

# 진화프로그래밍을 이용한 퍼지 신경망 지능 제어기 설계에 관한 연구

## A study on design of Fuzzy neural network Intelligence controller using Evolution Programming

°이 상 부\*, 임 영 도\*\*

\*제주도 제주시 노형동 1534번지 한라전문대학 전산정보처리과

\*\*부산시 사하구 하단동 840번지 동아대학교 공과대학 전자공학과

°Sang-Boo Lee\*, Young-Do Lim\*\*

\*Dept. of Computer & Information Processing, Halla College  
1534 Nohyong-dong Cheju City, Korea.

\*\*Dept. of Electronics Engineering, Dong-A University  
840 Hadan-dong Saha-gu Pusan, Korea.

### ABSTRACT

At the on-line control method FLC(Fuzzy Logic Controller) is stronger to the disturbance than a classical controller and its overshoot of the initialized value is excellent. The fuzzy controller can do a proper control, though it doesn't know the mathematical model of the system or the parameter value. But to make the control rule of the fuzzy controller through an expert's experience has a limit. Because the once determined control rule is fixed, it can't adjust to the environment changes of the control system, the controller output value has a minute error and it can't convergence correctly to the desired value[1][2]. There are many ways to eliminate the minute error[3][4][5], but in this paper suggests EP-FNNIC(Fuzzy Neural Network Intelligence Controller) intelligence controller which combines FLC with NN(Neural Network) and EP(Evolution Programming). The output characteristics of EP-FNNIC controller will be compared and analyzed with FLC. It will be showed that this EP-FNNIC controller converge correctly to the desirable value without any error. The convergence speed, overshoot, rising time, error of steady state of controller of these two kinds also will be compared.

### I. 서론

인간의 직관 및 전문가의 경험적 지식과 종래의 제어기법을 결합한 FLC는 수학적 모델이나 파라미터 값을 알지 못하더라도 제어가 가능하다. 그러나 FLC를 할 경우 PID 등의 고전적 제어기

에 비해 비선형 시스템에서는 유연한 제어성능을 발휘하고 있으나, 최적규칙의 선정 및 제어변수의 양자화 Level 선정 등에 있어서 문제점이 지적되고 있다. 특히, 속응성과 정확성이 요구되는 서보 시스템을 PID 제어하는 경우 출력응답은 목표치에 정확히 수렴하지만 부하변동에 의한 과도특성과 수렴속도의 성능도 문제가 된다[6]. 이에 비해 FLC는 부하변동에 대해 과도특성은 양호하지만 목표치에서 약간의 오차가 존재하는 결점이 있다. 학습의 능력을 가지고 있는 지능 시스템인 신경망은 오차 역전파(Error Backpropagation) 학습 규칙에 의해 환경변화에 적응성에는 우수하지만 최급강하법(Gradient Descent Rule)이라는 수학적인 방법에 근거함으로써 지역적 최소점(Local Minimum)에 수렴하여 최적화된 해를 구하지 못하는 경우도 생긴다. 진화프로그래밍은 전역적(Global)인 최적해를 구하는데 우수하지만 해집합이 클 경우 최적화 하는데 많은 시간이 걸리고 오프-라인(Off-Line)의 학습 구조이므로 제어 시스템에 적용할 경우 외부환경의 변화에 대하여 적응 제어를 할 수 없는 문제점을 가진다.

위와 같은 각각의 알고리즘의 문제점을 상호 보완하기 위하여 본 논문에서는 퍼지 제어기에 신경망과 진화프로그래밍을 결합한 EP-FNNIC(진화프로그래밍 퍼지 신경망 지능 제어기 : Evolution Programming - Fuzzy Neural Network Intelligence Controller)를 제안한다. 퍼지 제어기의 전진부를 신경망 구조로하고 비퍼지화부를 신경망으로 학습시켜서 오차역전파에 의해 전진부의 소속함수를 시스템에 적용토록 수정시킨다. EP-FNNIC는 먼저 진화프로그래밍을 이용하여 최적의 소속함수(Membership Function)값을 구해서 불확실한 시스템에 대한 근사적 소속함수를 구한다. 이 근사적 소속함수 값을 초기치로하여 지능 제어 구조를 가진 FNNIC 가 제어대상에 적용하는 퍼지 제어규칙의 전진부 퍼지 소속함수 값을 자동으로 적절히 수정하여 최적화된 지능 제어를 실현한다.

## II. FLC, NN, EP

본 논문에서 사용한 FLC, NN, EP의 이론적 배경은 다음과 같다.

### 2.1 퍼지 제어 규칙과 추론

FLC의 제어 규칙은 전문가의 지식이나 경험에 의해서 얻어진다[7]. 규칙은 "IF ~ THEN" 형식으로 구성되어 있기 때문에 전문가의 경험과 지식을 자연언어로 사용할 수 있게 한다. 제어규칙의 예는 다음과 같다.

$R_1$  : IF  $e_1$  is PB and  $ce_1$  is ZO THEN  $co_1$  is PB.

$R_2$  : IF  $e_2$  is PB and  $ce_2$  is NS THEN  $co_2$  is PM.

$R_n$  : IF  $e_n$  is ZO and  $ce_n$  is ZO THEN  $co_n$  is ZO.

여기서  $e_1, e_2, e_n, ce_1, ce_2, ce_n$  : 제어기 입력, PB, NS, ZO : 언어적인 값,  $co_1, co_2, co_n$  : 퍼지 제어기 출력

추론이란 어떤 주어진 사실이나 관계로부터 새로운 사실이나 관계를 유추하는 과정이다[4]. 그러므로 제어기 입력과 제어규칙에서 추론의 합성 규칙을 이용하여 정확한 제어기 출력값을 구할 수 있다. 추론의 합성 규칙에는 MAX-MIN, MAX-PRODUCT 등 여러 가지가 있지만 본 논문에서는 Mamdani가 제안한 MAX-MIN 합성 규칙을 사용한다. MAX-MIN 합성 규칙은 식(1), 식(2)와 같다.

$$\text{함의} : \mu_R(e, ce, co) = \text{MIN} (\mu_E(e), \mu_{CE}(ce), \mu_{CO}(co)) \quad (1)$$

$$\text{추론} : \mu_{CO'}(co) = \text{MAX-MIN} (\mu_E(e'), \mu_{CE}(ce'), \mu_{CO}(co)) \quad (2)$$

여기서  $e', ce'$  : 제어기 입력의 실수값  $co'$  : 추론된 제어기 출력값  $\mu$  : 소속함수 값

본 논문에서는  $e$ 와  $ce$ 에 대해서 어떤 측정값  $e'$ 와  $ce'$ 가 입력되면 그 값에 대한 퍼지 변수의 모든 소속함수를 취하지 않고 식(3)과 같이 그때 그 부분에 해당되는 값만을 취하는 결정론적인 입

입력형태  $e^0$ 와  $ce^0$ 를 사용한다[4].

$$\mu_E(e^0) = \begin{cases} 1, & e^0 = e' \\ 0, & \text{그 외} \end{cases} \quad \mu_{CE}(ce^0) = \begin{cases} 1, & ce^0 = ce' \\ 0, & \text{그 외} \end{cases} \quad (3)$$

그러면 식(2)는 식(3)에 의해서 식(4)와 같이 된다.

$$\mu_{CO}(co) = \text{MAX-MIN} (\mu_E(e^0), \mu_{CE}(ce^0), \mu_{CO}(co)) \quad (4)$$

규칙이  $n$ 개 있고 이 규칙으로부터 추론하여 결과값을 구하면 식(5)와 같다. 즉  $n$ 개의 추론결과를 더하는 것이다.

$$\mu_{CO}(co) = \mu_{CO1}(co_1) \cup \mu_{CO2}(co_2) \cup \dots \cup \mu_{CO_{n-1}}(co_{n-1}) \cup \mu_{CO_n}(co_n) \quad (5)$$

제어규칙에 의해서 추론된 출력값은 퍼지값이므로 이 값을 플랜트의 입력값으로 하기 위해서는 결정론적인 값 즉 비퍼지화(defuzzification)해야 한다. 본 논문에서 사용한 비퍼지화는 식(6)의 무게 중심법(center of gravity)에 의해 실행된다.

$$CO^* = \frac{\sum(\mu_s \times \mu_v)}{\sum \mu} \quad (6)$$

여기서  $CO^*$  : 비퍼지값,  $\Sigma$  : 값의 합,  $\mu$  : 소속함수 값,

$\mu_s$  : 소속함수값을 갖는 대집합,  $\mu_v$  : 해당 대집합의 소속 함수값

## 2.2 신경 회로망

본 논문에서 사용한 다층 신경망(multi layer neural network)의 구조는 그림 1과 같이 입력층, 은닉층, 출력층으로 구성되고 다층 신경망의 학습방법으로는 오차 역전파(error back propagation) 알고리즘을 사용한다. 오차 역전파 알고리즘은 학습시킬 목표치(desired value)와 신경망의 실제 출력치의 차이에 의한 오차항을 역전파시켜서 신경망의 연결 가중치를 수정한다. 오차 역전파 알고리즘에서 순방향은 신경망의 실제출력을 구하는 부분이고 역전파는 실제출력과 목표치와의 오차를 구하여 역전파 시켜서 델타규칙(Delta rule)으로 신경망의 연결 가중치를 수정하는 부분이다.

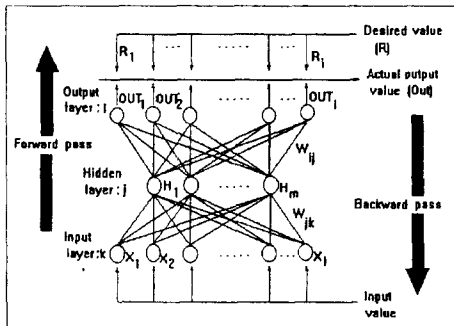


그림 1. 다층 신경망의 구조

fig. 1 Structure of multi-layer neural network

순방향 부분은 먼저 은닉층 신경세포의 출력을 구하고 난 다음 출력층 신경세포의 출력인 실제 출력을 구한다. 은닉층 신경세포 출력  $H_j$ 는 식(7)과 같다.

$$H_j = f\left(\sum_{k=1}^l W_{jk} \cdot X_k\right) \quad (7)$$

출력층 신경세포 출력  $OUT_i$  는 식(8)로 구해진다.

$$OUT_i = g\left(\sum_{j=1}^m W_{ij} \cdot H_j\right) \quad (8)$$

여기서,  $f(\cdot)$ 와  $g(\cdot)$  는 임계함수이다.

역전과 학습부분은 먼저 최소오차 자승법으로 평가함수를 구한다음 델타규칙으로 출력층과 은닉층 신경세포의 연결 가중치를 수정하고 그 다음 일반화된 델타 규칙으로 은닉층과 입력층 신경세포의 연결가중치를 수정한다.

최소오차 자승법으로 평가함수는 식(9)와 같다.

$$E = \frac{1}{2} \sum_{i=1}^n (R_i - OUT_i)^2 \quad (9)$$

출력층 신경세포와 은닉층 신경세포사이의 연결가중치는 델타규칙으로 식(10)과 식(11)에 의해서 수정되고 연결 가중치의 수정은 음의 gradient 하강법에 따른다.

$$W_{ij}(n) = W_{ij}(n-1) + \Delta W_{ij}(n) \quad (10)$$

$$\begin{aligned} \Delta W_{ij}(n) &= -\eta \frac{\partial E}{\partial W_{ij}} + \alpha \Delta W_{ij}(n-1) \\ &= \eta \delta_i H_j + \alpha \Delta W_{ij}(n-1) \\ &= \eta (R_i - OUT_i) g'(NET_i) H_j + \alpha \Delta W_{ij}(n-1) \end{aligned} \quad (11)$$

중간층 신경세포와 입력들사이의 연결 가중치는 일반화된 델타규칙으로 식(10) 과 식(11)에 의해 수정된다.

$$W_{jk}(n) = W_{jk}(n-1) + \Delta W_{jk}(n) \quad (12)$$

$$\begin{aligned} \Delta W_{jk}(n) &= -\eta \frac{\partial E}{\partial W_{jk}} + \alpha \Delta W_{jk}(n-1) \\ &= \eta \delta_j X_k + \alpha \Delta W_{jk}(n-1) \\ &= \eta f'(NET_j) \sum_{i=1}^n (\delta_i W_{jk}) X_k + \alpha \Delta W_{jk}(n-1) \end{aligned} \quad (13)$$

$$NET_i = \sum_{j=1}^m W_{ij} \cdot H_j, \quad NET_j = \sum_{k=1}^l W_{jk} \cdot X_k$$

여기서  $R_i$  : 설정치,  $\eta$  : 학습율 또는 학습속도계수,  $\delta_i$  : 출력층의 오차항,  $\delta_j$  : 은닉층의 오차항,  $\alpha$  : 모멘텀항에서의 모멘트 계수

### 2.3 EP

EP는 생물학적 유전인자인 염색체에 해당하는 문자열을 가지고 생물과 같은 재생산(reproduction), 교배(crossover), 돌연변이(mutation)를 통하여 진화과정을 발전하면서 다음 세대의 새로운 자손을 만들어 낸다. GA(Genetic Algorithm)와 EP는 전역적(global)인 최적의 해를 발견할 높은 가능성을 가진다는 장점과 목적 함수값(objective function value)에 대한 수학적 제약이 거의 없기 때문에 여러 분야에 적용할 수 있다[8][9][10]. GA, EP의 성능은 교배율, 돌연변이율, 해집단의 크기, 각 파라미터의 값, 목적함수 설정과 적합도 값의 scaling 등에 좌우된다.[8][9] 특히 GA에서는 교배율이 큰 영향을 미치고 EP에서는 돌연변이율이 큰 영향을 미친다.

EP에서 전역적인 최적의 해를 구하기 위한 순서는 다음과 같다[9][11].

- ① 부호화 및 초기화 : 염색체에 해당하는 파라미터를 랜덤한 실수값의 문자열로 초기 집단을 생성하여 초기 부모 집단을 형성한다.
- ② 재생산 : 실수값의 문자열을 목적함수에 대한 적합도(fitness)를 계산한다. 높은 적합도를

가진 문자열에 대하여 다음 세대로 복제될 확률을 높게 한다.

- ③ 교배 : 랜덤하게 생성된 초기 부모 집단에서 임의의 부모 두 개를 선택하여 교배확률( $P_c$ )에 의해 교배를 시킨다. 이와 같은 반복적 동작에 의해 새로운 자손을 생성시켜 나간다.
- ④ 돌연변이 : EP에서 가장 추가되는 동작으로서, 실수값의 문자열을 돌연변이 확률에 의해 각 값을 변화시킨다. 이와같은 동작은 과거에 잃어 버렸던 정보를 되찾을 수도 있으며, 탐색의 방향이 지역 최소치에 빠지는 경우나, 또는 조기 수렴하는 형태를 방지하는 역할을 한다.
- ⑤ 평가 : 이렇게하여 생성된 자손을 디코딩하여 목적함수에 대한 적합도를 계산한다.
- ⑥ 배열 : 부모와 자손 집단 모두에 대하여 계산되어진 문자열의 적합도값을 비교하여 높은 값부터 순위를 정한다.
- ⑦ 선택 : 유전자 문자열 ⑥에서 만들어진 순위에서 적합도 값이 높은 상위 반을 선택하여 부모 집단으로 교체한다. 목적함수가 최대가 될 때까지 단계 ②로 진행한다.

### III. FLC에 NN을 결합한 FNNIC

FLC에 NN을 결합하여 FLC의 소속함수 형태를 수정하여 플랜트에 적용하는 지능제어 신호를 구하는 지능제어기 FNNIC 구조는 그림 2와 같다.

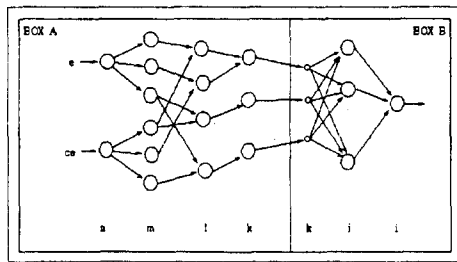


그림 2. FNNIC의 구조  
fig. 2 Structure of FNNIC

그림 2의 BOX B에서는 FLC의 비퍼지화를 연산하고 3층의 NN으로 구성하며 오차 역전파 방법을 사용한다. 성능평가 척도 E는 기대되는 값  $\delta_i$ 와 실제 출력값  $O_i$ 의 차의 제곱합을 식 (14)와 같이 사용한다.

$$E = \frac{1}{2} \sum (\delta_i - O_i)^2 \quad (14)$$

BOX B에서 K층은 BOX A의 K와 같은 층이고 K층 뉴런에는 조건부와 입력의 부합정도가 입력으로 들어온다. j층은 은닉층이며 i층의 뉴런은 부합정도에 따라 결론부가 합성되어 비퍼지화된 것과 같은 값이 출력된다. 그림 2의 BOX A에서는 입력과 퍼지 소속정도를 계산하고 4개의 층으로 구성되며 층과 층사이의 노드간 연결 강도는 1이다. 각 층의 기능은 다음과 같다.

k 층 : 결론부에 가지고 있는 퍼지규칙들의 소속정도를 MAX 연산에 의해서 합성한다. 노드의 수는 출력변수에 정의된 언어항의 개수와 같다.

l 층 : 퍼지규칙의 조건부에 해당하고 m층에서 계산된 소속정도를 MIN 연산으로 합성을 한다.

m 층 : 입력의 언어항에 대한 소속정도는 식(15)에서 계산하며 x는 입력 값이다.

$$\mu^{C^m, S_L^m, S_R^m}(x) = \begin{cases} 1 - \frac{x - C^m}{S_R^m} & x \in [C^m + S_R^m] \\ 1 + \frac{x - C^m}{S_L^m} & x \in [C^m - S_L^m, C^m] \\ 0 & \text{그외} \end{cases} \quad (15)$$

여기서  $C^m, S_L^m, S_R^m$  : 삼각퍼지 함수의 중앙점, 좌측폭 및 우측폭

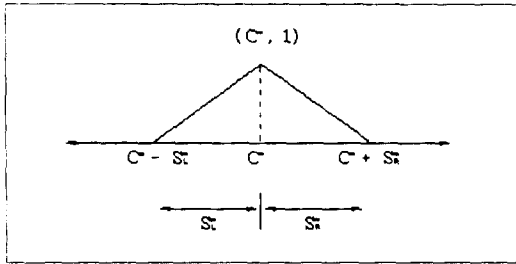


그림 3. 조건부 언어항의 삼각퍼지 소속함수

fig. 3 Triangle Fuzzy membership function of conditional linguistic term  
n 층 : 입력변수를 받는다.

소속함수 수정을 위한 학습방법으로 최대경사법(maximum gradient descent method)을 사용하고, 직접적인 방법으로 퍼지 신경회로망의 출력에 대한 성능평가를 할 수 없으므로 이미 NN으로 학습된 비퍼지화 연산부분의 오차를 역전파 시켜서 입력과 각 규칙의 소속정도에 대한 오차값을 간접적으로 계산하는 방법을 사용한다. 따라서 오차 역전파 알고리즘에 의한 비퍼지화 부분의 각 층에서 역전파되는 오차들은 다음과 같다.

$$\text{출력층의 오차 : } \epsilon_i = -\frac{\partial E}{\partial \text{net}_i} = -\frac{\partial E}{\partial o_i} \frac{\partial o_i}{\partial \text{net}_i} = (\delta_i - o_i) f'(\text{net}_i) \quad (16)$$

$$\text{중간층의 오차 : } \epsilon_j = -\frac{\partial E}{\partial \text{net}_j} = -\frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} = \sum_i \epsilon_i W_{ji} f'(\text{net}_j) \quad (17)$$

$$\text{입력층의 오차 : } \epsilon_k = -\frac{\partial E}{\partial \text{net}_k} = \sum_j \epsilon_j W_{kj} \quad (18)$$

소속함수를 시스템에 맞도록 형태를 수정하기 위해 m층의 각 파라미터 ( $C^m, S_L^m, S_R^m$ )를 연쇄 법칙(chain rule)에 의해 수정한다. 각 파라미터의 증분항 계산은 다음과 같다.

$$\begin{aligned} \Delta C^m &= -\eta \frac{\partial E'}{\partial C^m} = -\eta \sum_k \frac{\partial E'}{\partial o_k} \frac{\partial o_k}{\partial C^m} = -\eta \sum_k \frac{\partial E'}{\partial o_k} \sum_l \frac{\partial o_k}{\partial o_l} \frac{\partial o_l}{\partial C^m} \\ &= \eta \sum_k \epsilon_k \sum_l \frac{\partial o_k}{\partial o_l} \frac{\partial o_l}{\partial C^m} \end{aligned} \quad (19)$$

$$\Delta S_L^m = -\eta \frac{\partial E'}{\partial S_L^m} = -\eta \sum_k \epsilon_k \sum_l \frac{\partial o_k}{\partial o_l} \frac{\partial o_l}{\partial S_L^m}, \quad \Delta S_R^m = -\eta \frac{\partial E'}{\partial S_R^m} = -\eta \sum_k \epsilon_k \sum_l \frac{\partial o_k}{\partial o_l} \frac{\partial o_l}{\partial S_R^m} \quad (20)$$

$$\frac{\partial o_k}{\partial o_l} = \begin{cases} 1 & \text{if } o_k = o_l \text{ and } w_{lk} = 1 \\ 0 & \text{otherwise} \end{cases} \quad \frac{\partial o_l}{\partial o_m} = \begin{cases} 1 & \text{if } o_l = o_m \text{ and } w_{ml} = 1 \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

NN의 오차 역전파 알고리즘은 최대 경사법이라는 수학적 방법에 근거함으로 지역최소점(Local minimum)에 빠질수 있는 단점을 가진다. 이 단점을 예방하기 위하여 본 논문에서는 전역적인 최적해를 구하는데 우수한 EP를 사용하여 NN을 학습시킨다.

#### IV. FNNIC에 EP를 결합한 EP-FNNIC

본 논문에서 제안한 FNNIC에 EP를 결합하여 지능적인 퍼지 소속함수를 구하는 EP-FNNIC

구조는 그림 4와 같다. EP-FNNIC 구조에서 NN과 EP는 다음과 같은 문제점을 가지고 있다. NN은 인공지능적인 기능을 가지고 있지만 학습시간이 길고 지역최소점에 빠질수 있는 문제로 최적화된 해를 구하지 못하는 경우가 생긴다. EP는 전역적인 최적해를 구하는데 있어서는 가장 적합한 algorithm 이지만 적용제어를 위한 on-line 방식을 채택할 수 없는 단점을 가진다. 그래서 이들의 문제점을 서로 보완하기 위하여 먼저 EP를 이용하여 최적의 소속함수 해를 구해서 불확실한 플랜트에 대한 근사적 소속함수를 구한다. 이 근사적 소속함수값을 초기치로하여 지능제어 구조를 갖인 FNNIC가 플랜트에 적용하는 소속함수값을 구해서 최적화된 지능제어를 실행 시킨다.

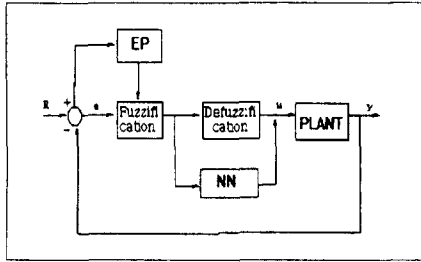
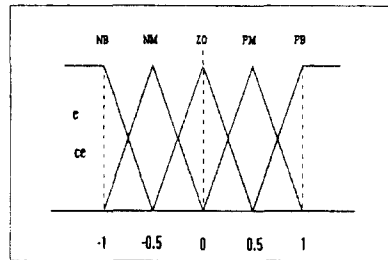


그림 4. EP-FNNIC의 구조  
fig. 4 Structure of EP-FNNIC

	c				ce			
	a	b	.....		.....		i	j
	PM center	PB center					PM right	PB left
POP 1	0.85	0.90	.....		.....		0.75	0.65
POP 2	0.81	0.88	.....		.....		0.75	0.66
POP A-1	0.89	0.85	.....		.....		0.77	0.69
POP A	0.87	0.84	.....		.....		0.79	0.71



(a) 10진 형태의 구조  
(a) Gene structure of decimal type

(b) e 와 ce의 퍼지 소속함수 형태  
(b) Triangle Fuzzy membership type of e and ce

그림 5. 10진 형태의 유전자 구조와 FLC의 삼각소속함수 형태  
fig. 5 Gene structure and FLC triangle membership type of decimal type

유전자 구조는 그림 5의 (a)와 같이 설정하였다. 여기서 각 유전자는 m개의 염색체를 갖는다. 오차(e)와 오차변화(ce)의 모든 퍼지소속함수에 대한 중심과 폭을 해로 하여 염색체를 설정할 경우 정상상태의 해를 얻기 위해서는 많은 세대를 반복해야 하므로 본 논문에서는 10개의 염색체로 줄여서 구성하였고 다음과 같이 설정한다.[12]

- a : e의 PM 중심 , b : e의 PB 중심 , c : e의 PM 좌폭 , d : e의 PM 우폭 ,
- e : e의 PB 좌폭 , f : ce의 PM 중심 , g : ce의 PB 중심 , h : ce의 PM 좌폭 ,
- i : ce의 PM 우폭 , j : ce의 PB 좌폭

각 염색체는 실수형 문자열 형태의 유전자 구조를 그림 5(a)와 같이 각 소속함수에 3개씩의 실수형 문자열을 배당하여 2진 비트에 대한 디코딩 작업을 생략하여서 세대진화 과정을 빠르게 하였다. 이렇게 얻어진 각 염색체 정보는 다음과 같은 시스템의 안정화를 위한 제약사항과 더불어 소속함수의 형태를 결정하였다. 먼저 ZO를 중심으로 좌측과 우측을 대칭으로 설정하였다. 즉 양의 부분만을 얻어서 음의 부분을 설정하였다. 그리고 ZO의 중심은 고정시키고 ZO의 좌, 우폭 값은 PB의 중심에서 PB의 좌폭값을 뺀 것으로 처리하였다. 시스템의 안정화를 위하여 PM의 좌폭은 PM의 중심값보다 크지 않도록 하였다.[12] 초기 세대 유전자는 안정화된 세대진화를 위하

여 각 퍼지수에 적절한 영역을 설정하여 다음 세대를 생성하도록 하였다. 세대를 진화시키는 요소는 앞의 염색체 구조와 교배율, 돌연변이율 그리고 다음의 평가함수에 의해 크게 좌우된다. 이 평가함수는 시스템에 있어서 적합한 유전자인가를 평가할 수 있는 적합도 함수이다. 여기서 사용한 평가함수는 오차와 오차변화를 최소화 하는 것을 목적으로 한다. 따라서 식(22)와 같이 오차와 오차변화의 제곱 누적을 평가함수의 항으로 사용하였다. 여기서,  $K_1 + K_2 = 1$  로 하여  $K_1$ ,  $K_2$  를 적절하게 설정한다.

$$F(\text{pop}_i) = \frac{K_1}{1 + \sum_k^{\text{step}} e_k^2} + \frac{K_2}{1 + \sum_k^{\text{step}} ce_k^2} \quad (22)$$

## V. 실험 및 결과

시뮬레이션에서 사용한 가상 2차계 플랜트 차분방정식은 식(23)과 같다.

$$Y(k) = 1.8187 * Y_1 - 0.8187 * Y_2 + 0.5 * U_1 - 0.45 * U_2 \quad (23)$$

여기서  $Y_1$  : 현재의 플랜트 출력,  $Y_2$  : 현재보다 한 샘플링 시점 이전의 플랜트 출력

여기서  $U_1$  : 현재의 제어기 출력,  $U_2$  : 현재보다 한 샘플링 시점 이전의 제어기 출력

설정치 R은 40과 80의 계단파로 정하였고 샘플구간 1에서 300까지는 설정치 40, 샘플구간 301에서 600까지는 설정치 80을 주었고 외란은 설정치 10분의 1로 가하였다. 샘플수는 600으로 하였으며 외란은 샘플구간 150에서 160까지, 450에서 460까지 2번 인가 하였고 샘플링 시간은 1[ms]이다. 그림 6은 퍼지 제어기의 출력응답이다. 그림 7은 EP-FNNIC 제어기의 출력응답으로서 이때 각각의 파라미터 값은 다음과 같다. 교배율 = 0.1, 돌연변이율 = 0.4, 한 세대의 유전자수 = 30개, 9세대 7개체 까지 진화, 비퍼지화부의 신경망 오차 역전파 학습에서 모멘텀율 = 0.7, 학습율 = 0.42, 학습수 = 1,500이다. 그림 8은 그림 6의 퍼지 제어기 출력시 소속함수 모양이고 그림 9는 그림 7의 EP-FNNIC 제어기의 출력시 소속함수 모양이다.

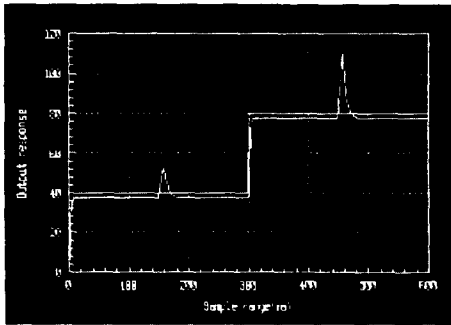


그림 6. 퍼지 제어기의 출력응답  
fig.6 Output response of Fuzzy controller

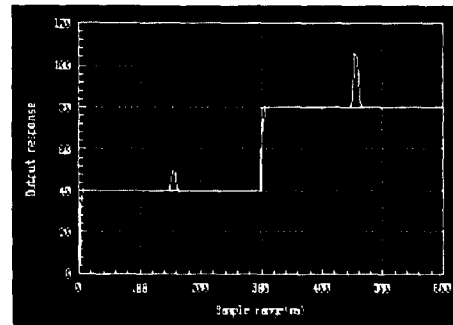


그림7. EP-FNNIC 제어기의 출력응답  
fig.7 Output response of EP-FNNIC controller



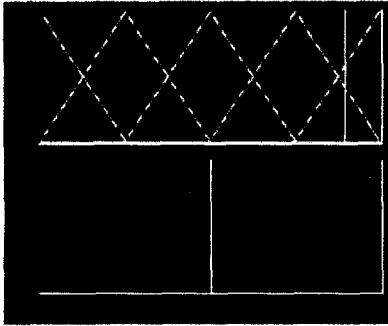


그림 8. 퍼지 제어기의 소속함수 형태  
fig.8 Membership function type  
of Fuzzy controller

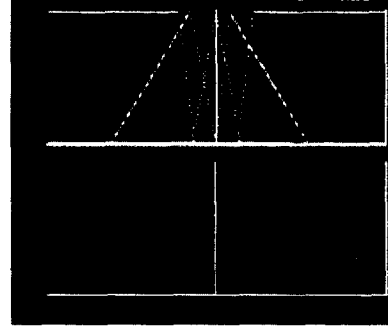


그림 9. EP-FNNIC 제어기의 소속함수 형태  
fig.9 Membership function type  
of EP-FNNIC controller

실험에서 각 제어기의 성능을 비교 검토한 결과는 다음과 같다.

1) 퍼지 제어기

- ① 초기의 과도응답 진동은 없다.
- ② 상승시간은 12[ms]이다.
- ③ 첫 번째 외란을 인가한 경우 최대 진동폭은  $52.5 - 40 = 12.5$ 이고, 두 번째 외란 인가시 최대 진동폭은  $108.5 - 80 = 28.5$ 이다. 외란을 인가한 후 정상상태 수렴시간은 첫 번째 외란후 37[ms]이고, 두 번째 외란후는 40[ms]이다.
- ④ 초기 설정치와 다른 계단파의 설정치를 인가한 경우에도 과도응답 진동은 없고 상승시간이 15[ms]이다.
- ⑤ 퍼지 제어기의 규칙 고정으로 정상 상태에서는 2.4의 오차값을 항상 가진다.

2) EP-FNNIC 제어기

- ① 초기의 최대 과도응답은 0.5로 2[ms]동안 진동하고 정상상태에 수렴한다.
- ② 상승시간은 8[ms]로 아주 빠르게 도달한다.
- ③ 첫 번째 외란을 인가한 경우 최대 진동폭은  $49.4 - 40 = 9.4$ 이고, 두 번째 외란 인가시 최대 진동폭은  $106 - 80 = 26$ 이다. 외란을 인가한 후 정상상태 수렴시간은 첫 번째 외란시 12[ms]이고, 두 번째 외란시는 14[ms]이다.
- ④ 초기 설정치와 다른 계단파의 설정치를 인가한 경우 최대 과도응답은 0.5로 2[ms]동안 진동하고 상승시간은 8[ms]이다.
- ⑤ 신경망 퍼지 지능제어기로 적응학습을 하므로 정상상태에 정확히 수렴하여 안정된 정밀제어를 한다.

## VI. 결론

본 논문에서는 FLC의 성능을 향상시키기 위하여 FLC에 NN과 EP를 결합한 EP-FNNIC 인공지능 제어기를 제안하였다. EP-FNNIC에서 NN은 함수근사화와 학습의 능력을 가져 지능제어를 할 수 있는 장점도 있지만 학습시간이 너무 길고 지역최소점에 빠질수 있는 단점도 가지고 있다. 반면에 EP는 전역적인 최적해를 구하는것에는 가장 적합한 algorithm이지만 off-line 구조를 가지므로 on-line 적용제어 system을 구성할 수 없다. 이러한 문제를 해결하기 위하여 본 논문에서는 EP를 이용하여 NN이 지역최소점에 빠지는것을 예방하여 전역적인 최적해를 구해서 불확실한 플랜트에 대한 근사적 소속함수값을 찾았다. 이 근사적 소속함수값을 초기치로하여 FNNIC의 on-line 제어기 구조로 플랜트에 적용하는 소속함수를 구하였다. 실험 결과 FLC는 목표치에서

미세한 오차를 가지고 있지만 EP-FNNIC은 목표치에 오차없이 정확히 수렴하여 안정된 지능제어가 실현됨을 알 수 있었다. 또한 EP-FNNIC이 FLC보다 수렴속도와 상승시간도 빠르며 외란에도 강한것을 실험적으로 알 수 있었다. 그러나 EP는 해집합 선택 방법에서 해집합이 클 경우 최적화 하는데 많은 시간이 소모된다. 또한 적당한 교배율과 돌연변이율을 찾는것도 많은 시간이 소모되므로 단 시간내에 이러한 파라미터 값을 찾는 것도 과제이다. EP는 off-line 구조이므로 제어 system에 응용할 경우 보다 효율적인 on-line 제어방식을 찾는 것도 앞으로의 연구과제이다.

## VII. 참고 문헌

- 1]. 이진명, 곽동훈, 이광형, "퍼지 추론 및 규칙조정을 위한 퍼지 신경회로망", 인공지능·신경망 및 퍼지시스템 종합 학술대회 논문집, pp. 78-82, 1993.
- 2]. 이영노, 김태완, 서일홍, "룩업 테이블을 이용한 자동학습 퍼지제어기의 설계에 관한연구", 전자 공학회 논문지, 제29권, B편, 제9호, pp. 723-734, 1992.
- 3]. 이상부, 김현정, 임영도, "퍼지와 비례제어기를 결합한 Hybrid 제어기의 성능향상에 관한연구", 인공지능 신경망 및 퍼지시스템 추계종합학술대회 논문집, pp. 183-188, 1993.
- 4]. 이상부, 김형수, "퍼지제어기와 신경망제어기의 응답특성에 관한연구", 정보처리학회 논문지, Vol.3, No.6, pp. 1473-1482, 11, 1996.
- 5]. James J. Buckley, Yoichi Hayashi, "Fuzzy neural networks : A survey", Fuzzy Sets and Systems, No. 66, pp. 1-11, 1994.
- 6]. Katsuhiko Ogata, Discrete-time Control System, Prentice-Hall, INC., pp. 198-230, 1987.
- 7]. 李勇, 米澤洋, "DC 모터 サーボ系の ファジィ 制御", 日本ファジィ 學會誌, Vol.2, No.4, pp. 146-151, 1990.
- 8]. David E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, INC., pp. 27-82, 1989.
- 9]. 이상부, 임영도 공저, 퍼지 신경망 유전진화, 도서출판 영과일, pp. 193-215, 1996.
- 10]. L. Davis, Handbook of Genetic Algorithms, Van Nostrand Reinhold, 1991.
- 11]. Zbigniew Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, INC., pp. 65-69, 1992.
- 12]. 이상부, 김승철, 추연규, 임영도, "진화프로그래밍을 이용한 퍼지제어기 소속함수의 추정에 관한 연구", 대한전자공학회 한국통신학회 부산경남지부 추계합동학술대회 논문집, Vol.5, No.1, pp. 187-194, 1996.