

퍼지 이론과 소프트 컴퓨팅

강 근 택(姜 根 澤)

부경대학교 전자정보통신공학부

전화 : (051) 620-6464 / 팩스 : (051) 620-6460

Fuzzy Theory and Soft Computing

Geuntaek Kang

Dept. of Electronic Engineering, Pukyong National University

E-mail : gtkang@dolphin.pknu.ac.kr

본 글은 soft computing에 대해 설명하고, soft computing의 주요 분야인 퍼지 이론에서 2가지 형태의 퍼지모델의 융합과, 퍼지이론과 진화적 알고리즘의 융합에 대해 설명한다.

1. Soft Computing[1]

soft computing이라는 용어는 1991년 Zadeh 교수가 미국 Berkeley 대학에 BISC (Berkeley Initiative in Soft Computing) 연구소를 설립할 때 처음 사용되었다.

soft computing은 기본적으로 높은 정밀성을 요구하지 않는 computing이다. 가령 헬리콥터의 자동 조종에서 “빨리 착륙하라”는 명령이 나왔을 경우, 지면을 보는 카메라에 호수가 보인다는지 풀숲이 보인다는지 할 때 어떻게 조종할 것인지 빨리 판단을 하여야 한다. 이 때 카메라에서 나오는 모든 정보를 정밀하게 분석할 필요는 없다. 단지 지면만 찾으면 된다. 즉, “빨리 착륙하라”는 모호하면서 고급의 명령에 대해 상황 판단을 하고 답을 내는데는 매우 높은 정밀성을 요구하는 계산 같은 것은 필요하지 않는 것이다. 그러면 착륙할 지면을 찾기 위해서는 카메라의 정보를 어느 정도까지 정밀하게 분석하면 되는가를 알아야 한다. 이러한 문제를 다루는 것이 soft computing이다.

실제 세계는 정확하지도 않으며 변하기 쉬운 성질을 갖고 있다. 이러한 실제 상황에서 정확함이라든지 확실함을 얻기 위해서는 cost가 필요하다. 즉 정확한 결

과를 얻는다든지 확실성을 높이기 위해서는 그 만큼의 cost를 지불하여야 한다. 이 때 cost라는 것은 금전뿐만 아니라, 주파수도 될 수 있으며, 시간도 될 수 있고, 그 외에 무엇인가도 될 수 있다. 이러한 cost를 낮추기 위해 우리는 정확성 또는 확실성을 어느 정도 까지 요구할 것인가를 판단하여야 한다.

즉 soft computing의 기본원리는 「다루기 쉬우며 (tractability), 강건하고(robustness), 낮은 cost를 달성하기 위해 불확실성(uncertainty)이 어느 정도까지 허용 가능한 가를 찾는 문제」이다. tractability를 만족하여야 문제를 해결할 수 있으며, tractability가 해결되고 난 후에야 cost 문제를 다룰 수 있다. robustness는 달성하고자 하는 목표와 관계된다. 항상 모든 것에 민감하도록 목표를 잡을 필요는 없는 것이다.

soft computing의 역할 모델(role model)은 인간의 지성(human mind)이다. 인간은 수치로 표현할 수 없는 부정확(imprecise)하고 불확실(uncertain)한 정보를 취급할 수 있는 탁월한 능력을 갖고 있다. 현재 매우 정밀한 계산 능력을 갖고 있는 컴퓨터도 이러한 인간의 능력을 따라가지 못하고 있다. 역할 모델이란 컴퓨터가 이러한 인간의 능력에 가까워지도록 만드는 것이다.

soft computing에 사용되는 분야에는 크게 퍼지이론, 신경망(neural networks), 확률적 추론(probabilistic reasoning)의 3개가 있다. 이 3분야는 각기 다른 특색을 갖고 있으며 서로 협력관계에 있다. 확률적 추론에 사용되는 수법으로는 진화적 알고리즘(EA), 카오스(Chaos) 시스템 등이 있다. 퍼지이론은 부정확

(imprecision), 신경망은 학습, 확률적 추론은 불확정 (uncertainty)에 각각 주로 관계하고 있다. 이 3분야는 서로 경쟁관계에 있는 것이 아니고 서로 겹쳐지기도 하여 큰 영역을 만드는 협력관계에 있다.

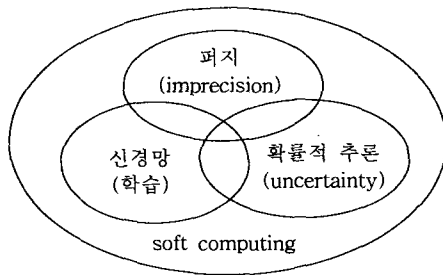


그림 1 soft computing

2. 퍼지이론

퍼지 이론은 soft computing에서 가장 중요한 분야이다. 퍼지이론은 Mamdani에 의해 제어에 응용될 수 있음이 보여진 후 초기에는 퍼지제어의 연구가 주를 이루었으나 요즘은 퍼지제어보다 퍼지이론을 이용한 지능시스템의 연구가 더욱 중요시되고 있다. soft computing이 사용될 수 있는 지능 시스템으로는, 필기 문자인식, 음성인식, 자연언어이해, 영상이해 및 처리, 진단, 정보검색, 학습, 추론, 계획, 문제해결, 의사결정 등이 있다.

퍼지제어에서도 이제까지는 입력신호가 스칼라 또는 다변수인 시스템을 대상으로 하였지만, 앞으로는 입력이 그와 같은 간단한 신호가 아니고 영상(Image) 또는 그 이상으로 복잡한 것이 될 것으로 본다. 그러한 예로 자동차 항공기 등의 운전, 로봇제어, 의학 등이 있다. 입력이 영상 또는 그 이상으로 복잡할 경우 대량의 정보 모듈을 이용하여 판단하는 것이 아니고, 많은 정보를 버리고 요약된 일부분의 정보로 판단하여야 한다. 그러나 광대한 정보로부터 필요한 정보를 찾아내는 것은 매우 어렵다. 퍼지이론뿐만 아니고 다른 soft computing 분야들과 협조하는 지능 시스템 기법이 필요하다.

2.1. 정수형 퍼지모델과 TSK퍼지모델의 협조

퍼지이론이 가장 많이 사용되는 형태는 if - then-의 퍼지규칙들로 구성되는 퍼지모델이다. 퍼지규칙에는 결론부의 형태에 따라 다음 3가지 종류가 있다.

$$\text{If } x_1 \text{ is } G_1^i, x_2 \text{ is } G_2^i, \dots, x_n \text{ is } G_n^i \quad (1)$$

$$\text{then } y = a^i$$

$$\text{If } x_1 \text{ is } G_1^i, x_2 \text{ is } G_2^i, \dots, x_n \text{ is } G_n^i \quad (2)$$

$$\text{then } y = B^i$$

$$\text{If } x_1 \text{ is } F_1^i, x_2 \text{ is } F_2^i, \dots, x_n \text{ is } F_n^i \quad (3)$$

$$\text{then } y = a_0^i + a_1^i x_1 + \dots + a_n^i x_n$$

결론부가 (1)의 퍼지규칙에서는 정수(singleton), (2)는 퍼지집합, (3)은 입출력의 선형식이며, 이 3가지의 형태는 각기 다른 특색을 갖고 있다.

(1) (또는 (2))의 형태의 규칙들로 구성되는 퍼지모델(지금부터 정수 퍼지모델이라 칭한다)은 언어적 형태이므로 규칙을 이해하기 쉽고 규칙의 파라미터 조절도 비교적 쉽다. 하지만 입출력 데이터만으로는 만들기는 어렵다.

반면에 (3)의 형태의 규칙들로 구성되는 퍼지모델(지금부터 TSK퍼지모델이라 칭한다)은 입출력의 수치 데이터로부터 만들어질 수 있으며, 복잡한 비선형 시스템을 적은 수의 규칙으로 높은 정밀도로 표현할 수 있다. 또한 페루프 시스템의 안정성을 보장하는 제어기 설계도 용이하다[2]. 그러나 결론부가 선형식이므로 각 규칙의 이해가 어려우며 파라미터를 손으로 조정하기가 어렵다.

따라서 이 두 종류의 퍼지모델의 장점을 모두 이용할 수 있는 방법 즉 두 퍼지모델을 융합하는 기법이 필요하다. 이제부터 TSK 퍼지모델을 정수 퍼지모델로 변환시키는 방법을 설명하고, 정수 퍼지모델과 TSK 퍼지모델이 서로 협조하는 기법의 응용 예를 보인다.

지금부터 TSK 퍼지모델의 전제부 퍼지집합은 그림 2(a)와 같은 pseudo-trapezoidal 멤버십 함수를 갖도록 하며, 정수퍼지모델의 전제부 퍼지집합은 그림 2(b)와 같은 삼각형 멤버십 함수를 갖도록 한다.

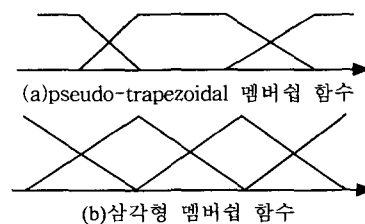


그림 2 퍼지집합

TSK퍼지모델과 같은 입출력 mapping을 갖는 정수 퍼지모델로 만들기 위해서는, 먼저 입력 공간 $X_1 \times$

$X_2 \times \dots \times X_n$ ($x_j \in X_j$)을 퍼지 부분공간들로 퍼지 분할하고, 각 퍼지부분 공간에 정수의 출력을 할당하여야 한다. 입력공간을 퍼지분할하기 위해서는 각 입력 변수에 그림(b)와 같은 퍼지집합들을 만들어야 한다. 그 방법은 TSK 퍼지모델에서 그 변수가 어디에 소속되어 있는지 즉 전제부에만 있는지, 결론부에만 있는지, 아니면 둘 다에 있는지에 따라 틀리며, 그림 3에 그 방법을 보인다. 그림 3에서 변수 x_j 의 작업구간이 $[a_j, \beta_j]$ 이고, 점선은 TSK퍼지모델에 있었던 퍼지 집합이며, 실선은 점선의 퍼지집합을 토대로 정수 퍼지모델에 만든 퍼지집합이다. 그림3(a)는 변수 x_j 가 TSK퍼지모델에서 결론부에만 있는 경우, 그림3(b)는 전제부에만 있는 경우, 그림3(c)는 결론부와 전제부 모두에 있는 경우이다.

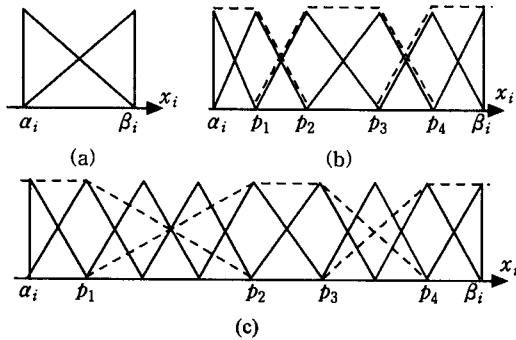


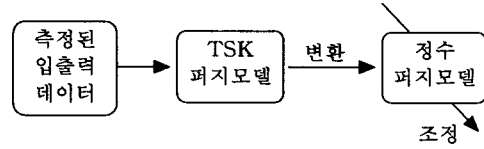
그림 3. TSK퍼지모델의 퍼지집합으로부터 정수퍼지모델의 퍼지집합을 만드는 방법

변수 x_j ($j=1, 2, \dots, n$)에 N_j 개의 퍼지집합이 정의되면 총 $\prod_{j=1}^n N_j$ 개의 퍼지 부분공간이 만들어지며 각 퍼지 부분공간에 다음과 같이 실수값의 출력을 할당하여 정수 퍼지규칙을 만든다. 먼저 i 번째 퍼지 부분공간이 퍼지집합 $G_1^i, G_2^i, \dots, G_n^i$ 들로 구성되었다 하자. 단, G_j^i 는 x_j 위의 퍼지집합이며 $G_j^i(\hat{x}_j) = 1$ 라 하자. 그러면 $(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$ 에 대한 출력을 TSK 퍼지모델로부터 추론하고 그 값이 a^i 라면, 그 값을 i 번째 퍼지 부분공간에 할당하여 (1)과 같은 정수 퍼지 규칙이 만들어진다.

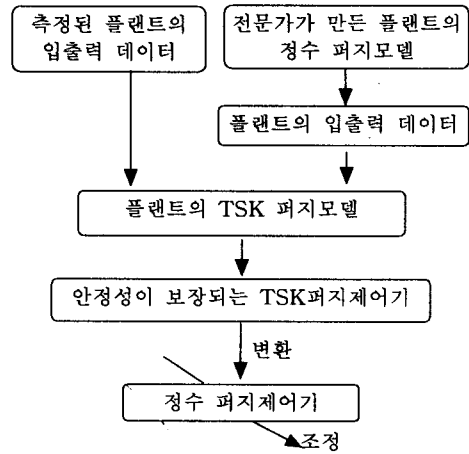
변환된 정수 퍼지모델과 원래 TSK퍼지모델과의 일치성은, 그림 3(a)와 (b)의 경우에는 정확하게 일치하지만, 그림 3(c)의 경우 즉 변수가 TSK 퍼지모델의 전제부와 결론부 모두에 있는 경우에는 두 퍼지모델의 출력에 오차가 생긴다. 오차가 생기는 부분은 그림

3(c)에서 $[p_1, p_2]$ 과 $[p_3, p_4]$ 구간 같이 TSK퍼지모델에서 비선형 부분이며, 그 구간에 퍼지집합을 많이 만들수록 오차는 줄어든다.

지금까지 설명한 것을 응용하는 방법을 그림 4에 보인다.



(a)퍼지 모델링에 응용



(b)퍼지제어기 설계에 응용

그림 4. TSK퍼지모델과 정수퍼지모델 융합의 응용

퍼지모델링의 응용의 하나로 반도체 소자 특성의 모델링을 보이겠다. 반도체 소자들 중에는 매우 강한 비선형성과 복잡한 동 특성을 갖는 것들이 있다. 그러한 소자들의 경우, 물리적 해석에 의해 구해지는 모델은 정확하게 동 특성을 잘 나타내지만 그러한 모델은 구하기가 불가능하거나 너무 복잡하여 사용하기가 어려워 soft computing 기법이 요구된다. 그러한 경우 앞에서 설명한 정수 퍼지모델과 TSK 퍼지모델의 협조에 의한 방법을 사용할 수 있다. 예로써 GaAs FET의 모델을 만들었다. 출력은 드레인-소스 전류이며 입력 변수는 드레인-소스 전압과 게이트-소스 전압이다. 먼저 실험에서 구한 입출력 데이터로 TSK 퍼지모델을 작성하였다. 그림 5에 그 모델을 보이며, 그 모델에서 추론한 출력을 그림 6에서 점선으로 보인다. 그림6에서 원으로 표현된 것은 실제 측정 데이터이다. TSK퍼지모델로부터 출력은 낮은 드레인-소스 전압과 낮은 게

이트-소스 전압 구간에서 오차가 심함을 볼 수 있다. 특히 게이트-소스 전압이 -2.5V인 경우 출력이 단조 증가 특성에 위배되고 있으며 제로 입력에 대해 출력이 제로가 되지 않고 있다. 이것을 교정하기 위해 앞에서 설명한 방법으로 그림 5의 퍼지모델을 정수 퍼지모델로 변환시키고 오차가 있는 부분의 결론부 파라미터를 조정하였다. 그림 7에 수정된 정수 퍼지모델을 보이며 그 모델로부터 얻은 출력은 그림 5에서 실선으로 보인다. 그림7에서 괄호안의 숫자는 조정되기 전의 파라미터이다.

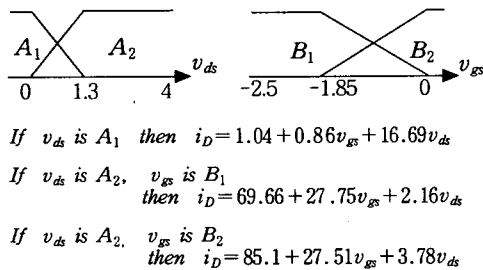


그림 5 GaAs FET의 퍼지모델

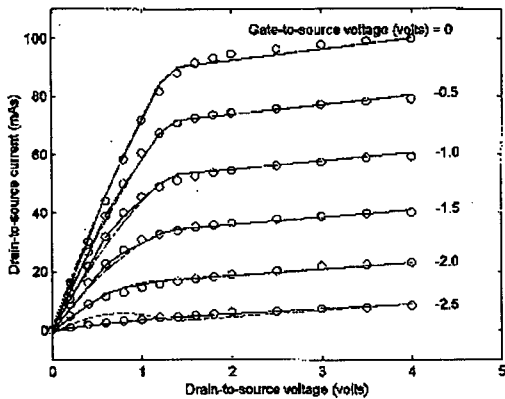


그림 6 TSK퍼지모델과 정수퍼지모델의 추론결과

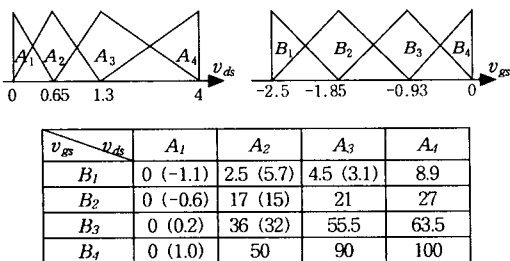


그림 7. GaAs FET의 정수 퍼지모델

3. 퍼지이론과 진화적 알고리즘의 협조

soft computing의 확률적 추론 분야에서 가장 많이 사용되는 기법이 진화적 알고리즘(EA: Evolutionary Algorithm)이다. EA는 생물의 진화 알고리즘을 공학에서 응용한 것으로, 유전적 알고리즘(Genetic Algorithm), 진화적 전략(Evolutionary Strategy), 진화적 프로그래밍(Evolutionary Programming), 유전적 프로그래밍(Genetic Programming) 등이 있으나 기본 형태는 그림 8과 같다.

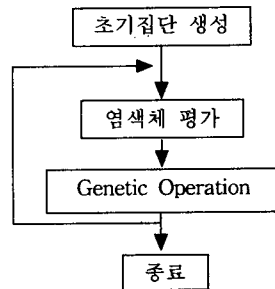


그림 8. 진화적 알고리즘

이제까지 퍼지이론과 EA의 협조에 관한 연구는 C.L.Karr[3]에 의해 처음 발표된 후 많은 연구가 발표되고 있다. 그 형태로는 퍼지에 EA를 사용하는 것과 EA에 퍼지를 사용하는 두 종류가 있다.

3.1. 퍼지에 EA를 이용

퍼지 모델링에 EA를 사용한다. 퍼지 모델링에서 EA를 사용하여 선택할 수 있는 것으로는 입력공간을 분할하는 방법, 입력변수, 멤버십 함수의 형태, 멤버십 함수 파라미터 등이 있다[3][4][5].

또한, 퍼지규칙들을 EA를 이용하여 선택할 수도 있다. 이 때 퍼지 규칙을 code화하는 방법에는 그림 9와 같은 방법들이 있다. 먼저 그림9(a)-(c)는 입력변수로 x_1 과 x_2 2개가 있고 y 가 출력인 퍼지규칙을 code화한 것으로 규칙 전체부에 항상 입력변수 2개 모두 있는 경우이다. 그림9(a)는 1개의 규칙을 1개의 염색체로 code화하고, 각 염색체를 평가하여 공헌도가 큰 염색체들을 선택하는 것으로 퍼지 classifier에 사용된다[4]. 그림9(b)는 규칙전부를 한 개의 염색체로 표현한 것으로 하나의 염색체가 하나의 퍼지시스템이 된다[3]. 그림9(c)도 한 개의 염색체가 하나의 퍼지시스템이 되지만, 퍼지규칙들이 입력공간 전체를 커버하지 않는 것이 그림9(b)와 틀린다.

그림 9(d),(e)는 5개의 입력변수(x_1, \dots, x_5)와 출

력 y 로 구성되는 규칙을 검색체로 coding한 것으로, 그림 9(a)-(c)와 틀리는 것은 규칙 전체부에 입력변수 5개 모두 있지 않을 수도 있는 것이다. 그림9에서 *부분은 해당 변수가 전체부에 없음을 나타낸다. 그림9(d)는 1개의 규칙을 1개의 검색체로 coding한 것이며, 그림9(e)는 한 개의 검색체가 한 개의 퍼지시스템이 되는 경우이다[6].

x_1	x_2	y
A ₃	B ₂	C ₁

(a)

y	x_1		
x_2			

(b)

x_1	x_2	y	x_1	x_2	y	...	x_1	x_2	y
A ₂	B ₅	C ₃	A ₄	B ₁	C ₂	...	A ₁	B ₃	C ₁

(c)

x_1	x_2	x_3	x_4	x_5	y
A ₃	*	C ₁	*	E ₂	F ₂

(d)

A ₂	*	...	*
B ₄	B ₂	...	*
*	C ₁	...	C ₃
D ₁	*	...	D ₃
E ₅	*	...	E ₂
F ₂	F ₅	...	F ₃

(e)

그림 9 퍼지규칙의 coding

그림 10는 입력공간을 분할하는 방법을 선택할 때 coding하는 방법을 보인다.

x_1	x_2	x_3	x_4	x_5	x_6
2	0	2	3	0	0

그림 10 입력공간의 분할을 위한 coding

3.2 EA에 퍼지를 이용

EA이 genetic operation에서 교배율, 돌연변이율, 집단 크기 등과 같은 파라미터는 EA의 성능에 크게 영향을 미친다. 또한 상황에 따라 그 파라미터의 최적

값들이 변할 수도 있다. 그러한 파라미터의 최적값들을 구하는 정성적인 지식은 갖고 있는 경우 퍼지시스템을 사용할 수 있다[7]. 가령 예로써, 집단의 평가값의 변동이 적을 경우 local 최적해에 빠졌을 가능성이 있으므로 돌연변이율을 증가시켜 본다. 또는 집단에서 평가값이 높은 검색체의 수가 적을 경우 집단의 크기를 줄이고, 반대로 평가값이 높은 검색체가 많을 경우 집단을 크게 만들어 본다. 이러한 과정을 퍼지 시스템으로 표현하여 파라미터의 최적값을 구한다.

EA의 종료 판단에 퍼지시스템을 이용할 수 있다. 최적해를 알 수 없을 경우 EA의 수축 상황에 따라 최적해를 예측하면서 종료를 판단할 수 있다.

EA에서 초기 집단의 생성에도 퍼지시스템을 이용할 수 있다. 검색체 평가를 실제 실험으로 구할 경우 수축이 늦어지면 상당한 시간 및 노력이 필요하다. 따라서 사전에 최적해에 대한 정성적인 지식이 있을 경우 초기집단을 랜덤하게 구하기보다는 퍼지이론의 수법을 이용하여야 한다.

그 외에 검색체의 평가에도 퍼지 시스템을 이용할 수 있다.

참고문헌

- [1] L.A. Zadeh, "Soft computing", Journal of Japan Society fo Fuzzy Theory and Systems, Vol.7-2, pp.262-269, 1995
- [2] 강근택, 이원창, "TSK퍼지모델로부터 TSK퍼지제어기의 설계" 대한전자공학회 논문지-S 35-11, pp.53-67, 1998.
- [3] C.L. Karr, "Design of an adaptive fuzzy logic controller using a genetic algorithm", Pro. of the 4th International Conference on Genetic Algorithms, pp.450-457, 1991
- [4] M. Valenzuela Rendon, "The fuzzy classifier system", Proc. of 4th ICGA, pp.346-353, 1991
- [5] K. Shimojima, T.Fukuda, Y.Hasegawa, "Self tuning fuzzy modelling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm", Fuzzy Sets and Systems, vol. 71-3, pp.295-309, 1995.
- [6] T. Hashiyama, T.Furuhashi, Y.Uchikawa, "A study finding fuzzy rules for semiactive suspension controllers with genetic algorithm", Proc. of ICEC '95, pp.279-282, 1995.
- [7] M.A. Lee, T.Takagi, "Dynamic control of genetic algorithms using fuzzy logic techniques", Proc. of 4th ICGA, pp.76-83, 1993