

# 진화 알고리즘을 기반으로한 지능 제어

이말레 · 김기태

## Intelligent Control Based on Evolution Algorithms

Mal Rey Lee, Ki Tae Kim

### 요 약

본 논문에서는 진화 알고리즘을 이용하여 퍼지 규칙 베이스의 최적 규칙들을 자동으로 생성하는 방법을 제안한다. 진화 알고리즘에 의한 퍼지 논리 시스템의 최적 규칙은 전문가의 사전 경험이나 지식이 없이도 자동 설계가 가능하고 이들 규칙을 이용하여 지능 제어를 할 수 있다. 본 논문에서 사용한 접근 방법은 퍼지 규칙 소속함수의 자동 조정으로 규칙을 생성하고, 최적의 제어 규칙 탐색은 퍼지 논리 시스템의 성능 기준으로 정의한 적합도 값을 기반으로 탐색한다. 제안한 방법의 유용성을 보이기 위해 비선형 시스템에서 컴퓨터 모의실험을 행하였다.

### ABSTRACT

In this paper, we propose a generating method for the optimal rules of the fuzzy rule base using evolution algorithms. With the aid of evolution algorithms optimal rules of fuzzy logic system can be automatic designed without human expert's priori experience and knowledge. and can be intelligent control. The approach presented here generating rules by self-tuning the parameters of membership functions and searches the optimal control rules based on a fitness value which is the defined performance criterion.

Computer simulations demonstrates the usefulness of the proposed method in non-linear systems.

## I 서 론

퍼지 논리를 이용한 퍼지 논리 제어기는 제어 대상의 정확한 수학적 모델링이 불필요하고 언어적인 제어 그리고 병렬 처리 계산을 가능케 하는 등의 장점을 갖고 있어 퍼지 논리를 기반으로 한 많은 제어기가 Ysunobu 와 Myamoto (1985), Kosko (1992), Hwang 등 (1993) 등에서 성공적으로 구현 되었다. 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 전문가의 경험이나 지식을 if-then 형식으로 구성된 지식 베이스이다. 그리고 퍼지 추론기는 규칙 베이스의 규칙을 이용하여 퍼지 입력에 대한 퍼지 출력을 추론하는 기능을 가지며, 비 퍼지화기는 추론기에서 출력한 값을 명확한(crisp) 제어 값으로 변환시키는 기능을 갖는다.

그러나, 퍼지 제어기의 구성에 있어 어려운 점은 퍼지 규칙 베이스를 구성하는 일이다. 전문가 또는 운전자의 경험과 지식 그리고 시행 착오를 겪은 실험에 의해 규칙 베이스를 구성할 수 있지만 실세계의 제어 대상은 선형적인 시스템이 아니라 고도로 비선형인 경우가 대부분이다. 이런 경우 규칙 베이스를 구성 하려면 매우 어렵다. 만일 규칙 베이스를 구성 하였다 하더라도 인간 전문가는 때때로 특별한 환경에서 의사 결정을 잘 못할 수 있을 뿐만 아니라 의사 결정을 했을때 제어값 또한 믿을만한 근거가 없기 때문에 그것이 최선인지는 알 수 없다.

이러한 어려움을 보완하기 위해 제어 규칙들을 자동으로 생성 할 수 있는 방법들에 대한 연구가 최근에 활발히 진행되고 있다. 전문가의 경험이나

지식을 기반으로 한 퍼지 제어 규칙은 퍼지 학습 규칙을 구성하는 방법과 각 퍼지 규칙 소속 함수의 파라미터 값을 동적인 시스템에 적절하게 적응 시키는 것이 시스템 성능 향상에 매우 중요한 역할을 하기 때문에 사전의 전문가의 경험이나 지식이 없이도 적절한 학습을 할 수 있도록 해야 한다(Xu 와 Lu, 1987). 이러한 연구들은 신경망(neural network) 과의 융합(fusion)을 통한 규칙의 자동 생성과 자기 동조에 의한 자동 생성 방법등으로 대분할 수 있다(Hideyuki (1990), Shihuang (1988)).

신경망과의 융합 방식은 제어 규칙의 언어 변수 소속함수를 신경망 구조와 학습 능력을 이용하여 제어 대상에 맞게 조정하는 방식으로 비선형 함수를 모델링 할 수 있고 불확실한 정보를 처리할 수 있으며 일반화 기능, 강인성등을 가지므로 그 유용성은 이미 알려져 있다. 한편 신경망의 학습 알고리즘은 오차 역전파(error backpropagation)로서 오차를 정정하여 입력에 대해 원하는 반응과 실제로 얻어진 것들에 대한 차이를 줄여 나가는 방법이다. 하지만 신경망 역시 학습 시간, 입력층과 은닉층에서 이용되고 있는 가중치를 적절한 학습을 위해 어떻게 선정해야 하느냐의 문제가 발생하고, 지역적인 탐색에서 적절한 해를 찾았을때 전역적인 탐색을 하지 않고 지역적으로 수렴하는 국부해(local minimum)에 빠지는 문제등을 해결해야 한다(Shaohua 와 Shi-zhong, 1992).

한편 자기동조 방식은 실행 측정표를 만들어 현재 사용된 측정표에 따라서 제어 규칙의 결론부를 강화 또는 약화 시켜 제어 규칙을 수정하는데 이 실행 측정표는 전문가의 경험과 지식을 기반으로 하기 때문에 효율적인 방식이라 할 수 없다.

이런 문제점을 해결하기 위해 본 논문에서는 진화 알고리즘을 이용하여 비선형 시스템에서 규칙들을 자동으로 생성하는 방법을 제안하고 이들 규칙으로 지능적인 제어를 할 수 있도록 한다. 퍼지 제어기의

규칙 베이스와 제어기 내의 각 파라미터의 최적 값을 진화 알고리즘에 의해 결정한다. 이 방법은 부정확하고, 불확실한 환경에 효과적이며 생성 방법으로 이용하는 진화 알고리즘은 전역적인 병렬 탐색 알고리즘이기 때문에 신경망의 국부해(local minimum)을 해결할 수 있으며, 사전의 전문가의 경험이나 지식이 없이도 불확실한 환경에 적응할 수 있는 지능적인 제어를 할 수 있다.

본 논문의 구성은 다음과 같다. II 장에서는 기본 개념으로 다윈의 진화론에 근거를 둔 유전 알고리즘과 진화 전략에 대해 알아 본다. III 장에서는 지능 제어를 할 수 있도록 퍼지 규칙 베이스의 제어 규칙들을 자동 생성하는 방법에 대해서 설명하고, IV 장에서는 컴퓨터 모의실험을 통해 본 논문의 유용성을 보이며 V 장에서는 결론 및 앞으로 연구 방향을 제시하고 본 논문을 맺는다.

## II. 진화 알고리즘 (Evolution Algorithms)

60년대 중반 부터 지속적으로 연구가 진행 되어온 진화 알고리즘은 자연의 유전학(natural genetics)과 자연 선택(natural selection)의 원리에 근거한 최적점 탐색 방법으로 기존의 최적점 탐색 방법 에서와 같은 국부탐색(local search)이 아닌 여러해를 동시에 탐색하는 전역탐색(global search)으로 국부 수렴의 확률이 줄어든다. 이런 장점으로 지난 30년 동안 진화와 유전 이론에 기반으로 한 문제 해결 시스템에 많은 관심을 가져왔다. 이들 시스템들은 잠재해의 집단을 유지하며 개체의 적합도에 의해 선택절차와 유전 연산자를 가지고 문제를 해결한다. 진화 알고리즘의 기초는 유전 알고리즘과 진화 전략이 있는데 진화 전략은 시스템 전체의 수렴 속도를 높이기 위해 적

용한다.

### 2-1. 유전 알고리즘(Genetic Algorithms)

유전 알고리즘은 1975년 John Holland에 의해 제안 되었다(Holland (1975), Goldberg (1989)). 이 방법은 가능한 해의 탐색 공간에서 국부탐색이 아닌 전역 탐색으로 최적의 해를 발견하며, 병렬 탐색으로 미지의 함수( $y=f(x)$ )를 최적화 하는 파라미터  $x$ 를 탐색하는 기법이다. 유전 알고리즘의 또다른 장점은 첫째, 탐색 공간에서 단일 해를 사용하지 않고 해 집단을 사용한다. 둘째, 파라미터 그 자체를 사용하지 않고 파라미터 집단을 부호화 하여 사용한다. 그리고 유전 알고리즘은 미분 가능성과 같은 최적화 함수의 정보를 필요치 않고 단지 적합도 함수 값만 이용하며, 결정론적인 전이 규칙이 아니라 확률적인 전이 규칙을 이용한다(Krishnakumar 와 Goldberg , 1992).

기존의 학습 알고리즘은 보통 Newton-Gauss와 강하 방법(descend method)과 같은 기울기 벡터(gradient vector)의 계산에 의해 주어진 방향에서 작은 적응을 할 수 있도록 하였다. 이 적응 탐색 기법은 몇가지 문제점을 가지고 있는데 하나는 초기의 해에 심하게 의존하며 보통 랜덤하게 생성되어 지역적 탐색을 피하지 못한다. 이런 기존의 문제점을 해결 하기 위해 유전 알고리즘은 함수의 최적화, 영상 처리, 세일즈맨 방문 문제, 제어 응용과 같은 영역에서 적용되고 있으며 계속 연구중이다.

유전 알고리즘의 진화과정은 재생산(reproduction), 교배(crossover), 그리고 돌연변이(mutation)등의 연산자에 의해 처리되며, 일반적인 과정은 다음과 같다.

[단계 1] 주어진 문제의 가능한 해를 부호화된 스트링으로 표현하여 초기집단을 구성한다.

- [단계 2] 집단의 각각의 스트링에 대한 적합도를 평가한다.
- [단계 3] 적합도에 따라 현재 집단에서 스트링을 선택한다(재 생산).
- [단계 4] 선택된 부모 스트링간의 교배 및 돌연변이 진화과정을 통해 새로운 자식 스트링을 생성
- [단계 5] 새로 생성된 스트링에 대한 적합도를 평가하여 이전 세대와 교체한다.
- [단계 6] 최적해를 구할때까지 3단계 이하의 과정을 반복한다.

재생산은 적합도 함수 값에 확률적으로 비례하여 현재 세대에 있는 스트링이 다음 세대로 복사되는 과정이다. 이때 적합도의 값이 높으면 높을수록 다음 세대에 선택될 확률이 높게 된다. 적합도는 평가함수가 된다. 교배 연산은 랜덤 과정을 통하여 스트링을 서로 섞는 연산이며 교배 확률로 이루어지고 교배 위치는 임의로 선택 한다. 돌연변이 연산은 아주낮은 돌연변이 확률로 한 비트를 0과 1사이를 교체함으로써 실행되며, 탐색의 방향이 지역적인 극값으로 빠질 경우 이를 해결 하는데 도와주고 임의 비트 전환으로 이루어 진다.

## 2-2. 진화 전략(Evolution Strategy)

진화 전략은 최적화 문제를 해결하기 위해 자연계의 진화가 적자 생존의 원리를 통하여 최적의 개체를 얻도록 이루어 진다는 개념을 이용한 방법들이다(Tomas, 1994). 유전 알고리즘은 비교적 많이 연구되고 있으나 진화 전략의 연구는 초기 단계에 있다. 유전 알고리즘은 재생산, 교배, 돌연변이 연산을 통해서 최적의 해를 찾아 가지만 진화 전략은 “돌연변이법”을 통하여 새로운 자손 세대를 만들어 내고

이를 확률적으로 선택하여 점차 최적의 개체를 찾아가는 방법이다. 유전 알고리즘과 진화 전략의 구조는 같으나 전자는 개체들에 대해 고정된 길이의 이진 스트링으로 표현하고, 주어진 문제를 이진 스트링 위에서 적절한 모양으로 수정을 필요로 한다. 후자는 유전자 표현을 위해 풍부한 데이터 집합으로 표현할 수 있으며, 문제를 변환 시키지 않고 적절한 유전 연산자를 적용한다.

$a$ 개의 문자열로 부터 적합도가 높은  $n$ 개를 생성한후  $n$ 개에 대하여 각각  $a$ 개의 문자열을 만들어 낸다.  $a * n$  개의 자손중  $r$ 개를 선택하여 다음 세대의 자손으로 만들어 낸다. 이들 과정은 다음과 같다.

- 1) 초기화(initial) : 문자열  $a$ 개를 랜덤하게 발생시켜 초기 해 집단을 생성한다.
- 2) 선택(selection) :  $a$ 개의 문자열에 대해 적합도를 계산하고 평가한 문자열 중 적합도가 높은  $n$ 개를 생성한다.
- 3) 돌연변이(mutation) : 선택한  $n$ 개에 대하여 평균은 0이고 표준 편차는 Gaussian 확률 변수를 랜덤하게 설정하여, 즉  $N(0, V)$ 을 더하여  $a * n$ 개의 문자열을 생성 한다.

진화 전략의 일반적인 구조는 다음과 같다.

```

procedure evolution strategy
begin
  t ← -0
  initialize p(t)
  evaluate p(t)
  while (not termination-condition) do
  begin
    t ← -t + 1
    select p(t) from p(t-1)
    alter p(t)
    evaluate p(t)
  end
end

```

그림 1. 진화 전략 구조

### III. 진화 알고리즘을 이용한 규칙 자동 생성

#### 3-1. 퍼지 논리 제어기

일반적으로 퍼지 논리 제어기의 구성은 그림 2와 같다. GE는 오차의 전체집합을 정하는 환산계수(scaling factor)이고 GC는 오차 변화분의 환산 계수이다. 그리고, R은 제어기의 명령이 되며 C는 제어 대상의 명령에 대한 결과이다.

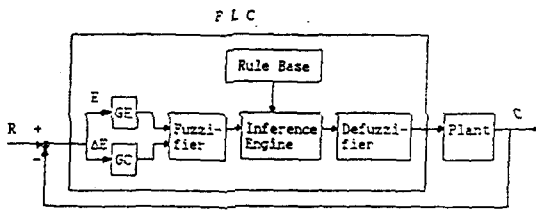


그림 2. 퍼지 논리 제어기

그림 2의 퍼지 논리 제어는 크게 네부분 즉, 퍼지 화부, 추론부, 퍼지 규칙 베이스, 그리고 비퍼지화부로 구성된다. 퍼지화부는 제어기 입력 변수 값을 측정하여 입력 변수 값의 영역을 이에 상응하는 전체집합으로 변환하여 적절한 언어적인 값 즉, 입력부 전체집합 내에서 정의된 퍼지 집합들의 이름을 사용한다. 그리고 추론부는 규칙 베이스의 규칙을 이용하여 퍼지 입력에 대한 퍼지 출력을 추론하는 기능을 가지며 여기서 사용하는 소속함수는 삼각형 함수를 사용한다. 또한 입력 퍼지 변수와 출력 퍼지 변수와의 관련 퍼지 제어 규칙을 퍼지 규칙 베이스라 한다. 그러나 제어 대상에 따라 적절한 규칙을 구성하는 일은 용이하지 않고 특히 제어 대상이 비선형일 경우 그 구성은 매우 어렵다. 이들 규칙은 추론의 합성 규칙을 사용하여 퍼지 변수를 기반으로하여 추론한다.

그리고 적절하게 계산된 제어값은 무게 중심법(centroid method)를 사용하여 결정된다. 이 방법은 연산이 간단하고 좋은 결과를 획득할 수 있다. 이때 비퍼지 화기는 명확한 값을 얻는다(Abraham, 1986).

#### 3-2. 퍼지 추론

본 논문에서 사용한 퍼지 규칙은 다입력 단일 출력의 집합 형태로 표현하며 규칙은 식 (1)과 같다(Kickart 와 Mandani, 1978).

$$\text{Rule } i: \text{ IF } E \text{ is } A_i \text{ and CE is } B_i \text{ THEN CA is } C_i \quad (1)$$

여기서  $i, (i=1, \dots, n)$ 은 퍼지 제어 규칙 수를 나타내고 E, CE 와 CA는 각각 입력 및 출력 변수들이다.  $A_i, B_i, C_i$ 는 그림 3의 소속함수들을 갖는 퍼지 변수이다. 퍼지 추론법은 가장 널리 사용되는 Mamdani의 MIN-MAX 방법을 사용하여 입력 (E, CE)에 대한 퍼지 추론치  $y_i$ 는 식 (2)와 같다.

$$\begin{aligned} \mu(y_i) &= \bigcup_{i=1}^n \{ \mu_{A_i}(E) \wedge \mu_{B_i}(CE) \wedge \mu_{C_i}(CA) \} \\ &= \bigcup_{i=1}^n \{ \alpha_i \wedge \mu_{C_i}(CA) \} \end{aligned} \quad (2)$$

여기서  $\alpha_i$ 는 전체부 조건의 소속정도 이다.

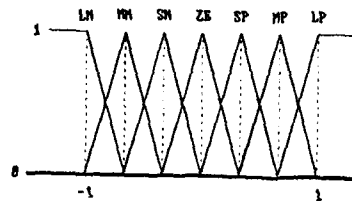


그림 3. 퍼지 규칙의 소속함수

일반적으로 소속함수의 모양은 종형과 이등변 삼각형등이 있으나 제어 시스템에서는 연산이 간단한 삼각형 모양을 많이 이용한다. 이에 본 논문에서도 각 퍼지 변수들은 삼각형 모양 소속함수를 사용한다. 소속 함수의 파라미터를 하나의 스트링으로 나타내면 소속 함수는 중앙값(center value)과 기울기에 관해 2개씩의 파라미터를 갖게 된다. 여기서 소속 함수의 퍼지 변수는 7개로 정의한다. 즉, LN(Large Negative), MN(Medium Negative), SN(Small Negative), ZE(Zero), LP(Large Positive), MP(Medium Positive), 와 SP(Small Positive)이다. 그래서 하나의 스트링은 49개의 소속 변수를 갖게 된다. 소속값의 범위는  $[-1, +1]$ 로 했다. 퍼지 논리 판단 과정에서 퍼지 규칙들의 집합은  $R = \{R_1, R_2, \dots, R_{49}\}$ 로 표시하고, 각 규칙  $A_i, B_i, C_i$ 는 퍼지 언어 변수이고 한 규칙의 예로 (SP, LN, CA)와 같이 표기 할 수 있다. 퍼지 제어 시스템의 규칙들이 표 1에 나타나 있다. 이 규칙들은 진화 알고리즘에 의해 자동적으로 결정 되게 된다. 각 규칙  $R_i = (A_i, B_i, C_i)$ 들은 주어진 입력쌍  $E(t)$ 와  $CE(t)$ 로 표현되고 추론식 (2)에 의해서 규칙  $i=1, \dots, 49$ 들을 결합함으로써 전체적인 규칙( $E(t), CE(t), CA$ )를 얻을 수 있다.

표 1. 퍼지 제어 시스템 규칙

CE(t) E(t)	(LN)	(MN)	(SN)	(ZE)	(SP)	(MP)	(LP)
(LP)	LP	LP	LP	LP	MP	SP	ZE
(MP)	LP	LP	LP	MP	SP	ZE	SN
(SP)	LP	LP	MP	SP	ZE	SN	MN
(ZE)	LP	MP	SP	ZE	SN	MN	LN
(SN)	MP	SP	ZE	SN	MN	LN	LN
(MN)	SP	ZE	SN	MN	LN	LN	LN
(LN)	ZE	SN	MN	LN	LN	LN	LN

한편 Mamdani의 추론 방법에 의한 퍼지 제어기의 비퍼지화 과정은 퍼지 논리 규칙 단계의 결과를 실제적인 출력으로 바꾸는 과정인데 제어기에서 많이 쓰는 무게 중심법 식 (3)을 사용한다.

$$CA' = \frac{\sum_{i=1}^n W_i \mu(y_i)}{\sum_{i=1}^n \mu(y_i)} \quad (3)$$

여기서  $w_i$ 는 소속함수  $C_i$ 의 중앙값(center value)이고 그림 4와 같다. 가로축의 값은 소속변수이고 세로축은 소속함수의 값이다.

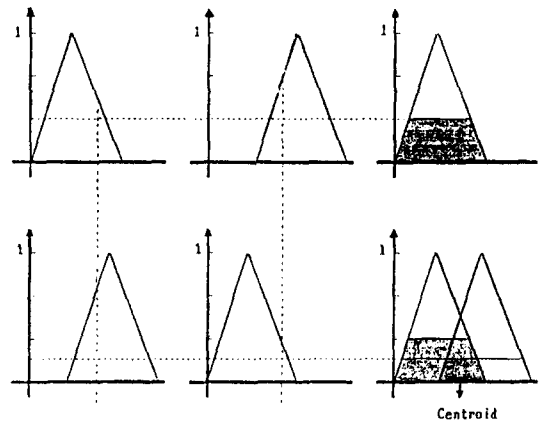


그림 4. 퍼지 추론 무게 중심법

### 3-3. 퍼지 규칙 베이스의 규칙 생성

전통적인 제어 시스템은 인간 운전자나 전문가의 경험과 지식을 기반으로 하여 제어 시스템을 구성했다. 이 방법은 전문가나 운전자의 많은 시행착오를 거쳐 환경에 가장 적절한 퍼지 규칙을 결정하기 때문에 많은 시간이 소요되며, 또한 결정된 규칙이 최적의 규칙인지 믿을 만한 근거가 확실하지 못하다. 이런 문제점을 해결 하고자 본 논문에서는 최적의

퍼지 규칙을 생성하는데 사전 전문가의 지식이 없이도 해결할 수 있는 진화 알고리즘을 이용한다. 본문에서 제안한 제어 시스템의 블록 선도는 그림 5와 같다. 그림 5의 ●은 분기점을 표시한다.

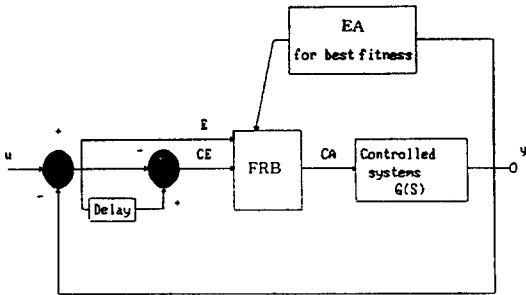


그림 5. 제안된 제어 시스템 블록 선도

퍼지 규칙 베이스는 오차( $E = u - y$ )와 출력 변수에 대한 오차변화율( $CE = E(t-1) - E(t)$ )의 입력 변수에 의존한다. 여기서는 49개의 매개변수를 사용하고 7개의 퍼지 소속값을 갖는다. 소속 함수는 다음과 같이 부호화 하여 사용한다. 즉, 0=None(unused), 1=LN(Large Negative), 2=MN(Middle Negative), 3=SN(Small Negative), 4=ZE(Zero), 5=SP(Small Positive), 6=MP(Middle Positive) 그리고 7=LP(Large Positive)과 같이 이용한다. 퍼지 규칙 베이스 각각의 매개변수는 각 변수의 영역을 포함하는 3비트의 이진수로 부호화 한다. 이렇게 소속 함수를 이진 스트링으로 나타내면 전체의 집단 스트링 갯수는 147개의 비트 스트링을 갖는다. 생물학의 자연 선택법을 생각해 볼때 부호화된 매개변수는 유전자가 된다. 여기서 매개변수의 값은 유전자의 특성이다. 스트링 집단은 모두 부호화된 매개변수에 의해 형성한다. 집단 스트링의 각각은 규칙이 된다.

본 논문에서의 적합도 함수는 시스템 성능 평가를 기준으로 하는 정상 상태 오차( $=J$ )로 정의한다. 정상 상태 오차는 식 (4)와 같이 구하고 적합도 함수는 식

(5)와 같다.

$$J = \frac{1}{2} (y^p(t) - y^p(\chi, t))^2 \quad (4)$$

여기서  $p$ 는 학습 반복수를 나타내고,  $y^p(t)$ 는 기준 출력이며  $y^p(\chi, t)$ 는 개체  $\chi = \{1, \dots, P\}$ 를 퍼지 논리 시스템의 규칙으로 하여 제어 하였을 때의 시스템 출력이다.

$$\text{Fitness}(i) = 1 / (J + 1) \quad (5)$$

단,  $i$ 는 문자열을 나타낸다.  $J$ 가 음수가 아니기 때문에  $J$ 가 최소 값일때 최대의 적합도 함수를 얻을 수 있다.

퍼지 규칙 베이스 생성은 다음과 같은 단계를 거친다.

1. 퍼지 논리 시스템의 규칙 중 결론부의  $CA = \{1, \dots, 49\}$ 를 변수로 갖는 실수 벡터( $\chi, \sigma^2$ ), 즉,  $\{1, \dots, P\}$ 를  $P$ 개의 개체로 생성한다.  $\chi = \{1, \dots, P\}$ 의 요소들의 초기값은  $-1$ 에서  $1$ 까지의 실수로 같은 확률을 가지고 선택된다. 적응 표준 편차인  $\sigma^2 = \{1, \dots, P\}$ 값을  $1.0$ 으로 초기화 한다.

2. 각 개체  $\chi = \{1, \dots, P\}$ 의 적합도 값을 적합도 함수식 (5)에 의해 계산한다.

3. 부모 개체( $\chi, \sigma^2$ )로부터 새로운 개체군 즉, 자손 세대를 생성하는데 다음과 같이 부모 세대에 외란을 주어서 생성한다. 자손세대 = 부모세대( $\chi, \sigma^2$ ) +  $N(0, V)$ ,  $V$ 는 가우시안(Gaussian) 확률 변수이다.

4. 새로운 개체군의 적합도를 계산한다.

5. 확률적인 선정으로 평가할 개체  $N$ 개의 벡터를 선택한후 이들 벡터와 자신의 적합도를 비교하여 선택하는 방법으로 최종적으로  $M$ 개의 새로운 세대를 만든다.

6. 모든 개체군을 적합도가 높은 순서대로 나열하여 그 중에서  $Q$ 개를 선택한다.

7. 만족한 해이면 끝내고, 그렇지 않으면 3으로 가서 반복한다.

### IV. 컴퓨터 시뮬레이션

제안한 방법의 유용성을 보이기 위해 컴퓨터 시뮬레이션을 수행하는데 사용된 진화 알고리즘의 파라미터는 다음과 같다.

1. 전체 집단 크기(Total population size)=20
2. 시스템 변수 수(System variable number)=49
3. 각 변수의 길이(Length of each variable)=3
4. 스트링(염색체(Chromosome length))의 길이=147
5. 최대 세대수(Maximum% of generation(maxgen))=500
6. 돌연 변이 확률(Mutation probability(pm))=0.001

많은 산업 응용 시스템은 비선형 시스템으로 구성된다. 본 논문에서 사용한 비선형 시스템은 포화

비선형 함수F(X)로 구성한다. 포화 비선형 시스템은 그림 6와 같다. 시스템에 이용한 전달 함수는 Chen, 1992에서 이용한 함수를 사용하며, 식 (6)와 같다.

$$G(s) = -s + 1 / (5*s + 1) * (s + 1) \tag{6}$$

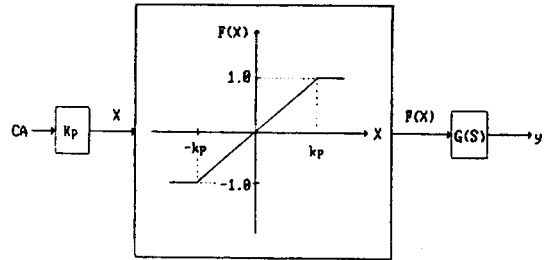
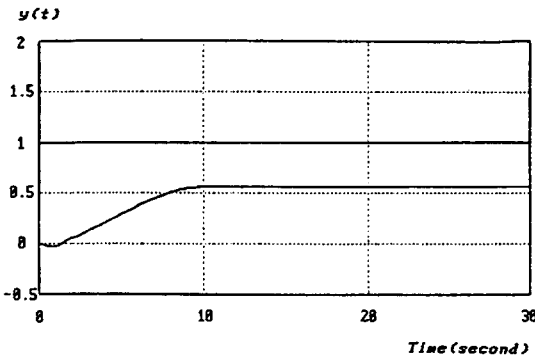
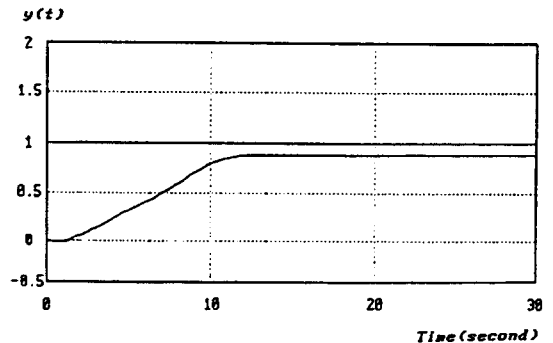


그림 6. 포화 비선형 제어 시스템 가진 블록 선도

그림 6의 제어 시스템 입력 변수는 CA이고 t스텝 시간에서의 출력 변수는 y(t)이다. 포화 비선형은 x와 F(x)에 의해서 표현된 비선형 함수로 나타낸다. 여기서  $k_p=1.0$  과  $5.0$ 를 이용한 제어기 반응이 각각 그림 7, 8과 같다.  $k_p=1.0$  을 사용 했을시 356세대를 거쳤을때 가장 적은 오차 반응으로 좋은 반응을 나타냈다.  $k_p=5.0$ 을 사용 했을 때에는 0, 5, 72세대를 생성 했을시 많은 진동이 있었으나 176세대 에서는 거의

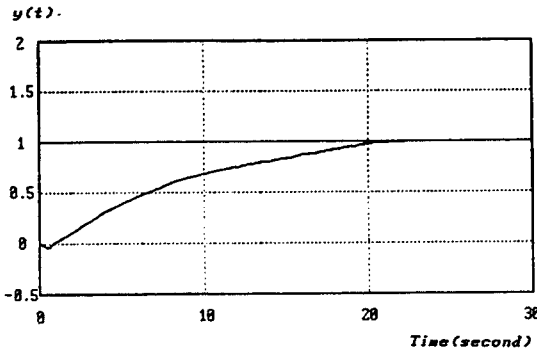


Generation 0

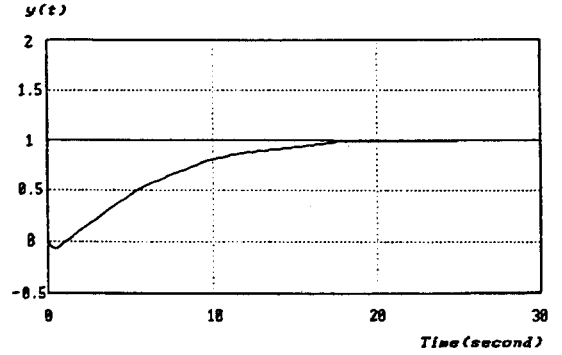


Generation 38



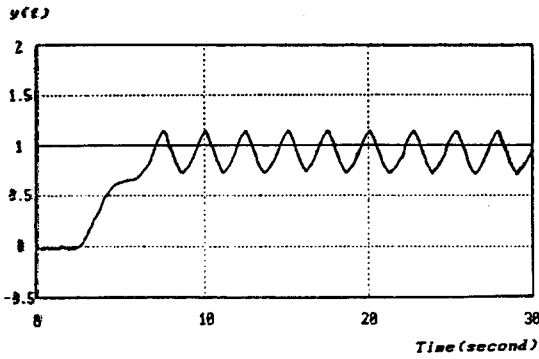


Generation 48

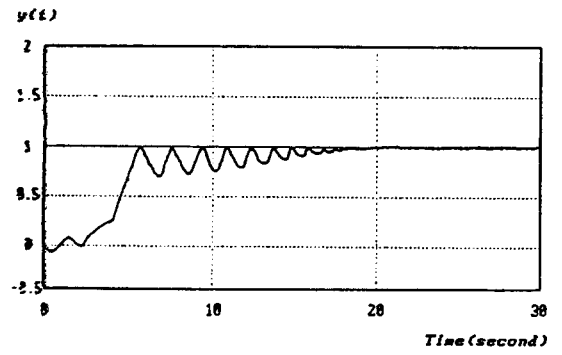


Generation 341

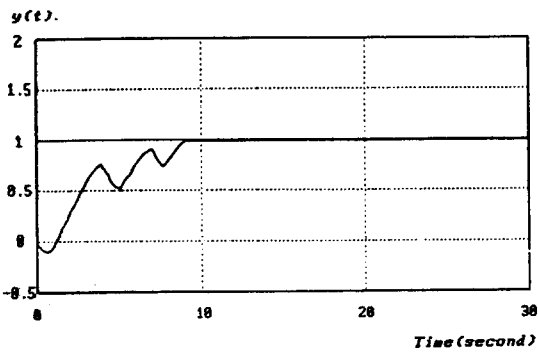
그림 7.  $K_p=1.0$ 일 때의 결과  $y(t)$



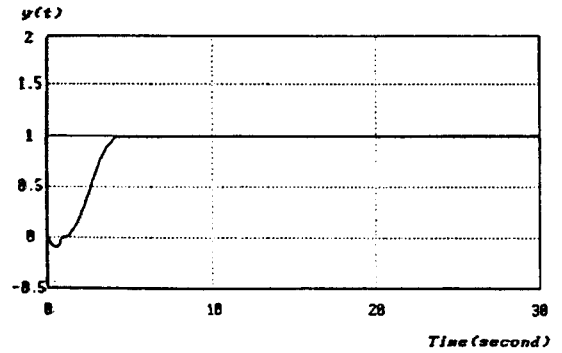
Generation 0



Generation 5



Generation 72



Generation 175

그림 8.  $K_p=5.0$ 일 때의 결과

진동이 없는 가장 좋은 반응을 나타 냈다. 그림에서 볼수 있듯이 제안한 방법은 비선형 시스템에서 좋은 반응을 나타낼 수 있었으며  $k_p$  의 값이 클수록 진동이 나타나며 정상 상태 오차가 커짐을 알 수 있다. 여기서 1은 기준 출력 값이다.

진화 알고리즘을 이용하여 생성된 각 규칙,  $R_i=(A_i, B_i, C_i)$ 들은 주어진 입력쌍 E 와 CE로 다음 표 2과 같이 생성된다. 이들 규칙은 다음과 같은 의미 이다.

- 1) if E is LP and CE is ZE then CA is LP,
- 2) if E is MP and CE is ZE then CA is MP,

.....

표 2. 진화 알고리즘에 의한 퍼지 규칙

CE(t) E(t)	(LN)	(MN)	(SN)	(ZE)	(SP)	(MP)	(LP)
(LP)		ZE		LP	SP	LP	SP
(MP)		MP		MP	LP	MP	MN
(SP)	MN			SP	SP	ZE	MN
(ZE)				ZE	SN	MN	LN
(SN)		MN		SN	LN	LN	
(MN)							
(LN)							

## V.결 론

본 논문 에서는 진화 알고리즘을 이용하여 퍼지 제어 시스템 에서의 규칙 생성 방법을 제안 하였고 생성된 규칙으로 제어 시스템에서 지능 제어를 할 수 있었다. 제안된 생성 방법은 원하는 출력응답을

갖도록 적합도를 정상 상태 오차로 했다. 진화 알고리즘은 전역탐색 알고리즘이기 때문에 기존의 최적 규칙 생성 방법에서 발생하는 국부해(local minimum)를 해결 할 수 있었다. 퍼지 제어 시스템의 사전 지식과 경험을 전문가로 부터 얻지 않고도 퍼지 규칙을 자동 조정 할 수 있었으며, 그 유용성은 포화 비선형 함수를 통해서 증명했다. 제안된 방법은 비선형 함수 모델에서 최적 규칙을 생성하는데 사용될 수 있다. 추후 과제로는 다양한 플랜트의 적용이 필요한데 특히, 고도의 비선형 시변 시스템으로써 정밀한 제어가 매우 어려운 제어 대상으로 인식되어온 로봇트 매니플레이터의 적용을 해볼려고 한다. 로봇트 매니플레이터는 산업용, 교육용, 특수 목적등으로 개발되고 있고 널리 사용되고 있지만, 시스템이 고도의 비선형 시변 특성을 갖고 있기 때문에 정밀한 제어가 매우 어려운 실정이다. 따라서 수많은 제어 이론이 적용되고 여러 형태의 제어 시스템이 설계되어 왔지만 기존의 방법들은 정확한 수학적 모델링을 필요로하는 제약 때문에 정밀 제어가 어렵거나 또는 불가능 했다. 그리고 제어 환경에 변화가 발생하였을 경우에는 제어 시스템의 안정도가 보장되지 않은 경우 마저 발생한다. 그러나 퍼지 논리 제어기 설계에서 가장 어려운점은 규칙 베이스를 구성하는 일이다. 전문가에 의한 규칙 베이스 구성은 상당히 주관적이고 경험에 의존하기 때문에 최적의 규칙 베이스를 결정하기란 쉽지 않다. 따라서 규칙베이스의 자동 생성을 위해서 학습능력을 갖는 진화 알고리즘을 적용하고자 한다. 또한 돌연변이 확률을 다양하게 적용하여 시스템에 미치는 영향이 어떠한지 조사하고자 한다.

## 참고 문헌

[1] Abraham Kandel, Fuzzy Mathematical Techni-

- ques with Applications, Addison-Wesley Publishing Company, (1986).
- [ 2 ] B. Kosko, Neural Networks and Fuzzy Systems : A Dynamical Systems Approach to Machine Intelligence, Prentice-Hall : Englewood Cliffs, NJ, (1992).
- [ 3 ] C. T. Chen, Analog and Digital Control System Design : Transfer-Function, State-Space, and Algebraic Methods, Raleigh, NC, (1992), 130-137.
- [ 4 ] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning., Addison-Wesley, (1989).
- [ 5 ] Hideyuki Takagi, "Fusion Technology of Fuzzy Theory and Neural Networks-Survey and Future Directions-", Proceedings of the International Conference on Fuzzy Logic & Neural Networks, (1990), 103-106.
- [ 6 ] J. H. Holland, Adaptation in Natural and Artificial Systems. Ann Arbor, MI : Univ. Mich. Press, (1975).
- [ 7 ] K. Krishnakumar and D. E. Goldberg, "Control System Optimization using Genetic Algorithms", Journal of Guidance, Control and Dynamics, Vol. 15, No. 3, (1992), 735-740.
- [ 8 ] Kickart, W. J. M., Mandani, E. H., "Analysis of fuzzy logic controller", Fuzzy sets and system, Vol. 1, No. 1, (1978), 29-44.
- [ 9 ] Shaohua Tan, Shi-zhong He, "Hybrid Control of Nonlinear Dynamical Systems using Neuralnets and Conventional Control Schemes Problem", IJCNN, Vol. 2. (1992), 805-810.
- [10] Shihuang SHAO, "Fuzzy Self-Organizing Controller and Its Application for Dynamic Processes", Fuzzy Sets and Systems 26, (1988), 13-26.
- [11] S. Ysunobu and S. Myamoto, "Automatic train operation by predictive fuzzy control", in Industrial Applications of Fuzzy Control, M. Sugeno, ed., Amsterdam, North Holland, (1985).
- [12] Tomas Back, "Selective Pressure in Evolutionary Algorithm : A Characterization of Selection Mechanisms", IEEE, Transaction. on Neural Networks, Vol. 5, No. 1, (1994).
- [13] W. R. Hwang, C. W. Tao, W. E. Thompson, and R. Paz, "A Comparison of a Fuzzy Controller with an Optimal PD Controller for a Nonlinear Control System", Proceedings of the International Conference on Fuzzy Logic & Neural Networks, (1993), 109-113.
- [14] Xu, C. W. and Lu, Y. Z., "Fuzzy model identification and self-learning for dynamic systems", IEEE Transaction. on Systems, Man and Cybernetics, Vol. SMC-17, No. 4, (1987), 683-689.