

퍼 지 제 어

강 근 택

부산수산대학교 전자공학과

1. 머리말

Zadeh교수가 제안한 퍼지알고리즘[1]과 복잡한시스템의 언어모델링[2]을 이용하여 1974년 Mamdani[3]가 제어에 응용한 것이 퍼지제어의 시초이며, 1980년에 덴마크의 F. L.Smith 사에서 시멘트 킬른 제어에 응용한 것이 퍼지제어 실용화의 최초이다. 그후 1980년경 부터 일본을 중심으로 퍼지제어에 관한 연구와 실용화가 활발하기 시작하였다. 그러나, 진작 퍼지알고리즘을 제안한 Zadeh 교수는, 퍼지알고리즘이 제어에도 유효하리라고 논문에서 기술은 하였지만, 이렇게 제어공학에서 활발히 연구되리라고는 생각하지 못했다고 한다. 그것은 아마 Zadeh가 제어공학의 전문가이기 때문이 아닐까라고 한다. 사실 퍼지제어의 창시자 Mamdani는 제어공학자가 아닌 전자공학자이며, 그 후 퍼지제어를 크게 발달 시킨 일본의 Sugeno[4]도 원래 제어공학자는 아니었다.

퍼지제어를 발달시킨 것은 퍼지알고리즘과 언어모델링이다. 퍼지알고리즘이란 crisp 알고리즘을 퍼지화한 것으로 “x가 작으면 y를 증가 시켜라”와 같은 모호한 지시를 포함한다. 또한 언어모델링이란, humanistic system 또는 그 만큼 복잡한 시스템의 모델링을 위해 제안된 것으로, Zadeh의 부적합성원리에 근거를 두고 있다. 부적합성원리란 「시스템의 복잡함의 정도가 어느 한도를 넘으면 그 시스템의 거동을 정확하고 의미가 있게 수리적으로 기술하는 것은 불가능하다」는 것이다.

기존의 제어공학자들 중에는 퍼지제어가 비과학적이라고 비판하는 사람들이 있다. 많은 제어공학자들은, 세상의 모든 동역학은 미분방정식(차분방정식)이라는 명확한 기술언어로 표현 가능하다고 생각한다. 그리고 그러한 기술언어로

표현된 것을 분석하고, 다시 재구성하고, 검증하고, 보편성을 확인하는 작업을 함으로써 합리적이다 라고 말한다. 그러한 관점에서 보면, 퍼지제어는 「해보니까 되더라」는 식으로 설득력이 없고, 컴퓨터의 힘에 의한 억지 힘(brute force)으로 문제를 해결하는 것으로 보인다.

이와 같은 퍼지제어의 비판에 대해, Sugeno는 근대합리주의 자체에 비판적이며 다음과 같이 말한다[5].

“지금까지의 과학은 17세기 말 데카르트에 의해 시작된 근대합리주의가 지배적인 사상이다. 근대합리주의는 서양의 logos중심주의 즉 「모든 현상의 뒤에는 불변의 질서(logos)가 있다」에 근거를 두고 있다. 그러나 퍼지이론의 가치관은 데카르트의 가치관과 전혀 다른 것으로 생각한다. 데카르트가 제기한 근대과학의 방법론은 명확히 판명 가능한 데이터 또는 사실만을 대상으로 한다. 그러나 퍼지이론은 명확한 것을 대상으로 하는 것이 아니다. 따라서 퍼지이론은 데카르트의 합리주의에 처음부터 위반된다. 즉, 퍼지이론의 중심 세계는 수리과학의 세계가 아니다. Zadeh의 부적합성 원리도, 명확하지 않는 대상을 기술하기 위해서는 수리과학의 언어보다 모호한 자연언어가 더욱 적절하며 그렇게 기술한 것이 오히려 더 정확한 경우가 있다는 것을 말한다. 가령 정상 체온을 말할 경우 「350C 이상에서 370C 이하까지」보다는 「약 36.50C 정도」가 더 정확하다.(정밀함과 정확함의 의미는 서로 틀린다) 시스템 이론이 미분방정식이라는 기술언어 지상주의인데 반해, 퍼지이론은 언어의 모호함이라는 대상 지상주의라고 말할 수 있다. 결론적으로 퍼지제어에 의해 제어시스템론은 미분방정식이라는 기술언어 체계주의에서 다양한 가치관을 토대로한 더 넓은 이론으로 발전이 가능 하게 되었다고 생각한다.”

이 글에서는 퍼지제어기의 일반적인 형태 및 설계 방법과

최근 연구가 많이 되고 있는 TSK 퍼지시스템에 대해 살펴보고자 한다.

2. 퍼지제어기

2.1 퍼지제어기의 형태

일반적인 퍼지제어기(FLC : fuzzy logic controller)는 다음과 같은 퍼지제어규칙들로 구성된다.

$$R^i : \text{If } x_1 \text{ is } A_1^i, x_2 \text{ is } A_2^i, \dots, x_m \text{ is } A_m^i \\ \text{then } u \text{ is } B^i \quad (1)$$

여기서 i 는 i 번째 규칙임을 나타내고, x_i 는 전제부 변수로 제어에 필요한 변수이며, u 는 결론부 변수로 제어입력이다. A_i^i, B^i 는 각 변수에서 퍼지 집합이다.

퍼지제어규칙 R^i 는 다음과 같은 퍼지관계(fuzzy relation)로 쓸 수 있다.

$$R^i = A_1^i \times \dots \times A_m^i \times B^i \quad (2)$$

여기서, $A_1^i \times \dots \times A_m^i \times B^i$ 는 퍼지집합 A_1^i, \dots, A_m^i, B^i 들의 Cartesian product이며, 멤버십 값이 다음과 같은 퍼지 집합이 된다.

$$\mu_{R^i} \times \dots \times \mu_{A_m^i} \times \mu_{B^i}(x_1^0, \dots, x_m^0, u^0) = \\ \mu_{A_1^i}(x_1^0) \otimes \dots \otimes \mu_{A_m^i}(x_m^0) \otimes \mu_{B^i}(u^0) \quad (3)$$

여기서, \otimes 은 min, 대수곱셈(algebraic product)과 같은 t-norm 연산자이다. 이와 같은 퍼지규칙이 n 개 있을 경우, 퍼지제어기 R 은 다음과 같이 쓸 수 있다.

$$R = R^1 \oplus \dots \oplus R^n \quad (4)$$

여기서 \oplus 는 max, 대수덧셈(algebraic sum)과 같은 t-conorm 연산자이다. 그 밖의 퍼지제어기의 형태로는 다음과 같은 것들이 있다

- (1) 간략화된 퍼지추론(simplified fuzzy reasoning) : 퍼지제어규칙의 결론부가 퍼지집합이 아니고 다음과 같이 실수로 된 형태이다.

$$R^i : \text{If } x_1 \text{ is } A_1^i, x_2 \text{ is } A_2^i, \dots, x_m \text{ is } A_m^i \\ \text{then } u \text{ is } u^i \quad (5)$$

(1)의 퍼지규칙에서, 결론부 퍼지집합이 모든 규칙에서 똑같은 형태의 퍼지수로 구성될 경우, 결론부에 퍼지집합 대

신에 실수를 사용하면 연산이 훨씬 간편 해지면서 제어 효과는 거의 변화가 없다.

- (2) TSK 퍼지 시스템[6] : 퍼지규칙의 결론부가 입력들의 선형식으로 된 형태이다. 이것에 대해서는 제3장에서 상세히 설명한다.

- (3) 예측퍼지제어기(Predictive Fuzzy Control Rules) [7] : 현재 및 미래의 제어 결과를 예측하고 제어입력을 정하는 것으로 퍼지규칙이 다음과 같다.

$$\text{If } (u \text{ is } B^i \rightarrow (x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i)) \\ \text{then } u \text{ is } B^i \quad (6)$$

이러한 형태의 퍼지제어기는 지하철 자동운전에 응용되었으며, 그 때의 규칙의 하나를 언어로 표현하면 “만약 제어입력이 변하지 않고 열차가 정해진 위치 안에 정지 하면, 제어입력은 변화 시키지 않는다”와 같다.

2.2 퍼지제어기의 연산

(1) 형태의 규칙들로 구성되는 퍼지제어기의 연산은 퍼지화(fuzzification), 추론(inference), 비퍼지화(defuzzification)로 나눌 수 있다.

- (1) 퍼지화(fuzzification)

확정적인 입력값($x_1^0, x_2^0, \dots, x_m^0$)을 퍼지집합으로 바꾸는 작업이다. 관측된 입력 데이터에 random noise가 포함될 경우 관측된 값을 퍼지수(fuzzy number)로 바꾸어 퍼지화 한다. 그러나 통상 퍼지제어에서는 입력값으로 퍼지수를 사용할 경우 offset 등 좋지 않는 제어 결과가 나오므로, 퍼지수 대신에 다음과 같은 singleton 퍼지집합 A_i^0 를 사용한다.

$$\mu_{A_i^0}(x_k) = \begin{cases} 1 & x_k = x_k^0 \\ 0 & x_k \neq x_k^0 \end{cases} \quad (7)$$

- (2) 추론(inference)

입력값($x_1^0, x_2^0, \dots, x_m^0$)을 퍼지화시켜 얻은 입력퍼지 집합 $A_1^0, A_2^0, \dots, A_m^0$ 에 대한 퍼지제어입력 B^0 를 다음과 같이 구한다.

$$B^0 = R \circ (A_1^0 \times \dots \times A_m^0) \\ = \{R^1 \circ (A_1^0 \times \dots \times A_m^0)\} \oplus \dots \oplus \{R^n \circ (A_1^0 \times \dots \times A_m^0)\} \quad (8)$$

여기서 \circ 는 compositional operator이며, $B_i^0 = R^i \circ (A_1^0 \times \dots \times A_m^0)$ 라면 퍼지집합 B_i^0 는 다음과 같이 구한다.

$$\mu_{B_i^0}(u) = \max_{x_1, \dots, x_m} [\mu_{R^i}(x_1, \dots, x_m, u) \otimes \mu_{A_1^0} \times \dots \times \mu_{A_m^0}]$$

$$\begin{aligned} & (x_1, \dots, x_m) \\ &= \max_{x_1, \dots, x_m} [\mu_{A_1^i}(x_1) \otimes \dots \otimes \mu_{A_m^i}(x_m) \otimes \mu_{B^i}(u) \otimes \mu_{A_1^o} \\ & (x_1) \otimes \dots \otimes \mu_{A_m^o}(x_m)] \end{aligned} \quad (9)$$

입력퍼지집합 A_k^o 가 (7)과 같은 singleton인 경우

$$\begin{aligned} \mu_{B^i}(u) &= \mu_{A_1^i}(x_1^0) \otimes \dots \otimes \mu_{A_m^i}(x_m^0) \\ \mu_{B^i}(u) \end{aligned} \quad (10)$$

가 되어, 추론된 퍼지제어입력 B^o 은 (8)에 의해 다음과 같이 된다.

$$\begin{aligned} \mu_{B^o}(u) &= \{\mu_{A_1^i}(x_1^0) \otimes \dots \otimes \mu_{A_m^i}(x_m^0) \otimes \mu_{B^i}(u)\} \\ & \oplus \dots \oplus \{\mu_{A_1^n}(x_1^0) \otimes \dots \otimes \mu_{A_m^n}(x_m^0) \otimes \\ & \mu_{B^n}(u)\} \end{aligned} \quad (11)$$

연산자 \otimes 과 \oplus 의 대표적인 것들을 표 1에 보인다.

표 1. t-norm과 t-conorm 연산자들.

t-norm	t-conorm
min : $a \wedge b$	max : $a \vee b$
Hamacher product : $ab/(a+b-ab)$	Hamacher sum : $(a+b-2ab)/(1-ab)$
algebraic product : ab	algebraic sum : $a+b-ab$
Einstein product : $ad/(1+(1-a)(1-b))$	Einstein sum : $(a+b)/(1-ab)$
bounded product : $0 \vee (a+b-1)$	bounded sum : $1 \wedge (a+b)$
drastic product : $\begin{cases} a & b=1 \\ b & a=1 \\ 0 & a,b < 1 \end{cases}$	drastic sum : $\begin{cases} a & b=0 \\ b & a=0 \\ 1 & a,b > 1 \end{cases}$

(9)의 \otimes 과 \oplus 연산에 Mamdani는 min과 max를 사용하였으며, 그 후 사용하기가 간편하여 min, max가 가장 많이 사용되고 있다. 그러나, Mizumoto[8]는 \otimes 과 \oplus 연산자로 min, max 보다 대수곱셈, 덧셈을 사용하는 것이 더욱 유효함을 밝혔다. \otimes 연산에 min을 사용할 경우 (10)에서 최소치 보다 높은 항의 값의 변화는 무시되고 만다. 또한 연산에 max를 사용할 경우에도, 멤버십 값이 적은 퍼지규칙의 결과는 무시되어 직관에 맞지 않으며 퍼지규칙의 tuning도 곤란 해진다. 따라서 \oplus 연산에는 덧셈과 같은 가법성을 갖고 있는 연산을 사용하는 것이 유효하다.

(3) 비퍼지화(defuzzification)

퍼지제어기에서 추론하여 얻은 퍼지제어입력 B^o 을 비퍼지 제어입력 y^o 로 바꾸는 작업이다. 비퍼지화의 전략은, 추론된 퍼지제어입력의 가능성 분포를 가장 잘 표현하는 비퍼지 제어입력으로 바꾸는 것이다. 그 방법으로 가장 많이 사용되는 것은 다음과 같은 중심법(Center of Area Method)이다.

$$y^o = \frac{\int B^o(y) y dy}{\int B^o(y) dy} \quad (12)$$

이외에, 멤버십 값이 가장 높은 점들의 중간값을 취하는 방법(mean of maximum method), 가능성 분포가 가장 높은 점을 찾는 방법(max criterion method) 등이 있다.

2.3 퍼지제어기의 설계

퍼지제어기의 일반적인 설계 순서는 다음과 같다.

(1) 제어입력과 그에 관계되는 변수들을 가능한 모두 찾아내고, 숙련가(expert)의 의견 또는 입출력 데이터의 해석 등으로 제어대상의 입출력 변수를 결정한다.

(2) 퍼지제어기의 형태, 퍼지집합의 멤버쉽함수의 형태를 결정한다.

(3) 퍼지제어규칙을 만든다. 퍼지제어 규칙을 찾아내는 방법은 다음과 같이 크게 4가지가 있다. 퍼지제어규칙을 만들 때는 이 4가지 방법들을 병행하여 사용할 수도 있다.

① 제어대상의 숙련가의 경험 및 지식과 제어공학자의 지식을 이용한다.

이 방법은 숙련된 운전자(operator)의 경험 또는 제어전문가의 지식 등을 정성적인 언어로 표현하고, 퍼지제어규칙 형태로 논리화 하는 것이다.

② 숙련된 운전자의 제어조작을 모델화하여 퍼지제어기로 사용한다.

제어 대상이 복잡한 경우, 숙련된 운전자는 운전은 매우 잘 하지만 그 조작을 논리화 하는 것은 쉽지 않다. 왜냐하면 운전자가 경험을 항상 언어로 표현하여 기억하고 조작하는 것은 아니다. 예로써 자동차 운전자는 몸에 벤 감각으로 운전한다. 이러한 경우, 숙련운전자의 운전을 관측하여 입출력 데이터를 얻고, 그 데이터를 이용하여 숙련운전자의 운전을 퍼지모델링하고, 그 퍼지모델을 퍼지제어기로 사용한다. 이

때 퍼지모델의 형태로는 TSK퍼지모델이 많이 사용된다[9].

③ 제어대상 플랜트(plant)의 퍼지모델을 작성하고 그 퍼지모델로부터 퍼지제어기를 설계한다.

앞의 2방법은 숙련운전자의 모델을 만들어 퍼지제어기로 사용하는 것이므로 숙련가의 능력 이상의 결과를 기대하기는 어렵다. 숙련가가 없는 경우 또는 숙련가보다 더 나은 제어를 원하는 경우, 플랜트의 퍼지모델을 작성하고 그 퍼지모델을 토대로 퍼지제어기를 설계 한다. 플랜트의 퍼지모델의 형태로 TSK퍼지모델을 사용하는 경우 퍼지제어기의 설계법은 제3장에 설명한다. 플랜트의 퍼지모델로 (1)과 같은 Mamdani형을 사용할 경우 퍼지제어기의 설계법은 아직 완전히 개발되지 않고 있으며, 현재 크게 연구 관심의 대상이 되고 있다.

2.4 퍼지적응제어

퍼지 적응제어의 형태는 퍼지제어기가 퍼지모델에 토대로 설계되느냐에 따라 크게 나뉜다.

(1) 퍼지제어기가 퍼지모델을 토대로 설계 되는 경우 [10][11]

퍼지모델을 프로세스의 입출력값으로 축차조정하고 그에 따라 퍼지제어기의 파라미터를 조정하는 방법이다.

(2) 퍼지제어기의 자기조정(self organization)

제어결과와 성능 평가 기준을 정하고, 그 평가 기준을 토대로 퍼지제어기를 학습시키는 방법이다. 제어성능 평가 기준으로는 제어편차의 제곱, overshoot 퍼센트, settle time 등과 같은 양적인 기준과 위험성, 승차감, 불안감 등과 같은 질적인 기준을 생각 할 수 있다. 그 방법들로는

① 경사하강법(gradient descent)를 이용하여, 퍼지제어기 퍼지집합의 멤버십 함수의 파라미터를 자동 조정한다 [12]. 그러나 이 방법은 on-line이 아니고 off-line이며 계산량이 많은 단점이 있다. .

② 퍼지제어기의 조정부에 meta-rule base를 만들어, 제어성능 평가에 따라 퍼지집합의 멤버십 함수 또는 scaling factor를 조정하거나 제어규칙을 조정한다.[13]

③ Neural Network를 이용하여 퍼지제어기의 파라미터를 학습시킨다.

3. TSK 퍼지시스템

3.1 TSK 퍼지모델

TSK(Tanaka - Sugeno - Kang) 퍼지모델은 1983년 Takagi와 Sugeno[6]가 맨 처음 제안한 것으로, 다음과 같이 퍼지규칙의 결론부가 시스템 입출력의 선형관계식으로 되어 있다.

$$L: \text{If } z_1 \text{ is } A_1^i, z_2 \text{ is } A_2^j, \dots, z_m \text{ is } A_m^i \quad (13)$$

$$\text{then } y^i = c_0^i + c_1^i/x_1 + c_2^i/x_2 + \dots + c_n^i/x_n$$

여기서, z_i 는 전제부 변수, x_i 는 결론부 변수이며 A_i^j 는 퍼지집합이다. 전제부 변수와 결론부 변수는 같은 경우가 많다.

어느 한 입력($z_1^0, z_2^0, \dots, z_m^0$)에 대한, 추론 결과 y^0 는 다음과 같이 가중평균법(weighted average method)에 의해 구해진다.

$$y^0 = \left(\sum_{i=1}^n w^i y^i \right) / \sum_{i=1}^n w^i \quad (14)$$

$$w^i = \prod_{j=1}^m \mu_{A_j^i}(z_j^0) \quad (15)$$

+

TSK퍼지 모델은, 입력 공간을 퍼지 분할하여 전제부에서 정해진 퍼지 부분공간에서 입출력의 선형관계식을 결론부에서 표현한 형태이며, 복잡한 비선형 시스템의 표현에 매우 뛰어난 능력을 갖고 있다. TSK 퍼지모델을 퍼지제어기 설계에 사용하는 방법은 2가지가 있다. 하나는 숙련된 제어 운전자의 제어조작을 TSK모델로 인식(identification)하여 퍼지제어기로 사용하는 것이고, 또 하나는 제어대상 플랜트(plant)의 TSK퍼지모델을 작성하고 그 TSK퍼지모델로부터 TSK퍼지제어기를 설계하는 방법이다.

3.1.1 TSK 퍼지모델의 인식[14]

입출력 데이터 주어졌을 때, 그 데이터를 이용하여 TSK 퍼지모델의 인식하는 알고리즘은 그림 1과 같다.

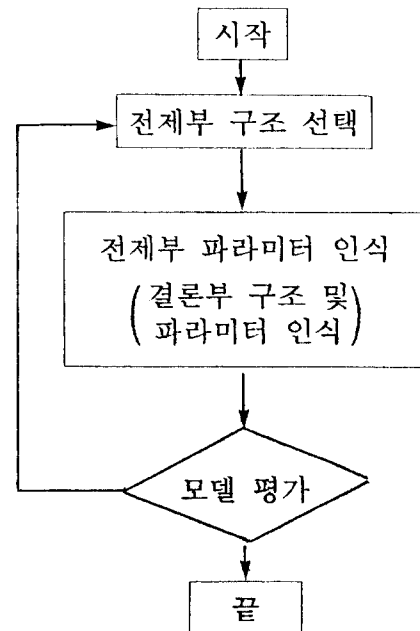


그림 1. TSK 퍼지모델의 인식.

(1) 전제부 구조 선택이란 전제부 변수공간의 퍼지분할을 찾는 것이며, 퍼지분할하여 만들어진 퍼지부분공간의 수 만큼 퍼지규칙이 만들어 진다. 전제부 구조 선택의 방법은 맨 처음 규칙이 한개인 모델 즉 선형모델로 부터 시작하여 모델평가 규범의 기준을 만족할 때까지 점차 퍼지부분공간의 수를 늘려간다.

(2) 전제부 파라미터 인식은 전제부 퍼지집합의 멤버쉽 함수를 결정하는 파라미터를 구하는 것이다. 전제부 퍼지집합으로 그림 2와 같은 형태를 사용할 경우 그림 2에서 p_1, p_2, p_3, p_4 가 전제부 파라미터이다. 인식 방법은 비선형계획법 또는 유전적 알고리즘(GA)을 이용하며, 이 때 사용하는 목적 함수는 퍼지모델 출력 오차의 평방합을 사용한다. 따라서 전제부 파라미터 인식은 결론부 인식도 하면서 진행하게 된다.

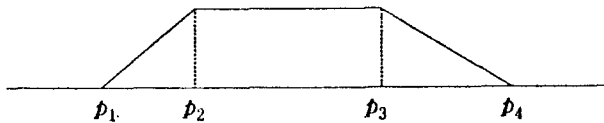


그림 2. 전제부 파라미터.

(3) 결론부 인식에서는, 출력 y^0 가 결론부 parameter들 ($c_0^i, c_1^i, \dots, c_m^i$)의 한개 선형식으로 표현 되어지므로, 선형 시스템에서 통상 하는 수법을 이용할 수 있다. 즉, 결론부 구조인식이란 어느 변수가 결론부에 관계하고 있는가를 찾는 것으로 변수감소법을 이용하여 구하며, 결론부 파라미터 인식이란 결론부 선형식의 계수를 구하는 것으로 최소자승법을 이용하며 그 방법은 다음과 같다.

퍼지모델로 부터 출력 y^0 는 (14)에서

$$\begin{aligned}
 y^0 &= \sum_{i=1}^n w^i (c_0^i + c_1^i x_1 + c_2^i x_2 + \dots + c_m^i x_m) / \sum_{i=1}^n w^i \\
 &= c_0^i g_0^i + c_1^i g_1^i + c_2^i g_2^i + \dots + c_m^i g_m^i + \\
 &\quad \vdots \\
 &= c_0^i g_0^i + c_1^i g_1^i + c_2^i g_2^i + \dots + c_m^i g_m^i \quad (16) \\
 g_i^i &= w^i x_i / \sum_{i=1}^n w^i \quad (\text{단, } x_0=1)
 \end{aligned}$$

와 같이 된다. 따라서 전제부가 정해지면 결론부 파라미터 ($c_0^i, c_1^i, \dots, c_m^i$)들은 통상의 최소자승법을 이용하여 구할 수 있으며, 축차인식(recursive identification)도 가능하다.

(4) 모델평가는, 인식용 데이터로 부터 인식된 퍼지모델이 인식에 사용된 데이터에만 적합한 것이 아니고 실제 시스템을 정확히 표현하는 보편성 있는 모델인가를 평가하는 것이다. 즉 인식용 데이터의 출력오차만을 생각하면, 규칙 수를 늘릴수록 출력오차는 줄어든다. 그러나 그러한 모델은

참 모델이 될 수 없다. 따라서 모델평가 검증의 규범을 만들어 규범값이 증가하면 전제부 구조선택에서 더 이상 퍼지부분공간의 수를 늘리지 않도록 한다. 평가 규범의 예로 GMDH에서 비선형모델의 구조검증에 쓰이는 UC(Unbiasedness Criterion)를 사용한다.

3.1.2 TSK 퍼지시스템의 안정성

동적 시스템을 TSK퍼지모델로 표현하였을 때, 퍼지규칙의 결론부를 상태표현형으로 하면 다음과 같이 쓸 수 있다.

$$\begin{aligned}
 L^i : \text{If } z_1 \text{ is } F_1^i, z_2 \text{ is } F_2^i, \dots, z_m \text{ is } F_m^i \text{ then } x^i \\
 (k+1) = A_i^i(k) \quad (17)
 \end{aligned}$$

여기서, z_i 는 전제부 변수, F_i^i 는 퍼지집합, $x(k)$ 는 상태변수 벡터이며, A_i^i 는 상태전이 행렬이다.

이러한 시스템의 안정성 판별법에 대해, Tanaka[15]는 Lyapunov 제2법을 사용하여 다음과 같은 정리를 만들었다.

정리 1. : (18)을 만족하는 positive definite matrix P 가 존재하면 (17)의 퍼지규칙들로 구성되는 TSK퍼지시스템은 globally 점근 안정하다.

$$A^i P A_i^i - P < 0 \quad \text{for } i=1,2, \dots, n \quad (18)$$

3.2 TSK 퍼지제어기

TSK퍼지모델로 부터, 전체시스템의 안정성을 고려하면서 TSK퍼지제어기를 설계하는 방법을 Kang[16]과 Tanaka[15]가 각각 제안하였다. Tanaka가 제안 방법은 Lyapunov 제2법을 사용하였으며, Lyapunov함수를 찾기가 어려운 단점이 있어 실제 사용하기는 어렵다. Kang이 제안한 방법은 극배치법(poleplacement method)을 이용하는 것으로 전체시스템의 pole이 원하는 값이 되도록 TSK 퍼지제어기를 간편하게 설계할 수 있다. 지금부터 그에 대해 설명한다.

동적시스템을 TSK퍼지모델로 표현하면 다음과 같다.

$$\begin{aligned}
 L^i : \text{If } z_1 \text{ is } F_1^i, z_2 \text{ is } F_2^i, \dots, z_m \text{ is } F_m^i \\
 \text{then } x(k+1) = A_i^i(k) + B^i u(k) \quad (19) \\
 y(k) = Cx(k)
 \end{aligned}$$

여기서, $x(k)$ 는 상태변수벡터, $u(k)$ 는 제어입력, $y(k)$ 는 출력변수이다. 지금부터, 전제부 변수 z_k 는 현재의 제어입력 값 $u(k)$ 에 관계 없다고 가정한다. (19)와 같은 퍼지규칙이 n 개 있을 때, 추론되는 상태변수값 $x(k+1)$ 은 (14)에 의해 다음과 같이 구한다.

$$x(k+1) = \left(\sum_{i=1}^n w^i x^i(k+1) \right) / \sum_{i=1}^n w^i \quad (20)$$

3.2.1 극 배치법에 의한 TSK퍼지제어기 설계

TSK 퍼지모델의 결론부가 선형식이므로 선형제어시스템에서 사용하는 극 배치법(pole assignment method)을 이용하여 퍼지제어기를 설계한다. 설계된 퍼지제어기는 전체 제어 시스템의 극(pole)이 원하는 값이 되도록 한다.

TSK퍼지제어기는 대상 시스템의 TSK퍼지모델과 같은 수의 TSK퍼지규칙들로 구성되며, 그 퍼지규칙의 전제부는 퍼지모델의 전제부와 같다. (19)의 퍼지규칙에 대한 퍼지제어기의 퍼지규칙은 다음과 같다.

$$\begin{aligned} C^i : & \text{If } z_1 \text{ is } F_1^i, z_2 \text{ is } F_2^i, \dots, z_m \text{ is } F_m^i \\ & \text{then } u(k) = -G^i x(k) \end{aligned} \quad (21)$$

여기서 G^i 는 상태피드백 이득(state feedback gain)이다.

전체 제어시스템의 원하는 상태천이행렬이 Φ 라면, 상태 피드백 이득 G^i 는

$$\Phi = A^i - BG^i \quad \text{for } i=1, 2, \dots, n \quad (22)$$

로 부터 구한다. TSK퍼지모델에서 출력을 추론 할때는 (14)와같이 가중평균법을 사용하였으나, (21)의 TSK 퍼지 제어규칙의 $u_i(k)$ 들로부터 제어입력 $u(k)$ 를 추론하는 방법은 다음과 같은 변형된 가중평균법을 이용한다.

$$\sum_{i=1}^n w^i(z) B^i u(k) = \sum_{i=1}^n w^i(z) B^i u^i(k) \quad (23)$$

위의 퍼지제어기를 사용할 경우 제어시스템의 상태천이행렬이 원하는 Φ 가 됨을 다음 정리에서 증명한다.

정리 2 : (22)에 의해 설계된 (21)의 TSK퍼지제어기에서 (23)의 추론법으로 제어입력 $u(k)$ 를 구하여 (19)의 TSK시스템을 제어하면 상태천이행렬이 원하는 값 Φ 와 일치한다.

증명 : (20)~(23)로 부터 상태벡터 $x(k+1)$ 는 다음과 같이 된다.

$$\begin{aligned} x(k+1) &= \sum_{i=1}^n w^i(z) (A^i x(k) + B^i u(k)) / \sum_{i=1}^n w^i(z) \\ &= \left(\sum_{i=1}^n w^i(z) A^i x(k) + \sum_{i=1}^n w^i(z) B^i u(k) \right) / \sum_{i=1}^n w^i(z) \\ &= \Phi x(k) \quad \blacksquare \quad (24) \end{aligned}$$

3.2.2 극 배치법에 의한 TSK퍼지관측기 설계

TSK퍼지관측기도 퍼지제어기와 마찬가지로 극 배치법에 의해 설계될 수 있다. (19)에서 구해지는 TSK퍼지관측기의 i 번째 규칙은 다음과 같다.

$$\begin{aligned} O^i : & \text{If } z_1 \text{ is } F_1^i, z_2 \text{ is } F_2^i, \dots, z_m \text{ is } F_m^i \\ & \text{then } \bar{x}(k+1) = A^i \bar{x}(k) + B^i u(k) + \\ & H^i (y(k) - C^i \bar{x}(k)) \end{aligned} \quad (25)$$

여기서 $\bar{x}(k)$ 는 예측 상태벡터이며, H^i 는 관측기 피이드백 행렬이다.

상태오차벡터 $\tilde{x}(k)$ 를 다음과 같이 정의하고

$$\tilde{x}(k) = x(k) - \bar{x}(k) \quad (26)$$

상태오차벡터 $\tilde{x}(k)$ 의 원하는 상태천이행렬을 Θ 라면, 관측기 피이드백 행렬 H^i 는

$$\Theta = A^i - HC^i \quad \text{for } i=1, 2, \dots, k \quad (27)$$

로 부터 구해진다. (25)의 TSK 퍼지관측기의 $\bar{x}(k+1)$ 들로부터 예측상태벡터 $\bar{x}(k+1)$ 를 추론하는 방법은 다음과 같이 가중평균법을 이용한다.

$$\bar{x}(k+1) = \sum_{i=1}^n w^i(z) \bar{x}^i(k+1) / \sum_{i=1}^n w^i(z) \quad (28)$$

위의 퍼지관측기를 사용할 경우 상태오차벡터 $\tilde{x}(k)$ 의 상태천이행렬이 원하는 Θ 가 됨을 다음 정리에서 증명한다.

정리 3 : (27)로 설계된 (25)의 TSK퍼지관측기에서 상태벡터를 예측하면 (26)의 상태오차벡터의 상태천이행렬이 원하는 Θ 와 일치한다.

증명 : (19), (25)~(28)로 부터 상태오차벡터 $\tilde{x}(k+1)$ 는 다음과 같이 된다.

$$\begin{aligned} \tilde{x}(k+1) &= \sum_{i=1}^n w^i(z) \{ A^i x(k) - A^i \bar{x}(k) - H^i (y(k) - \\ & C^i \bar{x}(k)) \} / \sum_{i=1}^n w^i(z) = \sum_{i=1}^n w^i(z) (A^i - HC^i) \\ & \bar{x}(k) / \sum_{i=1}^n w^i(z) = \Theta \bar{x}(k) \quad \blacksquare \quad (29) \end{aligned}$$

3.2.3 TSK 퍼지제어기와 관측기를 갖는 퍼지제어시스템

TSK 퍼지제어기와 관측기를 갖는 TSK퍼지제어시스템의 블록도를 그림 3에 보인다. 정리 4는 TSK퍼지제어시스템이 소위 separation property를 만족함을 보인다.

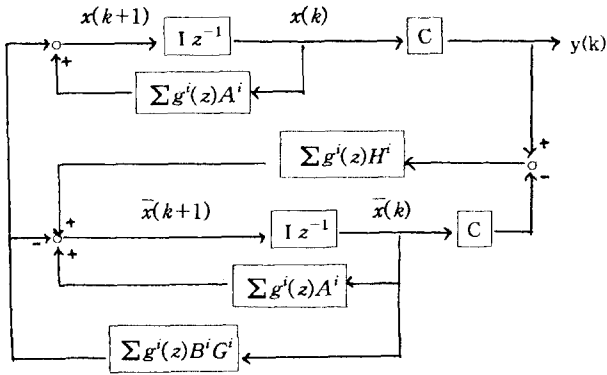


그림 3. TSK 퍼지제어시스템.

정리 4(Seperation Property) : TSK 퍼지제어기와 관측기를 갖는 TSK 퍼지제어시스템의 pole은, TSK 퍼지관측기가 없는 TSK퍼지제어기의 pole과 TSK퍼지관측기의 pole들로 구성된다.

증명

$$g^i(z) = w^i(z) / \sum_{j=1}^k w^j(z) \quad (30)$$

라면, (24), (29)와 $u^i(k) = -G^i \bar{x}(k)$ 으로 부터

$$\begin{pmatrix} x(k+1) \\ \bar{x}(k+1) \end{pmatrix} = \begin{pmatrix} \sum g^i(z)A^i - \sum g^i(z)B^iG^i & \sum g^i(z)B^iG^i \\ \sum g^i(z)B^iG^i & \sum g^i(z)A^i - \sum g^i(z)HC \end{pmatrix} \begin{pmatrix} x(k) \\ \bar{x}(k) \end{pmatrix} \quad (31)$$

가 되어 특성방정식은 (22), (27)과 $\sum g^i(z)=1$ 에 의해

$$|zI - \Phi| \times |zI - \Theta| = 0 \quad (32)$$

따라서 전체시스템의 pole은 Φ 과 Θ 의 고유치가 된다. ■

3.2.4 예제

프로세스의 TSK퍼지모델이 그림 4와 같을 때, TSK퍼지제어기와 관측기를 설계 한 것을 그림 5와 6에 보인다. 그림 5의 제어기 설계시 원하는 pole은 $0.4 \pm j0.4$ 이며, 그림 6의 관측기의 원하는 pole은 $0.4 \pm j0.2$ 이다. 그림 5의 TSK퍼지제어기로 부터 제어입력 $u(k)$ 는 (23)에 의해 다음과 같이 구해진다.

$$u(k) = (1.2w^1 u^1(k) + 1.4w^2 u^2(k)) / (1.2w^1 + 1.4w^2) \quad (33)$$

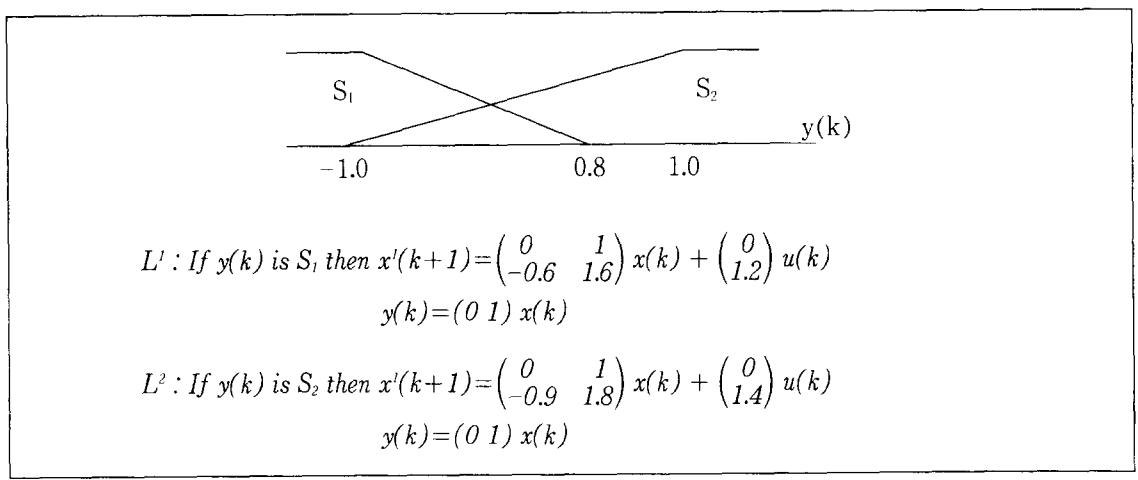


그림 4. 플랜트의 TSK 퍼지모델.

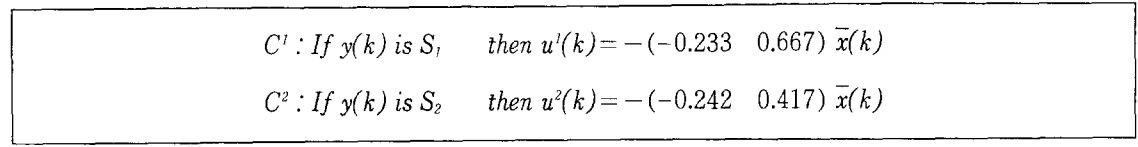
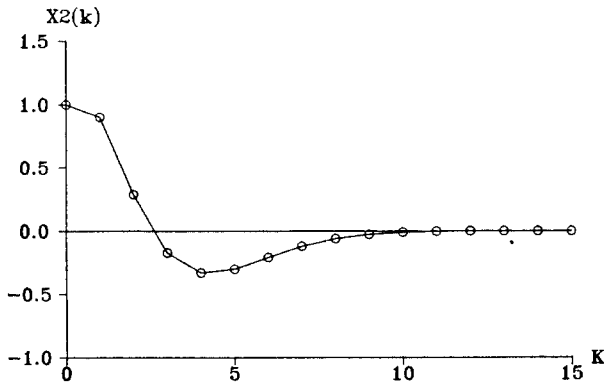


그림 5. TSK 퍼지제어기.

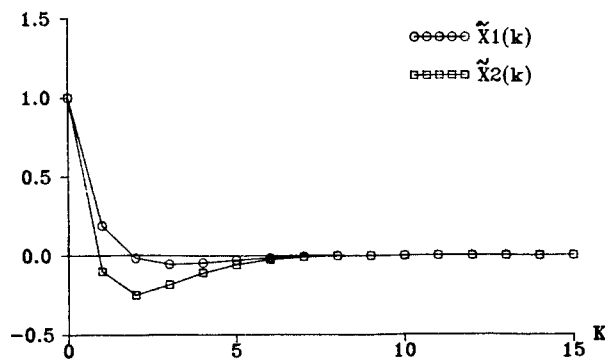
$$\begin{aligned}
 O^1 : & \text{ If } y(k) \text{ is } S_1 \\
 & \text{ then } \bar{x}^1(k+1) = \begin{pmatrix} 0 & 0 \\ -0.6 & 1.6 \end{pmatrix} \bar{x}(k) + \begin{pmatrix} 0 \\ 1.2 \end{pmatrix} u(k) + \begin{pmatrix} 0.667 \\ 0.8 \end{pmatrix} (y(k) - (0 \ 1) \bar{x}(k)) \\
 O^2 : & \text{ If } y(k) \text{ is } S_2 \\
 & \text{ then } \bar{x}^2(k+1) = \begin{pmatrix} 0 & 1 \\ -0.9 & 1.8 \end{pmatrix} \bar{x}(k) + \begin{pmatrix} 0 \\ 1.4 \end{pmatrix} u(k) + \begin{pmatrix} 0.778 \\ 1.0 \end{pmatrix} (y(k) - (0 \ 1) \bar{x}(k))
 \end{aligned}$$

그림 6. TSK퍼지관측기.

그림 7은 $x(0) = (1, 1)$, $\bar{x}(0) = (0, 0)$ 일 때, 제어 결과와 상태오차벡터의 거동을 보인다.



(a) 제어결과



(b) 상태오차벡터의 거동

그림 7. 제어 결과와 상태오차벡터의 거동.

4. 맺음말

퍼지제어기의 일반적인 형태 및 설계 방법과, TSK퍼지시스템에 대해 살펴 보았다. 동양사상이 서양의 합리주의와 거리가 먼 까닭인지 퍼지제어의 이론 및 응용이 일본을 중심으로 동양에서 많이 발전 되었으며, 한국에서의 퍼지학문도 출발은 늦었지만 급속히 발전하고 있다. 최근에는 미국

에서도 관심이 높아져 1993년부터 IEEE에서 Fuzzy systems의 논문지가 나오고 있으며, NASA 등에서도 응용에 많은 관심을 갖게 되었다.

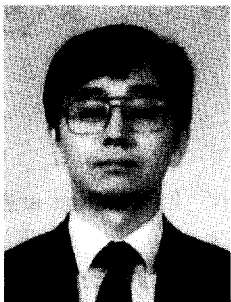
퍼지이론은 언어의 모호함을 대상으로 하는 것으로 부터 출발 하였다. 언어시스템은 우주에서 가장 복잡하고 가장 높은 수준의 시스템으로 알려져 있다. 이러한 언어시스템을 가장 단순한 시스템인 물리시스템을 대상으로 출발한 수리과학으로 접근하는 것은 무리일 것이다. 퍼지이론으로 언어시스템을 어떻게 잘 이용할 수 있는 가를 연구하는 것이 퍼지제어의 근본 과제라 할 수 있다.

참고 문헌

- [1] L.A. Zadeh, "Fuzzy Algorithms", Information and Control, 12-2, pp. 94-102, 1968.
- [2] L.A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", IEEE Trans. Systems, Man and Cybernetics, SMC-3-1, pp. 28-44, 1973
- [3] E.H. Mamdani, "Applications of Fuzzy Algorithms for Control of Simple Dynamic Plant", Proc. IEEE, 121-12, pp. 1585-1588, 1974.
- [4] 菅野道夫, ファジィ制御, 日刊工業新聞社, 1988
- [5] 菅野道夫 外3人 "ファジィ·AI·ニューロとシステム理論", システム/制御/情報 39-1, pp. 2-13, 1995
- [6] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modelling and Control" IEEE Trans. on SMC, 15-1, pp. 116-132, 1985
- [7] S. Yasunobu, S. Sekino, and T. Hasegawa, "Automatic Train Operation and Automatic Crane Operation Systems based on predictive Fuzzy Control", in Proc. 2nd IFSA Congress, pp. 835-838, 1987
- [8] M. Mizumoto, "Are Max Min Suitable Operations for Fuzzy Control Methods?", in Proc. 6th IFSA Congress, Vol. 1, pp. 497-500, 1995.

- [9] M. Sugeno and M. Nishida, "Fuzzy Control of Model Car", Fuzzy Sets and Systems 16, pp. 103-113, 1985.
- [10] B. Graham and R. Newell, "Fuzzy Adaptive Control of a First-order Process", Fuzzy Sets and Systems, 31, pp. 47-65, 1989.
- [11] G. Kang and W. Lee, "Design of Fuzzy Parameter Adaptive Controllers", in Proc. 6th IFSA Congress, pp. 609-612, 1995
- [12] H. Nomura, I. Hayashi, and N. Wakami, "A Self Tuning Method of Fuzzy Control by Descent Method", in Proc. 4th IFSA Congress, pp. 155-158, 1991
- [13] T.J. Procyk and E.H. Mamdani, "A Linguistic Self Organizing Process Controller", Automatica, 15-1, pp. 15-30, 1979.
- [14] M. Sugeno and G. Kang, "Structure Identification of Fuzzy Model", Fuzzy Sets and Systems 28, pp. 15-33, 1988
- [15] K. Tanaka and M. Sugeno, "Stability Analysis and Design of Fuzzy Control Systems", Fuzzy Sets and Systems 45, pp. 135-156, 1992.
- [16] G. Kang and W. Lee, "Design of State Fuzzy Controllers and Observers", in Proc. 4th IEEE Conference on Fuzzy Systems, pp. 1355-1360, 1995

저 자 소 개



강 근 택

1953년 3월 12일생

1976년 서울대학교 응용물리학과 (공학사)

1984년 일본 동경공업대학 시스템과학과(공학석사)

1987년 일본 동경공업대학 시스템과학과(공학박사)

1987-현재 부산수산대학교 전자공학과 부교수.