

유전 알고리즘을 이용한 퍼지 규칙 베이스의 자동생성

(Auto-Generation of Fuzzy Rule Base Using Genetic Algorithm)

朴世喜*, 金容虎*, 沈貴寶**, 全洪兌***

(Sae Hie Park, Yong Ho Kim, Kwee Bo Sim, and Hong Tae Jeon)

要 約

퍼지 논리 제어기는 구조가 간단하여 실시간 제어가 가능하고 시스템의 구조나 동적 특성을 필요하지 않는 등의 많은 잇점 때문에 최근에 그 응용 범위가 확대되고 있다. 그러나 퍼지 논리 제어기의 제어규칙은 일반적으로 전문가가 구성해 주기 때문에 제어대상이 시변 또는 비선형 시스템인 경우 정밀한 제어규칙을 만드는 일은 용이한 일이 아니다. 이에 본 논문에서는 최근에 최적화 알고리즘으로 크게 주목을 받고 있는 유전 알고리즘(Genetic Algorithm)을 이용해 제어 목적에 적합한 퍼지 제어규칙을 자동 생성할 수 있는 방법을 제안한다. 그리고 2자유도 로봇 매니퓰레이터의 궤적 제어에 적용하여 제안된 방법의 유효성을 입증한다.

Abstract

Fuzzy logic rule based controller has many desirable advantages, which are simple to implement on the real time and need not the information of structure and dynamic characteristics of the system. Thus, nowadays, the scope of the application of the fuzzy logic controller becomes enlarged. But, if the controlled plant is a time-varying/nonlinear system, it is not easy to construct the fuzzy logic rules which need the knowledge of an expert. In this paper, an approach by which the logic control rules can be auto-generated using the genetic algorithm that is known to be very effective in the optimization problem will be proposed and the effectiveness of the proposed approach will be verified by computer simulation of the 2 d.o.f. planner robot.

I. 서 론

1965년 Zadeh 교수에 의해 발표된 퍼지 논리(fuzzy logic)는 애매한 정보를 효과적으로 처리할 수

있어 최근에 그 응용 범위가 확대되고 있다. 특히 퍼지 논리를 이용한 퍼지 논리 제어기(fuzzy logic controller)는 제어 대상의 정확한 수학적 모델링이 불필요 하고 언어적인 제어 그리고 병렬처리 계산을 가능케 하는 등의 잇점을 갖는다.

기본적으로 퍼지 제어기는 퍼지화기(fuzzifier), 규칙 베이스(rule base), 퍼지 추론기(fuzzy inference engine) 그리고 비 퍼지화기(defuzzifier)로 구성되어 있다. 퍼지화기는 입력 데이터를 적당한 퍼지 집합으로 변환하는 기능을 갖고 있으며, 규칙 베이스는

*學生會員, ***正會員, 中央大學校 電子工學科
(Dept. of Elec. Eng., Chungang Univ.)

**正會員, 中央大學校 制御計測工學科
(Dept. of Control & Instrumentation Eng., Chungang Univ.)

接受日字 : 1991年 12月 6日

전문가의 경험 등을 "IF-THEN"형의 문장으로 구성한 지식 베이스이다. 그리고 퍼지 추론기는 규칙 베이스의 규칙을 이용 퍼지 입력에 대한 퍼지 출력을 추론하는 기능을 가지며 비퍼지화기는 추론기에서 출력한 값을 crisp한 제어값으로 변환시키는 기능을 갖는다(그림2 참조).

그러나, 퍼지 제어기의 구성에 있어 어려운 점은 규칙 베이스를 구성하는 일이다. 전문가의 지식이나 실험에 의해 규칙 베이스를 구성할 수 있지만 제어 대상이 고도의 비선형일 경우, 그 구성에 많은 어려움을 갖는다. 설사 규칙 베이스를 구축하였다 하더라도 그것이 과연 최선인지는 알 수 없다.

이러한 어려움을 보완하기 위해 제어 규칙들을 자동으로 생성할 수 있는 방법들에 대한 연구가 최근에 활발히 진행되고 있다. 이러한 연구들은 신경망(neural network)과의 융합(fusion)^{[1][2]}을 통한 규칙의 자동생성과 자기동조(self-tuning)^[3]에 의한 자동생성 방법으로 대분할 수 있다.

신경망과의 융합방식은 제어 규칙 언어 변수들을 소속함수(membership function)등을 신경망 구조와 학습 능력을 이용, 제어 대상에 맞게 조정하는 방식으로, 학습 알고리즘(learning algorithm)으로는 오차 역전파 학습 알고리즘(error back propagation learning algorithm)을 널리 응용한다. 그러나 이 방식은 교사신호(teaching signal)의 효과적인 결정과 local minimum에 빠지기 쉬운 역전파 알고리즘의 어려운 점을 극복할 수 없다.

한편 자기동조 방식은 실행 측정표(performance index)를 참고로 하여 제어대상의 현재 및 과거 상태에서부터 새로운 퍼지 제어 규칙과 환산계수(scaling factor)를 만들며, 동시에 제어규칙을 만들어 주는 방법이다. 실행 측정표는 현재 사용된 제어규칙의 기여도에 따라 제어규칙의 후건부를 강화 또는 약화시켜 제어규칙을 만들면서 수정하게 하는 참고표가 된다. 그러나 실행 측정표를 정해주는 기준은 전문가의 경험과 지식에 의해 정해주게 되므로 이 또한 효율적인 방식이라 할 수 없다.

본 논문에서는 상기의 어려움을 극복하고 제어 대상에 적합한 최적의 규칙을 자동 생성할 수 있는 방법을 제시한다. 제안된 방법은 자연계의 진화이론을 바탕으로 이루어진 유전 알고리즘(genetic algorithm)^[4]을 기본으로 하고 있으며 제어기의 규칙 베이스와 퍼지 변수들의 전체 집합(universe of discourse)에 입력 변수들의 위치를 결정하는 환산계수의 최적값을 유전 알고리즘에 의해 결정한다.

그리고 본 논문의 구성은 다음과 같다. 2장에서는

유전 알고리즘을, 3장에서는 유전 알고리즘을 이용한 퍼지 제어 규칙의 자동생성 방식에 대해 설명한다. 4장에서는 그 응용 예로 2자유도의 로봇트 매니플레이터의 궤적 제어를 위한 fuzzy-genetic 제어기를 제안하고, 5장과 6장에서는 컴퓨터 시뮬레이션과 결론을 논한다.

II. 유전 알고리즘(Genetic Algorithm)

유전 알고리즘은 자연 생태계의 '적자 생존(Survival of the fittest)'의 원리를 이용하여 1970년대 미국 J. H. Holland 교수에 의해 정립된 최적화(optimization) 알고리즘이다^{[4][5][6]} 이 알고리즘은 미지의 함수 $Y=G(X)$ 의 최적 해를 발견하는 simulated evolution 형의 탐색(search) 알고리즘으로 그 과정은 다음과 같이 요약할 수 있다.

[단계 1] 10진수 인자1과 0으로 구성된 string 형태로 재 표현하기 위한 string의 길이와 초기 string을 선정한다.

[단계 2] 단계1의 string을 이용해 초기 집단(population)을 구성함.

[단계 3] 집단의 각 string을 10진수 인자로 역표현하여 함수 $G(X)$ 의 적합도(fitness)를 구함.

[단계 4] 적합도에 따라 해당 string들을 선택함.

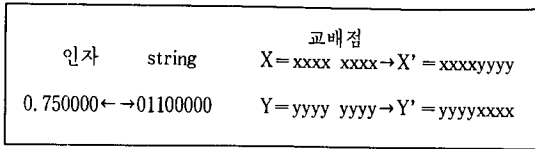
[단계 5] String 간의 교배(crossover)와 돌연변이(mutation)등의 진화 과정을 거쳐 새로운 집단을 구성함.

[단계 6] 단계3 이하의 과정을 반복함.

그림1(a)는 단계1에서의 인자의 string 표현과 그 역 표현(즉 string 표현으로부터 인자로의 표현)을 보여주고 있으며 단계3에서의 적합도는 함수에 대한 최적 정도를 의미한다. 그리고 단계4에서의 string 선택은 적합도의 높은 정도에 의해서 이루어진다. 그림1(b)는 8비트인 두 string X와 Y의 교배를 보여주고 있다. 이때 교배점(crossing site)은 임의의 난수로 선택하게 된다. 돌연변이는, 교배와 동시에 행해지며, 임의의 비트를 선택하여 비트 전환으로 이루어진다(단계5). 한 예로 $X=1001$ 일때 임의로 선택된 비트가 3번째이면 돌연변이를 한 새로운 x' 는 1011이 된다.

이와 같은 유전 알고리즘의 특성은 다음과 같다.

1. 유전 알고리즘은 해공간(solution space)에서 단일 해를 사용하는 것이 아니라 해 집단(population of solution space)을 이용한다.
2. 유전이론은 'blind' 하다. 즉 최적화 함수 정보(미분가능성, 연속성 등)를 필요치 않는다.



(a) 인자와 string (b) string X와 Y의 교배

그림 1. 인자의 string 재표현과 그 교배
 Fig. 1. Representation of parameter into string and its crossover.

3. 유전 알고리즘은 global 한 최적해의 발견을 가능케 한다.

Ⅲ. 퍼지논리 제어 규칙의 자동 생성

일반적으로 퍼지 논리 제어기의 구성은 다음 그림 2와 같다. GE는 Error의 전체집합(universe of discourse)을 정하는 환산계수(scaling factor)이고 GC는 Change of Error의 환산계수이다. 그리고 C는 제어기의 명령이 되며 R은 제어 대상의 명령에 대한 결과이다.

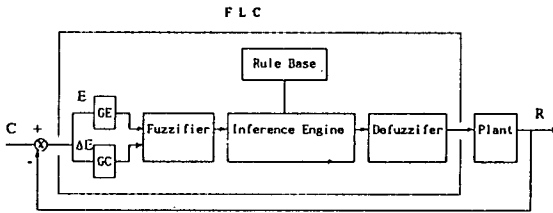


그림 2. 퍼지 제어기
 Fig. 2. Fuzzy logic controller.

그림2의 퍼지 논리 규칙 베이스는 식(1)과 같은 "If-Then" 형식의 규칙들로 구성되어 있다. 그러나 제어 대상에 따라 적절한 규칙을 구성하는 일은 용이하지 않고 특히 제어 대상이 시변·비선형일 경우 그 구성은 매우 어렵다.

$$\underbrace{\text{If } X \text{ is } A_i \text{ and } Y \text{ is } B_i}_{\text{전건부}} \text{ then } U \text{ is } C_i \underbrace{\quad}_{\text{후건부}} \quad (i=1 \dots N) \quad (1)$$

여기에서 X, Y와 U는 각각 입력 및 출력 변수들이고, A_i, B_i, C_i 는 그림3과 4의 소속함수들을 갖는 퍼지 변수들이다. 그림의 소속함수는 환산계수에 의해 정규화된 범위를 나타낸다. (일반적으로 소속함수의

모양은 종형 (bell type)과 이등변삼각형 (isosceles triangle type) 등 들이 있으나 제어 시스템에서는 연산이 간단한 이등변삼각형을 많이 사용하고 있다. 이에 본 절에서는 각 퍼지 변수들이 이등변 삼각형 모양의 소속함수를 갖는다고 가정한다).

식(1)의 N개로 구성되어 있는 규칙 베이스를 제어 대상에 맞게 조정할 수 있는 방법은 각 규칙의 전건부(antecedent)와 후건부(consequent)의 퍼지 변수 소속함수(membership function)들을 변형하거나 입력들의 (universe of discourse)상의 설정을 위한 환산계수를 조정하여 이루어질 수 있다. 본 논문에서는 제어 환경에 맞는 최적 퍼지 규칙 베이스를 자동 생성하기 위해 규칙 베이스의 후건부 $C_i (i=1 \dots N)$ 의 소속함수들과 환산계수를 주요 조정대상 혹은 최적화 인자로 삼는다.

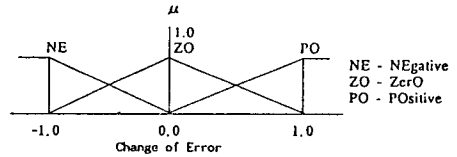
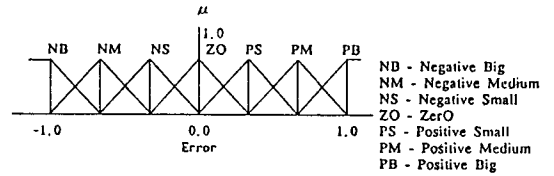


그림 3. 전건부의 소속함수
 Fig. 3. Membership function of antecedent.

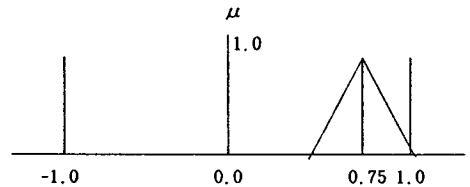


그림 4. 후건부 언어 변수의 소속 함수
 Fig. 4. Membership function of consequent.

환산계수(GE와 GC)는 입력 변수를 전체집합 상에 설정하기 위한 파라미터로 퍼지 변수의 소속함수에 못지 않게 퍼지 논리 제어기의 성능에 큰 영향을 미친다. 이에 (10진수 형의) 환산계수를 최적화 대상의 하나로 선정하고, 유전 알고리즘을 적용하기 위

해 이들 값들을 unsigned binary로 표현한다.

그러나, 또 하나의 최적화 대상인자인 후건부 소속 함수들을 특정 지우는 일은 환산계수처럼 단순하지 않다. 일반적으로 전체집합 상에서 이등변 삼각형의 소속함수를 특정 지우는데에는 3개 꼭지점들의 위치들을 필요로 하지만, 삼각형 소속함수의 밑변 길이를 일정하게 유지하면 상단 꼭지점 만으로도 그 소속함수를 특정지울 수 있다. 이에 전체집합 상의 상단 꼭지점 위치를 다른 하나의 최적화 대상으로 삼으며, 그 위치를 규칙 베이스의 적합도에 따라 유전 알고리즘을 이용 조정한다. 이를 위해 꼭지점의 상단 위치들을 signed binary로 표현한다.

그림5는 binary로 표현된 환산계수와 (한 규칙 베이스를 구성하는 후건부 퍼지 변수들의) N개 소속 함수들을 하나의 string으로 합성한 예를 보여 주고 있다. 이 string을 이용 유전 알고리즘에 의한 최적화 과정이 이루어진다.

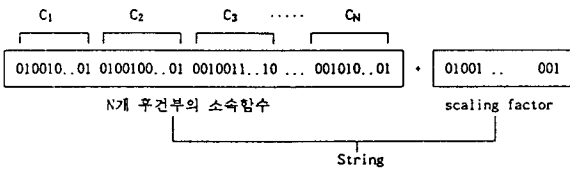


그림 5. 하나의 string으로 합성한 환산계수와 N개의 소속함수

Fig. 5. A string representing a scaling factor and N membership functions.

한편, 적합도는 퍼지 논리 규칙의 성능 평가의 기준이 되는 함수이며 집단내에서 진화 대상이 요구되는 string을 선택하는 기준이 된다. 따라서 그림 2의 시스템에서의 평가 함수는 다음과 같이 설정한다.

$$Fitness = \frac{K_1}{Error} + \frac{K_2}{Change\ of\ Error} + \frac{K_3}{Energy} \quad (2)$$

여기에서 Energy는 plant에 가해지는 입력이며 K1, K2, K3는 상수이다.

식(2)에서 높은 적합도는 Error 및 Change of Error, Energy의 감소를 의미하기 때문에 집단내에서의 높은 적합도를 갖는 string은 제어 목적에 좀 더 접근한 제어 규칙들로 구성된 규칙 베이스를 의미한다.

2. 진화에 의한 규칙 베이스의 생성

최적화 대상 인자들 및 이들의 string 표현법을 결정 후 유전 알고리즘에 한 규칙 베이스의 최적화 과정은 다음과 같이 이루어진다.

1) 초기 집단의 구성

초기 집단을 구성하기 위해서는 우선 초기 string을 결정해야 한다. 초기 string은 초기에 구성된 규칙 베이스와 초기 환산계수값을 이용 구성할 수 있고, string의 표현법 (이진수 표현 및 string length)에 맞추어 난수 발생에 의해 구성된 M개의 string (혹은 규칙 베이스들)을 이용 하나의 초기 집단을 구성한다. 이때 집단은 임의의 M개의 규칙 베이스들을 binary로 표현한 집단을 나타낸다. 그림6은 집단의 한 예를 보여준다.

규칙들의 후건부의 C_i 환산계수

규칙 베이스 1	001001000...100010010100 1010100...1010
규칙 베이스 2	001001001...010010010101 0010010...1101
규칙 베이스 3	010010010...010001000100 0100000111011
	001010100...000101101011 1011101...1001
	...
규칙 베이스 M	001001010...110101000100 1010011...0101

그림 6. 초기 String 집단

Fig. 6. Initial population of strings.

2) 진화에 의한 새로운 규칙 베이스 구성

초기 string 집단의 각 행 (혹은 string)은 그림6에서 보여지는 것처럼 하나의 규칙 베이스와 환산계수들을 나타낸다. 집단의 각 행 (혹은 string)에 의한 적합도는 다음과 같이 구해진다.

[단계 1] String을 10진수 값으로 역변환하여 퍼지 규칙들의 후건부 변수들의 소속함수와 환산계수를 결정한다(그림7 참조).

[단계 2] 구성된 규칙 베이스와 환산계수들을 이용 (그림2 참조) 퍼지 논리 제어기를 구성한다.

[단계 3] 단계2의 퍼지 논리 제어기를 이용하여 Error, Change of Error 및 Energy 그리고 식 (2)를 이용하여 한 string의 적합도를 구한다.

이와 같은 과정을 집단의 모든 행 (혹은 string)에 적용하여 모든 규칙 베이스들의 적합도를 결정한다.

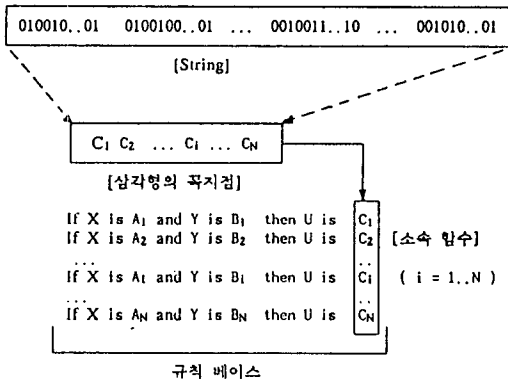


그림 7. String으로 부터 후건부 소속함수의 결정
Fig. 7. Determination of membership function from the string.

그림7은 단계1에서 한 string을 규칙 베이스의 소속함수로 역변환 하는 과정을 보여주고 있다.

모든 string(혹은 규칙 베이스)들의 적합도를 구한 후 이들 중 우성인(적합도가 높은) string(혹은 규칙 베이스)들을 이용, 다음 세대가 될 string들을 선택한다. 즉, 높은 적합도를 갖는 string에 높은 선택 확률을 주어 선택될 가능성을 보여준다.⁴⁾ 한 예로 집단 내에 4개의 string A₁, A₂, A₃, 그리고 A₄가 있고 각각의 적합도가 0.33, 0.17, 0.04, 0.46이라면 A₁은 33%, A₂는 17%, A₃은 4%, 그리고 A₄는 46%의 선택확률을 가진다. 표1은 각각으로 부터 예상할 수 있는 선택 횟수 분포의 한 예를 보여준다.

표 1. 적합도로부터의 선택
Table 1. Selection to fitness.

String	Fitness	선택확률	예상분포(1)	예상분포(2)
A ₁	0.33	30%	2	1
A ₂	0.17	17%	0	1
A ₃	0.04	4%	0	0
A ₄	0.46	46%	2	2
합	1.00	100%	4	4

선택된 string들로부터 교배(crossover)와 돌연변이(mutation)의 진화 과정⁴⁾을 거쳐 다음 세대 집단을 구성한다. 교배는 임의의 두 string간에 교차로 일어난다. 한 예로 두 string A₁과 A₄를 교배시키면 그림8과 같이 A₁'와 A₄'로 되어 다음 집단을 구성하는 string이 된다. 여기서 교배점은 임의의 난수로 선택된다.

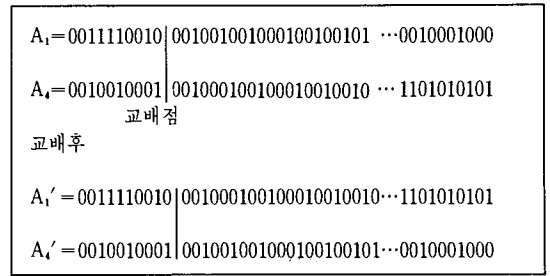


그림 8. 두 string A₁, A₄의 교배
Fig. 8. Crossover of two string A₁ and A₄.

그리고 돌연변이는 임의로 정해진 확률로 일어나며 binary string의 임의 bit 전환으로 이루어진다. 가령 돌연변이가 일어날 확률이 0.033이고 string 길이가 184비트이면 전체 비트 중 6비트 (≅184×0.033)를 비트 전환 (즉 0→1 그리고 1→0)하여 돌연변이를 수행한다. 돌연변이는 진화 과정에서 초기에 local extrimal에 빠지는 것을 방지하는 역할을 한다.⁴⁾

이와 같은 진화를 통해 구성된 새로운 집단은 이전 집단보다 적합도가 높은 집단이 되며, 상기 과정을 반복할 경우 집단내의 어느 한 string이 최대의 적합도를 갖게 되는 상태에 이르게 된다. 이때 최적의 규칙 베이스는 적합도가 가장 높은 string을 이용하여 구성하게 되며 이렇게 구성된 규칙 베이스는 plant의 Error, Changer or Error 그리고 입력 Energy를 최소화 시키는 최적의 규칙을 의미하게 된다.

IV. 로봇트 매니플레이터의 퍼지 논리 제어기예의 응용

로봇트 매니플레이터(robot manipulator)는 고도의 비선형·시변 시스템으로 이를 위한 퍼지 논리 제어 규칙을 구성하는 것은 어려운 작업이다. 본 절에서는 앞서 제안한 유전 알고리즘을 이용한 퍼지 논리기를 응용하여 로봇트 매니플레이터 논리 제어기를 구성한다.

그림9는 로봇트 매니플레이터 위치 제어를 위한 퍼지 논리 제어기의 구성을 보여주고 있다. 제어기의 유전 알고리즘 (혹은 GA)은 규칙 베이스를 구성하는 규칙들의 후건부 소속함수들과(fuzzifier의 일부분인) 환산계수들을 퍼지 논리 제어기에 할당시키고 제어기가 제어를 행하는 동안 매니플레이터의 상태, 즉 error와 change of error 그리고 추론되는 토크를 이용하여 현재 규칙 베이스의 적합도를 계산한다.

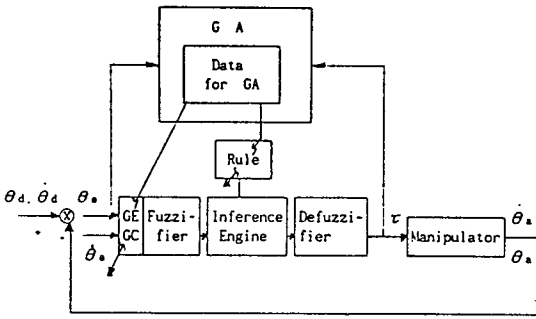


그림 9. 매니플레이터의 퍼지 논리기
 Fig. 9. Fuzzy logic controller of a robot manipulator.

퍼지화기는 singleton 기법에 의해 구성되며 crisp 한 값인 각도 $\theta_e (= \theta_a - \theta_a)$, 속도 오차 $\dot{\theta}_e (= \dot{\theta}_a - \dot{\theta}_a)$ 를 퍼지 변수로 변환한다. 추론부 (Inference Engine)은 Mamdani의 직접법을 채택하고 규칙 베이스는 다음의 "If-Then"의 문장으로 구성한다.⁷⁾⁸⁾

$$\text{If } \theta_e \text{ is } A_i \text{ and } \dot{\theta}_e \text{ is } B_j \text{, Then } U \text{ is } C_k \quad (3)$$

($i=1 \dots 7, j=1 \dots 3, k=1 \dots 21$)

여기서 A_i, B_j, C_k 는 그림3, 4의 소속함수를 갖는 퍼지 변수이다.

한편 비 퍼지화기는 제어기에서 많이 쓰는 무게 중심법 (식(4))을 사용한다.

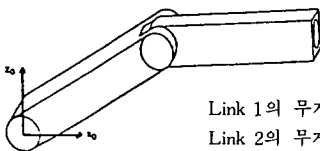
$$U^* = \frac{\int \mu_u \times U}{\int \mu_u} \quad (4)$$

여기서 U^* 는 비퍼지화된 토크이고, μ_u 는 추론 결과의 소속함수이다.

V. 컴퓨터 시뮬레이션

1. 명령 제적 및 입력

앞서 제안한 알고리즘의 효용성을 입증하기 위해 그림10의 2축 평면 매니플레이터를 시뮬레이션 대상으로 삼는다.



- Link 1의 무게 $M1=2.5\text{Kg}$
- Link 2의 무게 $M2=2.0\text{Kg}$
- Link 1의 길이 $L1=0.53\text{M}$
- Link 2의 길이 $L2=0.47\text{M}$

그림 10. 2축 평면 매니플레이터
 Fig. 10. 2 link planner manipulator.

그리고, 주행 경로는 다음과 같은 사각형으로 하였다.

- 사각형 한변의 길이 : 0.25m
- 사각형의 각 꼭지점 :
 $(0.40, 0.00, 0.20), (0.65, 0.00, 0.20),$
 $(0.65, 0.00, 0.45), (0.40, 0.00, 0.45)$

또한 유전 알고리즘의 적용을 위해서 string 길이는 368비트 (환산계수 32비트, 후건부 소속함수 336 비트)로 하였으며 돌연변이 확율은 0.033, 그리고 string 갯수는 250개로 하였다.

그리고 사각형 제적의 주행을 통한 규칙의 적용성을 확인하기 위해 식 (5a)~(5c)와 같은 원형 명령을 입력한다.

$$\text{Position } X=0.6+0.20 \cos (\theta t) \quad (5a)$$

$$\text{Position } Y=0.0 \quad (5b)$$

$$\text{Position } Z=0.3+0.15 \sin (\theta t) \quad (5c)$$

2. 결과

표2는 16번 진화를 거친후에 생성된 최적의 규칙을 보여준다. 표에서의 수치는 삼각형 모양을 갖는 후건부의 소속함의 꼭지점 값을 나타낸다. 로봇 매니플레이터 제적 제어를 위해서는 표의 규칙들만으로도 충분히 제어가 됨을 알 수 있었다. 조인트 1의 퍼지 논리 제어기에서 Error의 환산계수는 0.034 (rad)이고 Change of Error의 환산계수는 0.0353

표 2. 로봇 매니플레이터의 최적 규칙 베이스
 Table 2. Optimal rule-base for the robot manipulator.

		Error		
		NS	ZO	PS
Change of Error	NE		0.42	0.71
	ZO	-0.80	0.0	0.87
	PO		-0.28	0.72

Scaling factor of Error:0.059
 Scaling factor of Change of Error:0.337

(a) 조인트 1의 최적 규칙 베이스

		Error		
		NS	ZO	PS
Change of Error	NE	0.18	0.48	
	ZO	-0.87	0.0	0.26
	PO	0.06	-0.33	

Scaling factor of Error:0.034
 Scaling factor of Change of Error:0.353

(b) 조인트 2의 최적 규칙 베이스

(rad), 그리고 조인트 2의 퍼지 논리 제어기의 각 환산계수는 0.059(rad)와 0.337(rad)임을 알 수 있다.

한편, 그림11은 초기 임의로 만들어준 초기 집단중에서 적합도가 가장 높은 규칙 베이스에 의한 로봇의 주행 궤적을, 그리고 그림12는 16번 진화 후의 사각형 주행 궤적을 보여주고 있다. 이들 결과들로 볼때 초기 규칙 베이스에 의해서 발생한 오차들이 진화를 거듭함에 따라 오차가 거의 사라짐을 알 수 있다. 또한 16번 진화후의 규칙 베이스가 최적 상태임을 그림13의 적합도의 변화에서 알 수 있다. 즉, 16번 진화 이후에는 적합도의 증가가 거의 없음을 알 수 있다.

또, 16번 진화후 찾은 퍼지 논리 규칙 베이스 (표 1의 규칙 베이스)의 적응성을 실험하기 위해 식(5)의 원형 궤적을 주행하도록 한다. 그림14는 주행 원형 궤적을 보여주고 있다. 이 결과에서 사각형 주행으

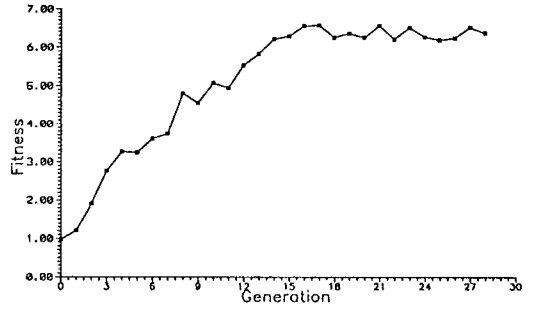


그림13. 각 진화단계에서의 적합도
Fig. 13. Fitness value at each generation.

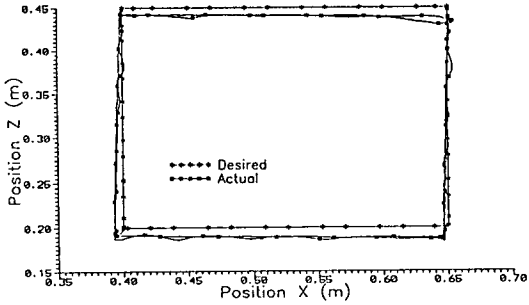


그림11. 초기 임의의 규칙 베이스에서 가장 높은 적합도를 갖는 string에 의한 주행 결과
Fig. 11. Highest fitness result of random initial control rule.

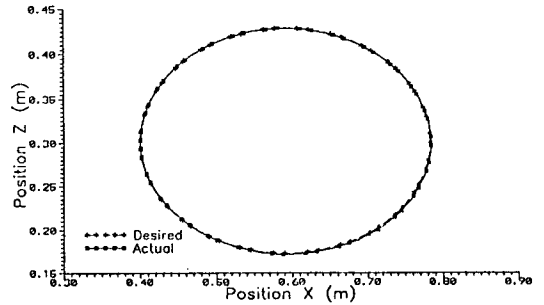


그림14. 16번 진화후 찾은 최적 규칙에 원형 궤적의 실험
Fig. 14. Test to circular path with best rule found in 16'th generation.

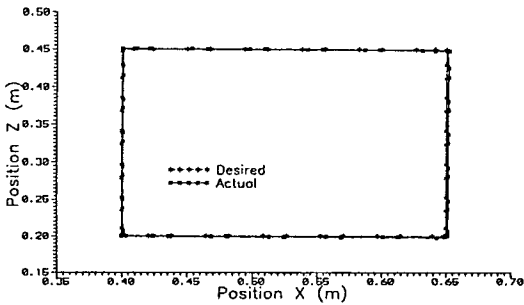


그림12. 16번 진화후의 가장 높은 적합도를 갖는 string의 주행 결과
Fig. 12. Best result after 16'th generation.

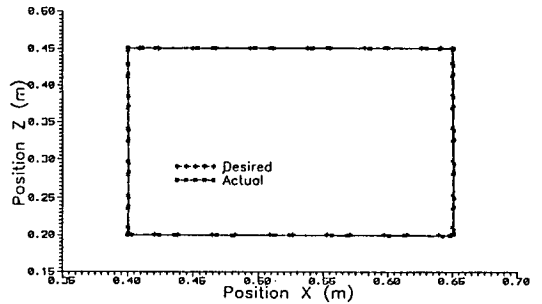


그림15. 파라미터 변경에서의 결과 (각 링크의 질량을 30% 증가)
Fig. 15. Result to parameter variation.

로부터 얻어진 규칙 베이스가 더 이상의 진화 없이 변화된 명령 코드를 훌륭히 수행하는 것을 알 수 있다. 그리고 그림15에서는 제어 대상의 파라미터 변경(로봇 매니플레이터 각 링크의 질량을 30% 증가)에 대해서도 효율적인 제적 제어가 이루어짐을 알 수 있다.

Ⅶ. 결론 및 앞으로의 연구 방향

퍼지 논리 제어기 구성에 있어 어려운 점의 하나는 최적의 규칙 베이스를 구성하는 일이다. 이러한 규칙 베이스를 구성하는데에는 전문가의 지식이나 실험적인 경험을 필요로 한다. 따라서 제어 대상이 고도의 비선형 시스템이거나 복잡한 동적 특성을 갖는 시스템에서 효과적인 규칙 베이스를 구성하는 일은 용이한 일이 아니다.

이에 본 논문에서는 최근 최적화 문제에서 강력한 탐색 알고리즘으로 각광받고 있는 유전 알고리즘을 이용, 제어 환경에 적합한 규칙 베이스를 자동 생성할 수 있는 방법을 제안하고 있다. 이 방식은 기존의 방식보다 실제 구현화(implement)하기 용이하고 최적해를 비교적 쉬운 방법으로 발견할 수 있는 잇점을 갖고 있다. 그리고 이 방식의 효용성은 로봇트 매니플레이터의 위치 제어에 응용한 결과로부터 입증되고 있다.

그러나 유전 알고리즘은 그 특성상 퍼지 규칙 베이스를 생성하는 진화 단계에 다소 시간이 소요된다는 단점이 있다. 이에 속도 개선 시키는 방향의 연구가 이루어져야 할 것이다. 아울러 본 연구에서 조정 대상으로 삼은 후건부의 퍼지 변수 및 환산계수 뿐만 아니라 전건부의 퍼지 변수를 포함하는 포괄적인 연구가 필요하며 현재 이에 대한 연구가 진행 중이다.

參 考 文 獻

[1] Shin-ichi Horikawa et al, "A fuzzy controller

- using a neural network and its capability to learn control rules," *Proceedings of the International Conference on Fuzzy Logic & Neural Networks*, pp. 103-106, 1990.
- [2] Hideyuki TAKAGI, "Fusion technology of fuzzy theory and neural networks-survey and future directions," *Proceedings of International Conference on Fuzzy Logic & Neural Networks*, pp. 13-26, 1990.
- [3] Shihuang SHAO, "Fuzzy self-organizing controller and its application for dynamic processes," *Fuzzy Sets and Systems* 26, pp. 151-164, North-Holland, 1988.
- [4] David E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley Publishing Company, 1989.
- [5] Yuval DAVIDOR, *Genetic Algorithms and Robotics, A Heuristic Strategy for Optimization*, World Scientific Publishing Co. 1991.
- [6] Joey K. Paker, Ahmad R. Khoogar, David E. Goldberg, "Inverse kinematics of redundant robots using genetic algorithms," *Proceedings 1989 IEEE International Conference on Robotics And Automation*, pp. 271-276, 1989.
- [7] Abraham Kandel, *Fuzzy Mathematical Techniques with Applications*, Addison-Wesley Publishing Company, 1986.
- [8] Witold Pedrycz, *FuzzyControl and Fuzzy Systems*, Research Studies Press LTD., 1989.
- [9] 박세희, 조현찬, 심귀보, 전홍태, "유전 알고리즘을 이용한 퍼지 제어 규칙의 생성," 한국 퍼지 시스템 학회, 1991년도 추계학술대회 학술발표논문집, pp. 108-119, 1991.
- [10] K.S. Fu, R.C. Gonzalez, C.S.G. Lee, *Robotics Control, Sensing, Vision and Intelligence*, McGraw-Hill

 著 者 紹 介



朴 世 喜 (學生會員)

1968年 3月 4日生. 1991年 중
양대학교 전자공학과 졸업. 1991
년~현재 중앙대학교 대학원 전
자공학과(석사과정). 주관심분야
는 Fuzzy Control, Genetic
Algorithm, Neural Networks등임.



金 容 虎 (學生會員)

1962年 3月 4日生. 1986年 광
운대학교 전자공학과 졸업. 1988
년 중앙대학교 대학원 전자공학
과(석사). 1992년~현재 중앙대
학교 대학원 전자공학과 (박사과
정). 주관심분야는 Robotics, Ad-
aptive Control, Fuzzy-Neural Control 등임.

沈 貴 寶 (正會員) 第29卷 B編 第1號 參照
현재 중앙대학교 제어계측
공학과 조교수

全 洪 兌 (正會員) 第29卷 B編 第1號 參照
현재 중앙대학교 전자공학과
부교수