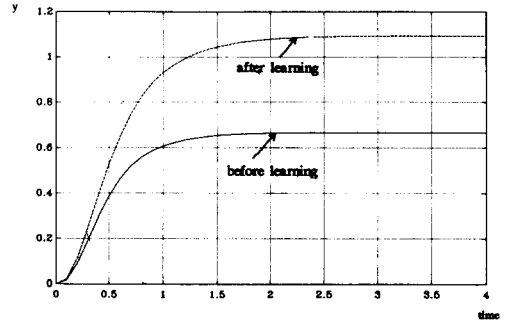


그림 10. 학습전후의 계단 입력에 대한 응답
Fig. 10. Response of (14) with step input before and after learning.

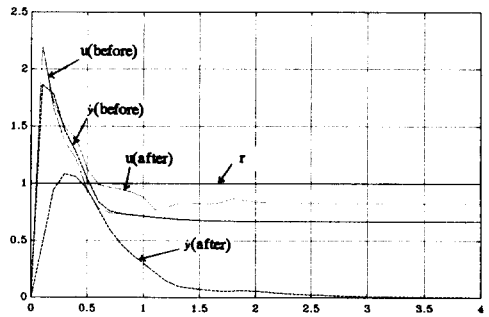


그림 11. 학습전후의 계단 입력에 대한 응답의 \dot{y} , u
Fig. 11. \dot{y} , u of (14) for step input before and after learning.

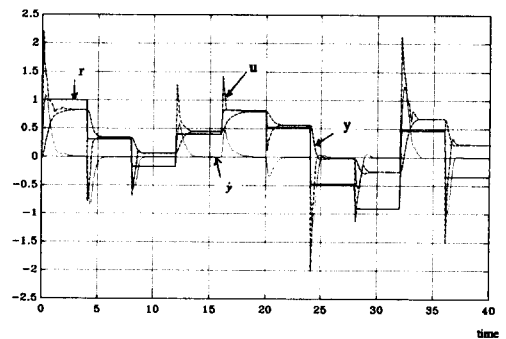
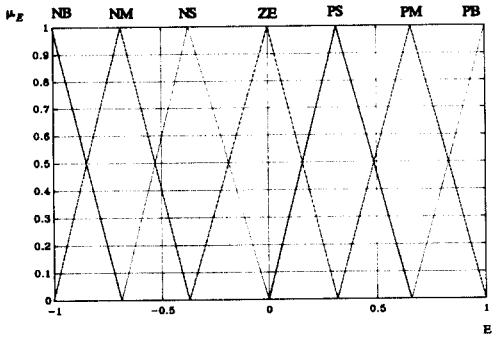
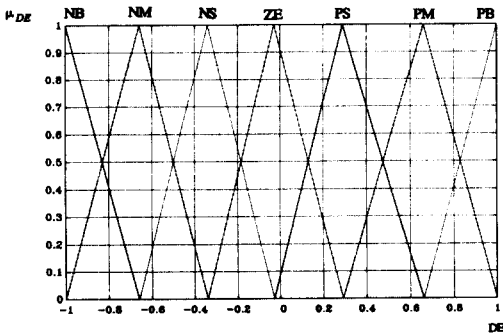


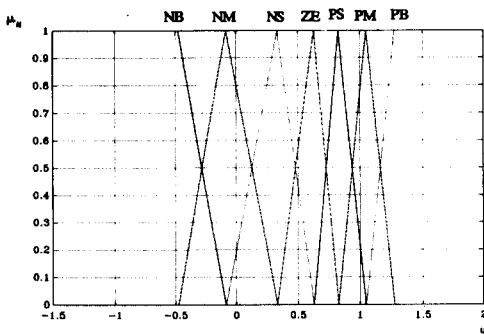
그림 12. 학습 과정
Fig. 12. Learning process of (14).



(a)



(b)



(c)

그림 13. 학습후의 멤버쉽함수 (a)e (b) \dot{e} , (c)u
 Fig. 13. Membership function after learning for (a)e (b) \dot{e} , (c)u.

학습과정의 상태는 그림 12에 나타나 있으며 결과로 구해진 퍼지 집합은 그림 13에 보였다. 이상의 결과로 다른 초기상태 $(y, \dot{y})=(2,0)$ 로 운전한결과 그림

14를 얻어 정상상태 오차가 개선되었음을 알 수 있는데 그 이유는 결과로 구해진 퍼지 집합의 비 대칭성으로부터 발생함을 알 수있다. 여기서 $\Delta y/\Delta u_0$ 로 이용되는 테이블을 구할때 사용되는 Δu_0 의 역할에 대해 알아볼 필요가있다. (17)의 시스템은 정지점 (equilibrium point)가 존재하는데 Δu_0 의 값을 너무 작게하면 그 정지점 근처에서 기울기가 $\Delta y/\Delta u_0$ 의 값이 아주 작은 값이 되어 학습이 정지하는 수가 있다. 물론 기준입력을 불규칙하게 변화시킴으로써 정지상태를 벗어날 수도있지만 Δu_0 의 크기를 적당히 정함으로써 학습의 효율도 증가시킬 수있다.

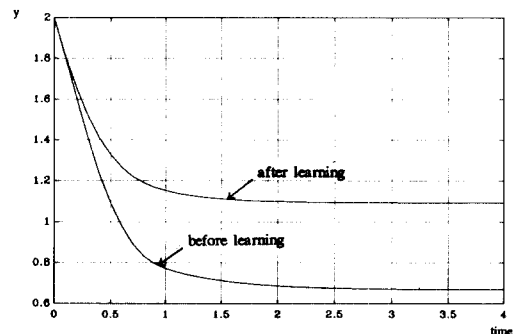


그림 14. (2,0)으로 부터의 응답

Fig. 14. Response of (14) with the initial condition $(y, \dot{y})=(2,0)$ and reference input $r=1$ before and after learning.

IV. 결 론

퍼지제어기에서 중요한 역할을 하는 언어변수와 제어규칙에 대해대상 시스템에 대해서 잘 알고있는 전문가에 의해서 제어기 설계자에게 전달된 초기상태로부터 학습을 통해 전달과정에서 발생할 수있는 불확실성을 극복하는 방법을 제안하였다.

본 논문에서는 중심값과 support만 고려한 삼각형 폴의 멤버쉽함수를 가정하고 제어대상 시스템의 운전을 통해 점차적으로 튜닝해가는 방법을 제안하였다. 이를 위해 일차적으로 퍼지제어기내의 모든 언어정보를 간단한 삼각형폴로 가정하고 이 각 삼각형의 세 꼭지점과 제어시스템의 성능과 연관시켜 제어 대상 시스템을 운전해가면서 제어기의 성능이 개선되는 방향으로 각 삼각형의 꼭지점의 위치를 변경시켜나가는 일종의 학습방법을 제시하였다.

제안된 알고리즘은 type 1 시스템에 적용되어 rise

time을 증가시키는 결과를 얻었으며 type 0 시스템에 적용되어 steady state error를 감소시키는 결과를 얻었다.

이상의 결과를 일반적으로 튜닝이 어렵다고 알려져 있는 PID type의 퍼지제어기로 확장적용하는 방법과, 성능지수를 어느 한순간에 국한하지않고 어떤 시간 구간으로 확장하여 학습의 효율을 구하는 방법과 일반적인 모양의 퍼지집합에 대해서도 가능한 학습방법등에 관한 연구를 지속적으로 수행해나갈 필요가 있다고하겠다. 그리고 이러한 학습에 의한 조정에 관한 여러방법을 목적에 따라 적절히 사용하기위하여는 이러한 종류의 각 방법의 특성을 비교하고 우수한 성능을 낼 수있는대상 시스템을 선정하는데 이용한다면 더욱 효과적인 응용이 가능하리라고 본다.

參考文獻

- [1] L. A. Zadeh, "Fuzzy sets", *Imformt. Control*, vol. 8, pp. 338-353, 1965.
- [2] S. S. L. Chang and L. A. Zadeh, "On Fuzzy Mapping and Control," *IEEE Trans. Syst. Man Cybern.*, vol. 15, no. 1, pp. 175-189, 1985.
- [3] C. C. Lee, "Fuzzy Logic in Control Systems : Fuzzy Logic Controller- Part I, II," *IEEE Trans. Syst. Man Cybern.*, vol. 20, no. 2, pp. 404-435, 1990.
- [4] R. M. Tong, "A Control Engineering Review of Fuzzy Systems," *Automatica*, vol. 13, pp. 559-569, 1977.
- [5] R. M. Tong, "Some Properties of Fuzzy Feedback Systems," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-10, no. 6, pp. 327-330, 1980.
- [6] E. H. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis witha Fuzzy Logic Controller," *Int. J. Man-Machine Studies*, vol. 7, no. 12, pp. 1-13, 1975.
- [7] M. Sugeno and K. Murakami, "Fuzzy Parking Control of Model Car," *Proc. 23rd IEEE Conf. on Decision and Control*, Las Vegas, NV, 1984.
- [8] M. Maeda, T. Sato, and S. Murakami, "Design of the Self-Tuning Fuzzy Controller," *Proc. Int. Conf. on Fuzzy Logic and neural Networks*, Iizuka, Japan, pp. 393-396.

著 者 紹 介

李 志 弘 (正會員)

1983年 서울대학교 전자공학과 졸업(학사). 1985年 한국과학기술원 전기 및 전자공학과 졸업(석사). 1985年 ~ 1988年 현대 중공업 로봇 사업부 근무. 1991年 한국과학기술원 전기 및 전자 공학과 졸업(박사). 1991年 3月 ~ 1991年 6月 한국과학기술원 전기 및 전자공학과 연수 연구원. 1991年 7月 ~ 현재 금오공과대학 전자제어공학과 조교수. 주관심분야는 지능제어, 로봇틱스, 뉴로 퍼지 시스템 등임.

蔡 錫 (正會員)

1978年 서울대학교 전기공학과 졸업(학사). 1980年 한국과학기술원 전기 및 전자공학과 졸업(석사). 1980年 3月 ~ 1983年 2月 현대 자동차 근무. 1989年 한국과학기술원 전기 및 전자 공학과 졸업(박사). 1983年 3月 ~ 현재 금오공과대학 전자제어공학과 부교수. 주관심분야는 퍼지제어, 공장 자동화, 대규모 시스템 제어 등임.

吳 英 碩 (正會員)

1982年 서울대학교 전자공학과 졸업(학사). 1984年 한국과학기술원 전기 및 전자공학과 졸업(석사). 1991年 한국과학기술원 전기 및 전자공학과 졸업(박사). 1989年 3月 ~ 현재 금오공과대학 전자제어공학과 조교수. 주관심분야는 로봇틱스, 컴퓨터 비전, 마이크로 프로세서 응용 등임.