

## 범용 퍼지 제어기 자동생성 시스템 개발 및 구현

### Development of Fuzzy Controller Automatic Generation System

이 상 형\*, 김 은 태\*\*, 권 철\*, 박 민 용\*  
연세대 전자공학과\*  
국립 한경대 제어계측공학과\*\*

#### Abstract

Since the inception of fuzzy control, lots of methods to design fuzzy controller have been reported. However, it is admitted that these methods are tailored to special problems and cannot be used in general control situation.

Therefore this paper proposes automatic generation algorithm of fuzzy control system and develops an automatic fuzzy controller generator. For that purpose, the genetic algorithm is used and it searches for the optimal parameters to design the fuzzy controller

#### 1. 서론

퍼지 제어 이론은 퍼지 집합 이론을 제안한 미국의 Zadeh 교수가 1973년 이 이론을 시스템 자동 제어에 응용할 수 있다는 의견을 발표한 후, 1974년 영국의 Mamdani 교수가 스팀 엔진 제어에 처음으로 응용하여 실용 가능성을 보여주면서 연구에 박차를 가하게 되었다. 그 후 퍼지 이론은 여러 가지 제어분야에 성공적으로 이용되고 있다. 이러한 퍼지 제어 응용의 성공과 함께 퍼지 제어기를 설계하기 위한 여러가지 방법론이 대두되게 되었는데 처음 Mamdani가 사용한 방법같은

직관적 방식에서부터, Sliding 이론을 이용한 퍼지 제어기 설계 방식, 비선형 적응 제어의 기법을 이용한 방식, 선형 제어 이론을 변형한 Sugeno형 퍼지 제어기 설계 방식, 무작위 생성을 한 후 신경회로망을 이용하여 미세 조정하는 방식, simulated annealing 방식, 유전인자 알고리즘을 이용한 설계 방식등이 대두되었다. 그러나 이러한 방식들은 모두 각 주어진 상황에만 적합한 방식으로 제어 대상이 차원이 높은 시스템이거나, 제어 대상이 불확실하거나 이 밖의 모든 경우에 적용되는 범용 퍼지 제어기 설계 방식은 사실상 없는 게 현실이다.

이런 이유로 퍼지 제어의 등장과 함께 많은 퍼지 제어 시뮬레이터가 상용으로 개발되었지만, 모두 설계된 퍼지 제어기를 시뮬레이션하는 기능만 가지고 있을 뿐, 실제로 퍼지 제어기를 자동으로 생성하는 능력을 포함한 퍼지 제어 시뮬레이터 시스템은 상용화되어 발표된 바가 없다.

따라서 본 논문에서는 유전자 알고리즘을 이용한 퍼지 제어 시스템의 자동 생성 알고리즘을 제안하고 이를 이용하여 범용으로 사용될 수 있는 퍼지 제어기 자동 생성 시스템을 개발하고 이를 구현한다.

#### 2. 제안된 퍼지 제어기 자동생성 시스템

퍼지 제어기의 자동 설계에 유전자 알고리즘을 이용하기 위하여 퍼지 제어기를 유전정보로 코딩하는 전략이 필요하게 된다. 이때 유전 정보 코드에 들어가게

본 논문은 현대 자동차의 지원에 의해 연구되었음

되는 퍼지 제어기의 파라미터는 입출력 소속함수  $A_i^l$ ,  $B_j^r$ 의 파라미터와 퍼지 규칙을 나타내는  $r_j^l$ 값이다. 입출력 소속함수  $A_i^l$ ,  $B_j^r$ 의 파라미터는 그림 1과 같이 정의되며 이에 따른 퍼지 제어기의 유전자 코드는 그림 2와 같은 방식으로 정의된다.

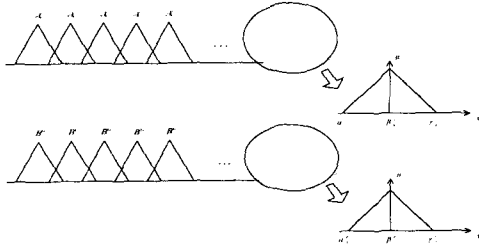


그림 1 입출력 소속함수의 파라미터

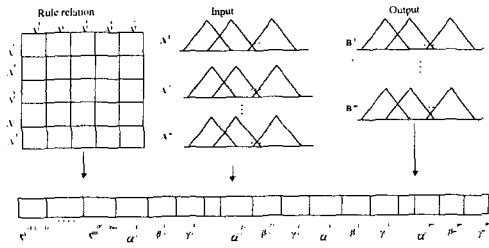


그림 2 퍼지 제어기의 유전자 코딩

이렇게 실수로 코딩된 유전자는 다음과 같은 유전자 연산자를 통하여 최적의 파라미터들을 찾아낸다.

**단순 교차 (simple crossover)**

랜덤하게 생성된 절단선(cut-point)으로써 양부모 염색체의 문자열을 분할한 다음 한 부모의 왼쪽 유전자 문자열과 다른 부모의 오른쪽 문자열을 결합하여 자손을 생성한다.

**산술 교차 (arithmetic crossover)**

0에서 1사이 무작위수  $r$ 을 발생시키고 이를 이용하여 두 개체  $\bar{X}$ 와  $\bar{Y}$ 를 교차하여  $\bar{X}'$ 과  $\bar{Y}'$ 를 생성한다.

**인위적 교차 (heuristic crossover)**

주어진 두개의 개체에 대하여 적합도 정보를 이용하여 선형 외삽한 것이다. 주어진 두개의 개체  $\bar{X}$ 와  $\bar{Y}$ 에 대하여  $\bar{X}$ 가  $\bar{Y}$ 보다 뛰어난 경우, 인위적 교차는 0에서 1사이 무작위수  $r$ 을 발생시키고 다음의 새로운 개체를 생성한다.

**균등 돌연 변이 (uniform mutation)**

돌연 변이를 일으킬 위치  $j$ 를 무작위로 결정, 이에 대하여  $\bar{X}$ 를 새로운  $\bar{X}'$ 로 돌연변이를 일으킨다.

**경계 돌연 변이 (boundary mutation)**

돌연 변이를 일으킬 위치  $j$ 를 무작위로 결정하고 0에서 1사이 무작위수  $r$ 을 발생시키고 이에 대하여 해당 유전자를 변화 가능한 최대값이나 최소값으로 돌연 변이 시킨다.

**비균등 돌연 변이 (non-uniform mutation)**

돌연 변이를 일으킬 위치  $j$ 를 무작위로 결정하고 이에 대하여  $\bar{X}$ 를 다음의 새로운  $\bar{X}'$ 로 돌연변이를 일으킨다.

이렇게 유전자 알고리즘을 이용하여 퍼지 제어기를 생성하는데는 첫째 유전자 알고리즘만 사용하는 경우 제어기 생성에 오랜 시간이 소요되고 둘째, 퍼지 제어기의 모든 파라미터가 유전자의 유전 정보로 코딩되는 경우 유전 정보가 너무 많아지며 유전자 진화 과정에서 원하지 않는 퍼지 제어기가 발생할 수 있다. 마지막으로 다입력 다출력의 경우, 퍼지 시스템의 근본적인 문제인 차원의 저주 (the curse of dimension)[11]를 피할 수 없다. 이같은 문제점을 해결하기 위해서 다음과 같은 전략이 채택되었다.

**3.기존 문제의 해결 방안**

(1) 기존의 제어기를 근사화한 퍼지 제어기

기존의 제더기가 존재하는 경우, 이 제어기를 바탕으로 이와 상당히 유사한 성능을 내는 퍼지 제어기를 생성하고 이를 초기치로 하여 유전자 알고리즘이 퍼지 제어기의 파라미터를 탐색하는 방식을 취하여 복잡하고 큰 시스템에서도 유전자 알고리즘에만 의존하는 방식에 비해 빠른 탐색 및 최적화가 가능하다. 기존제어기를 근사하는 퍼지 제어기를 구하는 알고리즘은 다음과 같다.

1단계 : (입력 퍼지 소속 함수 구성)

주어진 시스템이  $x_1$  에서  $x_n$ 까지의  $n$ 개의 입력  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$  을 사용하는 시스템인 경우,  $x_i$  에 대하여  $p_i$ 개의 퍼지 소속 함수  $A_i^{l_i}$  ( $l_i=1, \dots, p_i$ ) 를 정하고,  $A_i^{l_i}$ 는 주어진 구간  $[\underline{x}_i, \bar{x}_i]$ 를 포함 하도록 등분배(equal distribute)한다.

2 단계 : (출력 퍼지 소속 함수 구성)

1단계와 유사하게 수행한다.

3 단계 : ( 퍼지 규칙의 형성 )

다음의 식 로 표현되는  $\prod_{i=1}^n p_i$ 개의 규칙으로 만든 퍼 지 시스템

$$\hat{f}(\mathbf{x}) = (\hat{f}_1(\mathbf{x}), \hat{f}_2(\mathbf{x}), \dots, \hat{f}_m(\mathbf{x}))^T \quad (1)$$

이 주어진 기존의 제어기  $f(\mathbf{x})$ 를 다음과 같은 방식으로 근사하도록 퍼지 규칙을 형성한다:

$l = (l_1, l_2, \dots, l_n)$  번째 규칙에 대하여 다음과 같이 해당하는 삼각형 소속 함수의 중간 값으로 만든 입력 벡터를 형성한다.

$$\mathbf{x}^l = (x_1^{l_1}, x_2^{l_2}, \dots, x_n^{l_n})^T \quad (2)$$

$$x_i^{l_i} \equiv \arg_{x \in [\underline{x}_i, \bar{x}_i]} \{A_i^{l_i}(x) = 1\}, \quad (i = 1, \dots, n)$$

계산된  $\mathbf{x}^l$ 에 대하여 기존의  $f(\mathbf{x})$ 의 출력 값을 계산 한다.

$$\hat{\mathbf{y}}^l = (\hat{y}_1^l, \hat{y}_2^l, \dots, \hat{y}_m^l)^T = f(\mathbf{x}^l) \quad (3)$$

이제  $j=1, \dots, m$ 에 대하여 다음의식 (4-6)과 같이 기존의 제어기  $f(\mathbf{x})$ 를 가장 근사할 수 있도록

$l = (l_1, l_2, \dots, l_n)$  번째 퍼지 규칙을 결정한다.  $l$  번째 규칙의  $j$ 번째 출력에 대하여 퍼지 규칙  $B_j^l$ 을 다음과 같이 정한다.

$$B_j^l = B_j^{l_j}$$

$$r_j^l = \arg_{r_j=1, \dots, q_j} \min \|y_j^{r_j} - \hat{y}_j^l\|$$

$$y_j^{r_j} \equiv \arg_{y_j \in [\underline{y}_j, \bar{y}_j]} \{B_j^{r_j}(y_j) = 1\}, \quad (4)$$

$$(j=1, \dots, m, \quad r_j=1, \dots, q_j)$$

(2) 퍼지제어기의 코딩

코딩된 퍼지 제어기를 유전확습하는 과정에서 부적절 한 퍼지제어기가 발생할수 있는데 이를 막기 위하여 퍼지 제어기의 파라미터가 완전성(completeness)와 비 모순성 (consistency)를 갖도록 파라미터를 제한한다. 이를 위해 퍼지 제어기의 파라미터가 변화 가능한 영 역을 다음과 같이 정하여 진화과정의 경계 값으로 사 용하므로써 퍼지 제어기의 파라미터 값들이 부적절한 값으로 생성되지 않게 한다.

$$\Delta x_i = \frac{1}{p_i - 1} (\bar{x}_i - \underline{x}_i)$$

$$\underline{x}_i + (l_i - 2)\Delta x_i \leq \alpha_{\underline{x}}^i$$

$$\leq \underline{x}_i + (l_i - 2)\Delta x_i + \frac{1}{2} \Delta x_i, \quad (i \neq 1)$$

$$\underline{x}_i + (l_i - 1)\Delta x_i - \frac{1}{2} \Delta x_i \leq \beta_{\underline{x}}^i \quad (i \neq 1, p_i)$$

$$\leq \underline{x}_i + (l_i - 1)\Delta x_i + \frac{1}{2} \Delta x_i;$$

$$\underline{x}_i + l_i \Delta x_i - \frac{1}{2} \Delta x_i \leq \gamma_{\underline{x}}^i \leq \underline{x}_i + l_i \Delta x_i \quad (i \neq p_i)$$

$$(5)$$

출력 변수에 대하여도 비슷한 방식으로 경계 값을 설정한다.

(3) 입출력변수의 블록화

변수의 증가에 따라 문제의 복잡성이 지수적으로 증가하는 현상 이른바 차원의 저주 문제를 해결하기 위해서 따라서 본 논문에서는 입력 변수들의 블록화 를 통하여 다 변수 시스템의 퍼지 제어기를 설계하도 록 한다. 본 방식은 블록화를 통하여 사람들이 이해할 수 있게 하는 장점을 유지하면서, 퍼지 규칙의 수를 줄일 수 있는 특징이 있다. 예컨대 설계하고자 하는 제어기가 입력으로  $x_1, \dots, x_n$ 일 때 퍼지 제어기는 다음의 식으로 표현되게 된다.

$$\mathbf{u} = f(x_1, x_2, \dots, x_n) \quad (6)$$

$n$ 이 큰 경우, 식 (6)에서처럼 입력을 한번에 모두

사용하지 않고 식 (7)과 같이 출력  $u$ 에 영향을 주는 블록으로 나누어 구현한다.

$$u = f_1(x_1, \dots, x_i) + f_2(x_{i+1}, \dots, x_m) + \dots + f_b(x_{i+1}, \dots, x_n) \quad (7)$$

이 같은 구조는 식 (6)에 비해 제어 성능이 떨어지는 단점이 있으나 퍼지 규칙의 수를 줄이고 의미가 이해 가능하게 하는 장점이 있다

#### 4. 퍼지 제어기 자동 생성 시스템의 구현

본 논문에서 개발한 시스템은 퍼지 제어기를 세 가지 형태로 표현하며 그 종류는 다음과 같다.

- (1)FDL(fuzzy description language):입출력 형식 지정
- (2)멤버십 행렬: 개발 퍼지 제어기의 성능 확인에 이용
- (3) 유전인자 코드: 유전자 알고리즘에 의한 퍼지 제어기 최적화에 이용

본 시스템에서 이들 사이 변환 및 역할은 표1의 함수에 의해 이루어지며 퍼지 제어기의 형태 및 용도, 주요 함수들의 역할을 정리하면 그림 3과 같다. 그리고 최종적으로 실제 구현된 퍼지 제어기 자동 생성기는 그림 4와 같다. 안정된 GUI를 구현하기 위해서 MATLAB 5.2가 tool로 사용되었다.

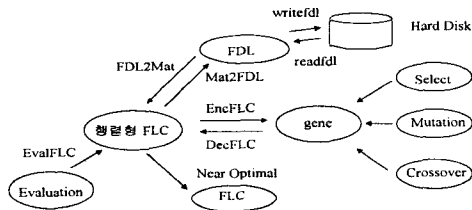


그림 3 퍼지제어기의 형태 및 용도, 주요함수의 역할

#### 5. 결론

본 논문에서는 퍼지 제어기의 파라미터들 즉 소속 함수의 값과 규칙들을 유전정보로 코딩한 후 유전자 알고리즘을 이용하여 최적의 파라미터들을 탐색하고 이를 통하여 퍼지제어기를 자동 생성하는 알고리즘을 제안하고 실제로 퍼지제어기 자동생성기 및 시뮬레이

표 1 퍼지제어기 자동생성 시스템의 주요함수와 그 역할

함수	역할
EvalFLC	제어기를 성능 평가하는 함수
FDL2Mat	fdl을 행렬형 FCL로 전환한다.
Mat2FDL	행렬형 FCL를 fdl로 전환한다.
EncFLC	행렬형 FLC를 유전인자코드로 코딩한다
DecFLC	유전인자 코드를 행렬형 FLC로 디코딩
readfdl	하드디스크로 부터 fdl을 읽어온다.
writefdl	fdl 을 하드 디스크에 저장한다.

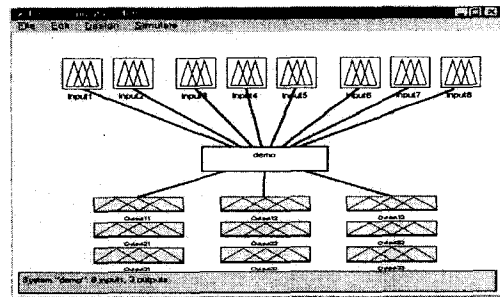


그림 4 실제로 구현된 퍼지제어기 자동생성 시스템

터를 구현해 보았다. 이렇게 자동생성기로 생성된 퍼지 제어기는 다양한 플랜트에 적용될 수 있을 것이다.

#### 6. 참고 문헌

- [1] L. A. Zadeh, "Fuzzy algorithm," *Inform. Contr.*, vol. 12, pp. 94-102, 1968.
- [2] Mamdani, E. H, "Applications of fuzzy algorithms for simple dynamic plant," *Proc. IEE*, 121, 12, pp 1585-1588
- [3] A. Homaifar and E. McCormick, " Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms," *IEEE Trans. on Fuzzy Syst.* vol.3 no. 2 1995.