

2000년도 대한전기학회 전문대학교육연구회 하계학술대회 논문집 2000. 7. 7-8

## 실수형 유전알고리즘과 신경회로망을 이용한 적응 퍼지제어기의 설계

°남정락\*, 김동완\*, 활기현\*\*, 안호균\*\*\*  
 °동명대학 전기전자계열\*, 부산대학교 전기공학과\*\*, 창원대학교 전기공학과\*\*\*

### Design of Adaptive Fuzzy Logic Controller Using Real-Coding Genetic Algorithm and Neural Network

°Jing-Rak Nam\*, Dong-Wan Kim\*, Gi-Hyun Hwang\*\*, Ho-Kyun Ahn\*\*\*  
 Tongmyong college\*, Pusan National Univ.\*\*, Changwon National Univ.\*\*\*

**Abstract** - 본 논문에서는 진화연산 중에서 해의 다양성과 수렴속도면에서 좋은 성능을 나타내는 실수형 유전 알고리즘과 신경회로망을 이용한 적응 퍼지제어기를 설계하였다. 실수형 유전알고리즘을 이용하여 퍼지제어기의 입·출력 이득과 실시간으로 퍼지제어기의 입·출력 이득을 적응적으로 변경하는 신경회로망의 가중치를 튜닝하였다. 제안한 방법의 유용성을 평가하기 위해 시지연을 갖는 제어시스템[14]에 적용하였다. 컴퓨터 시뮬레이션 결과, 제안한 적응 퍼지제어기가 기존의 퍼지제어기보다 오버슈트, 정정시간, 상승시간면에서 더 우수한 제어성능을 나타내었다.

#### 1. 서 론

제어 시스템은 일반적으로 제어하고자 하는 대상 시스템의 정확한 수학적 모델링을 필요로 한다. 그러나 실제 시스템이 비선형이고 시변인 경우에는 그 시스템을 정확하게 모델링을 하기가 어렵고, 이득이 일정한 선형제어기만으로는 고성능의 제어특성을 기대하기 어렵다. 이와 같은 문제점을 해결하기 위해서 PID 제어기의 이득을 제어시스템의 변화에 따라 적절히 조정하는 적응제어(adaptive control)방법이 연구되어 좋은 결과를 얻고 있다[1-3]. 그리고 비선형 제어방법으로 주목받고 있는 가변구조(variable structure)제어는 미리 정해진 스위칭 평면상에서 시스템의 구조를 바꾸어 줌으로써 시스템의 상태가 슬라이딩 평면으로 주어지는 동특성을 갖게 하는 제어법인데, 슬라이딩 평면에서 고속스위칭을 하는 경우 스위칭 소자의 손실과 스위칭 동작에 의해 발생되는 멀림(chattering)현상으로 인해서 모델링시 무시되었던 고주파 동특성을 야기시킬 수 있는 문제점을 가지고 있다[4,5].

최근에는 전문가의 경험이나 지식을 바탕으로 인간의 의사결정을 모방한 퍼지 제어기와 인간의 두뇌 구조를 모의하여 인지적 정보 처리 구조를 갖는 신경회로망 제어기와 같은 지능제어 이론을 비선형 시스템에 적용하여 좋은 제어 성능을 얻고 있다[6-9]. 그러나 퍼지제어기가 좋은 제어 성능을 얻기 위해서는 퍼지규칙과 퍼지변수의 소속함수 모양을 조정해야하는데, 종래에는 전문가의 경험과 시행착오법에 바탕을 두었기 때문에 시스템에 대한 전문적인 지식 없이는 시스템에 적합한 소속함수를 결정하기가 어려웠다. 그래서 최근에는 자연의 유전학과 진화이론에 바탕을 둔 확률적인 최적화 알고리즘인 진화연산을 이용하여 퍼지규칙과 퍼지변수의 소속함수 모양을 튜닝하여 좋은 성능을 얻고 있다[10,11].

자연생태계의 진화를 모의한 확률적인 최적화 탐색 알고리즘인 진화연산(evolutionary computation:EC)은 유전알고리즘(genetic algorithm:GA), 진화 전략(evolution strategy:ES), 진화 프로그래밍(evolutionary programming:EP) 등의 기법으로 대별된다[12,13]. 이들 알고리즘들의 유사점은 주어진 최적화 문제에 대해 하나의 해가 아닌 해집단(population)으로 탐색하며, 해집단내에

서 더 좋은 적합도(fitness)를 가지는 해들을 이용하여 새로운 해집단을 생성한다는 점이다. 진화연산은 해집단을 이용하여 해를 탐색함으로써 전역최적해를 찾을 가능성이 높은 확률적인 최적화 알고리즘으로 목적함수의 미분가능, 연속성에 영향을 받지 않는다.

본 논문에서는 진화연산 중에서 해의 다양성과 수렴속도면에서 좋은 성능을 나타내는 실수형 유전알고리즘과 신경회로망을 이용한 적응 퍼지제어기(adaptive fuzzy logic controller : AFLC)를 설계하였다. 실수형 유전 알고리즘을 이용하여 퍼지제어기의 입·출력 이득과 실시간으로 퍼지제어기의 입·출력 이득을 적응적으로 변경하는 신경회로망의 가중치를 튜닝하였다. 제안한 방법의 유용성을 평가하기 위해 시지연을 갖는 제어시스템[14]에 적용하였다. 컴퓨터 시뮬레이션 결과, 제안한 적응 퍼지제어기가 기존의 퍼지제어기보다 오버슈트, 정정시간, 상승시간면에서 더 우수한 제어성능을 나타내었다.

#### 2. 본 론

##### 2.1 실수형 유전알고리즘

자연의 유전학 및 자연도태의 원리에 근거한 실수형 유전알고리즘(real-coding genetic algorithm:RGA)은 확률적인 방법과 개체간의 체계적인 정보 교환을 통해 탐색 공간을 조사해 나감으로써 주위환경에 알맞은 가장 적절한 해를 얻고자 하는 최적화 알고리즈다. 본 논문에서 사용한 유전알고리즘은 실변수형 코딩방식으로 해집단을 이진수의 스트링이 아닌 실변수로 부호화하였다. 실수형 유전알고리즘을 이용하여 최적해를 구하는 과정은 다음과 같다.

단계 1) 초기화(initialize) : 주어진 문제의 제약 조건을 고려하여 랜덤하게 초기 해집단을 생성한다.

단계 2) 평가(evaluation) : 각 스트링을 주어진 목적 함수에 따라 적합도를 계산한다.

단계 3) 복제(reproduction) : 자연 생태계의 적자생존의 원리를 이용한 것으로 높은 적합도를 가진 스트링일수록 다음 세대에 복제될 확률이 높다. 본 논문에서 사용한 복제방법은 적합도에 비례하여 복제하는 룰렛휠(roulette wheel) 방법을 사용한다.

단계 4) 교배(crossover) 및 돌연변이(mutation) : 복제한 후 일정한 확률을 가지고 스트링을 변화시켜 새로운 해집단을 생성하는 과정이다. 본 논문에서 사용한 교배 및 돌연변이 방법으로 단순교배와 균일 돌연변이를 사용하였다[13].

단계 5) 수렴조건 : 일정한 세대수만큼 단계 2) ~ 단계 4)를 반복한다.

## 2.2 퍼지제어기 및 전방향 신경회로망

퍼지제어기는 시스템의 특성이 복잡하여 기존의 정량적인 방법으로는 해석할 수 없거나 얻어지는 정보가 정성적이고 불확실한 경우와 기존의 수학적 모델링을 기초로 한 제어기와는 달리 시스템의 입·출력 관계만으로도 제어기를 설계할 수 있다. 퍼지제어기는 일반적으로 퍼지화(fuzzifier), 퍼지추론(fuzzy reasoning) 및 비퍼지화(defuzzification) 과정으로 구성되어 있으며, 본 논문에서는 사용한 퍼지화 방법으로 이동변 삼각형 방법, 퍼지추론 방법으로 Mamdani의 최대-최소 퍼지추론 방법, 비퍼지화 방법으로 무게중심법을 사용하였다. 신경회로망은 인간의 두뇌 구조를 모의하여 인지적 정보 처리 구조를 갖는 시스템으로써, 주어진 입력들에 가중치를 조정하여 원하는 출력력을 생성하는 학습 기능과 왜곡되거나 잡음을 가진 입력에 대해 바른 인식이 가능하며 또한, 불완전한 입력에 대해서도 학습에 의해 가장 유사한 출력값을 낼 수 있는 일반화 기능을 가지고 있다. 본 논문에서 이용한 전방향 신경회로망의 구조는 입력층, 은닉층, 출력층, 수렴 속도를 개선하기 위한 바이어스 항으로 구성하였다. 전방향 신경회로망의 학습 알고리즘은 출력층에서 실제로 계산된 출력값과 학습쌍에 의해 제시된 기준 입력값과의 오차를 줄이는 방향으로 가중치를 조정하는 방법으로써 현재 널리 이용되고 있는 확률적인 최적화 알고리즘인 실변수 유전알고리즘을 이용하였다.

## 3. 실수형 유전알고리즘과 신경회로망을 이용한 적응 퍼지제어기의 설계

일반적으로 퍼지제어기는 외란 및 제어시스템의 파라메타 변동에 대해서 뛰어난 제어 성능을 나타내지만, 다양한 외란 및 제어시스템의 파라메타 변동에 대해서 최적의 제어 성능을 얻을 수 없다. 따라서 본 논문에서는 다양한 외란 및 제어시스템의 파라메타 변동에 대해서 최적의 제어 성능을 얻기 위해서 퍼지제어기의 입·출력 이득을 전방향 신경회로망을 이용하여 실시간으로 매 샘플링시간마다 적응적으로 변경하는 방법을 제안하였다. 그리고 퍼지제어기의 입·출력 이득과 전방향 신경회로망의 가중치를 반복학습 방법으로 이진수 유전알고리즘보다 구현하기가 쉽고 해의 다양성과 수렴속도면에서 뛰어난 성능을 실수형 유전알고리즘을 이용하여 투닝하였다. 그림 1은 실수형 유전알고리즘을 이용하여 퍼지제어기의 입·출력 이득과 신경회로망의 가중치를 투닝하기 위한 구조도를 나타낸다. 그림 1에서 보는 것처럼 퍼지제어기의 입력으로 오차와 변화분을 사용하였고, 전방향 신경회로망의 입력을 기준입력과 실제 출력사이의 오차를 사용하였다. 그리고 퍼지제어기의 입·출력 이득 및 전방향 신경회로망의 가중치를 학습하기 위해 사용된 실수형 유전알고리즘의 입력으로 오차의 절대치합을 이용하였다. 그리고 전방향 신경회로망의 출력은 식(1)과 같이 퍼지제어기의 입·출력 이득의 변화분이다. 식(1)에서 보는 것처럼 퍼지제어기의 입·출력 이득은 전방향 신경회로망에 의해 실시간으로 매 샘플링시간마다 적응적으로 변경됨을 알 수 있다. 이렇게 전방향 신경회로망을 이용하여 매 샘플링시간마다 퍼지제어기의 입·출력 이득을 변경한 이유는 변화하는 제어시스템에 대해서 최적의 제어성능을 얻기 위해서이다.

$$SF_i(t) = SF_i(t) + \Delta SF_i(t) \quad (1)$$

여기서,  $SF_i(t)$  : 퍼지제어기의 입·출력 이득

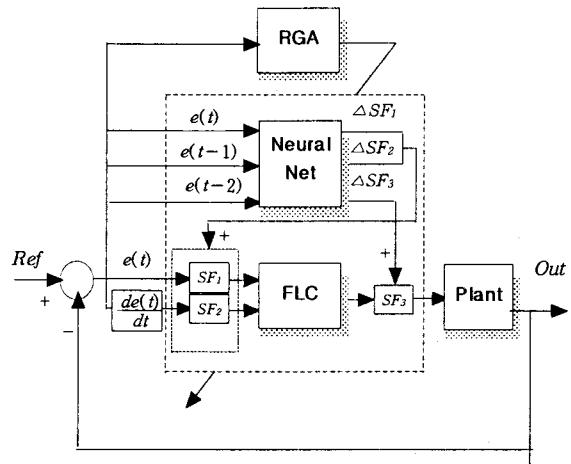


그림 1 실수형 유전알고리즘을 이용한 AFLC 투닝을 위한 구조도

Fig. 1 Configuration for the tuning of AFLC using RGA

실수형 유전알고리즘을 이용하여 퍼지제어기와 전방향 신경회로망의 가중치를 투닝하기 위한 해집단의 구조는 그림 2에 나타내었다. 그림 2에서 보는 것처럼 실수형 유전알고리즘을 이용하여 퍼지제어기의 입·출력 이득을 투닝하는데 필요한 파라메타 수는 3개이고, 본 논문에서 사용한 전방향 신경회로망의 입력층 뉴런의 수는 3개, 은닉층 뉴런의 수는 5개, 출력층 뉴런의 수는 3개를 사용하였으므로 학습해야 할 파라메타 수는 38개이다. 따라서, 실수형 유전알고리즘을 사용하여 투닝해야 할 총 파라메타 수는 41개이다. 그리고 해집단의 각 스트링을 평가하기 위하여 사용된 적합도 함수는 식(2)와 같이 오차의 절대치 합을 사용하였다.

$$\text{적합도} = \frac{1}{1 + \int_{t=0}^T |Ref(t) - Out(t)|} \quad (2)$$

여기서,  $Ref(t)$ : 기준입력

$Out(t)$ : 제어대상의 출력

T: 특정한 시간 동안 획득한 데이터의 수

$S_1$	$SF_{1\_1}$	$SF_{1\_2}$	$SF_{1\_3}$	$W_{1\_1}$	$W_{1\_2}$	$\dots$	$W_{1\_38}$
$S_2$	$SF_{2\_1}$	$SF_{2\_2}$	$SF_{2\_3}$	$W_{2\_1}$	$W_{2\_2}$	$\dots$	$W_{2\_38}$
.	.	.	.	.	.	.	.
$S_n$	$SF_{n\_1}$	$SF_{n\_2}$	$SF_{n\_3}$	$W_{n\_1}$	$W_{n\_2}$	$\dots$	$W_{n\_38}$

여기서,  $SF_{ij}$  : 퍼지제어기의 입·출력 이득

$W_{ij}$  : 전방향 신경회로망의 가중치

n : 스트링 개수

그림 2 해집단의 구조

Fig. 2 Structure of population

실수형 유전알고리즘과 전방향 신경회로망을 이용하여 적응 퍼지제어기를 설계하기 위한 전체적인 흐름도는 그림 3에 나타내었고, 설계 과정은 다음과 같다.

단계 1) 해집단 구성

그림 2와 같이 퍼지제어기의 입·출력 이득과 전방향

신경회로망의 가중치를 이용하여 해집단을 구성한다.

단계 2) 평가

- 식 (2)의 적합도 함수를 이용하여 단계 1)에서 생성된 각 스트링을 평가하기 위한 과정은 다음과 같다.
- ① 퍼지제어기의 입력인 오차( $e(t)$ )와 오차의 변화분 ( $de(t)$ )을 계산한다.
  - ② 전방향 신경회로망의 입력인  $e(t)$ ,  $e(t-1)$ ,  $e(t-2)$ 를 계산한다.
  - ③ 매 샘플링 시간마다 전방향 신경회로망의 출력 ( $\Delta SF_i$ )을 계산한다.
  - ④ 퍼지추론과 비퍼지화 과정을 거쳐서 퍼지제어기의 출력을 계산한다.
  - ⑤ 비선형 미분방정식의 해법으로 제어대상 시스템을 풀다.
  - ⑥ 시뮬레이션 시간 동안 ① ~ ⑤를 반복한다.

단계 3) 복제

적합도에 비례하여 복제하는 루ول렛휠 방법을 사용하여 복제한다.

단계 5) 교배 및 돌연변이

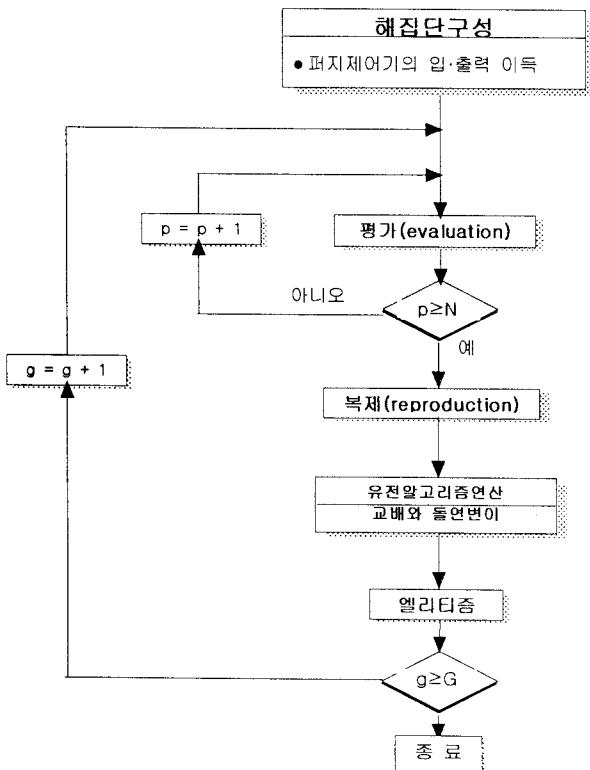
실수형 유전알고리즘의 연산자인 교배와 돌연변이를 수행하여 자손을 생성한다.

단계 6) 엘리티즘

적합도가 가장 높은 개체를 실수형 유전알고리즘의 해집단에 하나씩 복제하는 엘리티즘을 사용하였다.

단계 7) 수렴조건

특정한 세대수를 만족할 때까지 단계 2) ~ 단계 6)을 반복한다.



여기서,  $N$  : 해집단의 크기  
 $G$  : 특정한 세대수

그림 3 실수형 유전알고리즘을 이용한 적응 퍼지제어기의 설계를 위한 흐름도

Fig. 3 Flow chart for the design of AFLC using RGA.

#### 4. 사례 연구

제안한 적응 퍼지제어기의 성능을 검증하기 위해 참고문헌[14]에 있는 시지연 갖는 제어시스템을 사용하였다. 본 논문에서 사용한 시지연을 가지는 제어시스템은 식 (3) ~ 식 (5)에 나타내었고, 여기서,  $L$ 은 시지연 상수이다. 적응 퍼지제어기의 튜닝에 사용된 실수형 유전알고리즘의 시뮬레이션 계수는 표 1과 같다. 표 1과 같은 조건에서 실수형 유전알고리즘으로 적응 퍼지제어기를 튜닝하는데 소요되는 실행시간은 PC 586 500 MHz로 약 20분 정도 소요된다. 본 논문에서 제안한 적응 퍼지제어기와 성능을 비교하기 위해 참고문헌[14]에 있는 기존의 퍼지제어기와 비교·검토하였다.

##### (1) Second-Order Linear System

$$G(s) = \frac{e^{-LS}}{s^2 + s + 0.2} \quad (3)$$

##### (2) Marginally Stable System

$$G(s) = \frac{e^{-LS}}{s(s+1)} \quad (4)$$

##### (3) Second-Order Nonlinear System

$$\frac{d^2y(t)}{dt^2} + \frac{dy(t)}{dt} + 0.25y(t) = u(t-L) \quad (5)$$

표 1 실수형 유전알고리즘의 시뮬레이션 계수

Table 1 Coefficients for simulation using RGA

적용방법	해집단수	교배확률	돌연변이 확률	실행시간
RGA	20	0.95	0.05	20분

그림 4는 매 세대별 적합도 수렴특성을 나타내었다. 그림 4에서 보는 바와 같이 세대가 증가함에 따라서 적합도가 개선되고 있음을 알 수 있다. 이것은 적응 퍼지제어기가 실수형 유전알고리즘에 의해서 최적화되고 있다는 것을 증명한다.

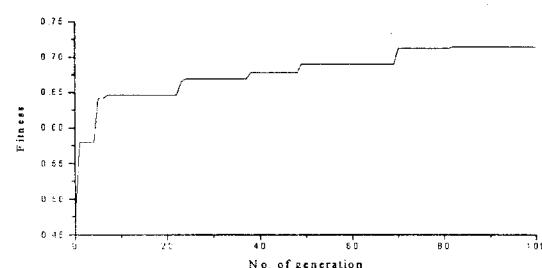


그림 4 매 세대별 적합도 추이

Fig. 4 The change of fitness in each generation

#### 4.1 Second-Order Linear System의 시뮬레이션 결과

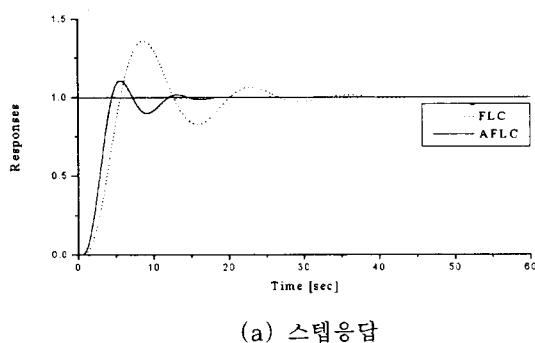
실수형 유전알고리즘에 의해서 최적화된 적응 퍼지제어기와 기존의 퍼지제어기와 제어성능을 비교하기 위해 사용된 조건은 다음과 같다. 성능 비교에 사용한 조건은 적응 퍼지제어기의 설계에 사용된 조건과 적응 퍼지제어기의 강인성 평가에 사용된 조건으로 분류한다. 적응 퍼지제어기의 설계에 사용된 조건은 실수형 유전알고리즘으로 적응 퍼지제어기의 입·출력 이득 튜닝에 사용된 조건이다. 본 논문에서 적응 퍼지제어기의 설계에 사용된 조건은 제어시스템에 대한 시지연 상수가  $L =$

$0.0[\text{sec}]$ ,  $L = 0.2[\text{sec}]$ 을 이용하였다. 적응 퍼지제어기의 강인성 평가에 사용된 조건은 시지연 상수가  $L = 0.1[\text{sec}]$ ,  $L = 0.3[\text{sec}]$ 과  $L = 0.3[\text{sec}]$ 일 때 부하가 감소한 경우를 사용하였다.

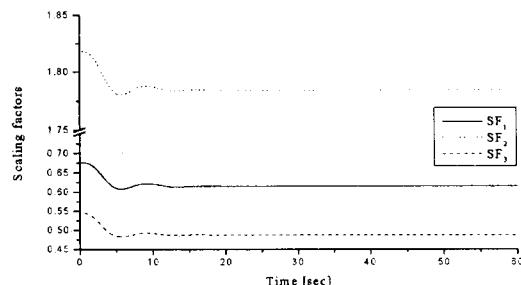
그림 5는 2차 선형시스템에 대해서 실수형 유전알고리즘으로 퍼지제어기의 입·출력 이득과 전방향 신경회로망의 가중치 튜닝에 사용된 시지연 상수가  $L = 0.2[\text{sec}]$ 일 때 스텝응답과 입·출력 이득의 변화를 보였다. 그림 5에서 보는 것처럼 적응 퍼지제어기가 오버슈트, 정정시간, 상승시간면에서 기존의 퍼지제어기보다 더 우수한 제어성능을 보였고, 적응 퍼지제어기의 입·출력 이득이 매샘플링 시간마다 적응적으로 변경되고 있음을 알 수 있다. 그림 6의 (a)는 2차 선형시스템에서 시지연 상수가  $L = 0.3[\text{sec}]$ 으로 증가했을 때의 응답특성을 보였고, 그림 6의 (b)는  $L = 0.3[\text{sec}]$ 이고 부하가 변화했을 때의 응답특성을 보였다. 그림 6에서 보는 바와 같이 적응 퍼지제어기가 오버슈트, 정정시간, 상승시간면에서 기존의 퍼지제어기보다 더 좋은 제어성능을 보였다. 이상의 결과를 요약하면 표 2와 같다.

표 2 2차 선형시스템에 대한 제어 성능 분석  
Table 2 Performance analysis for the second-order linear system

$L$	제어기종류	최대오버슈트 [%]	상승시간 [sec]	정정시간 [sec]
0.0	FLC	30.0	3.37	25
	AFLC	0.00	3	11
0.1	FLC	32.9	3.30	25
	AFLC	4.65	2.74	11
0.2	FLC	36.0	3.22	31
	AFLC	10.0	2.54	11
0.3	FLC	39	3.16	36
	AFLC	13	2.48	14



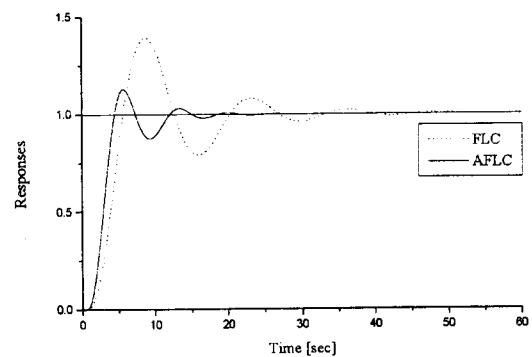
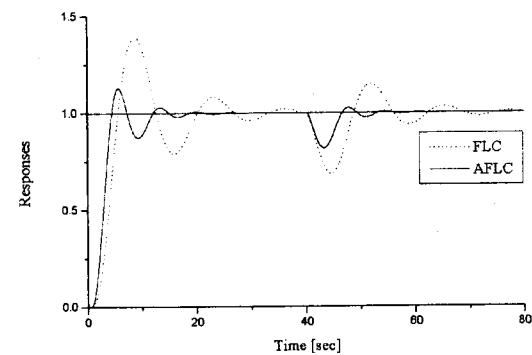
(a) 스텝응답



(b) 적응 퍼지제어기의 입·출력 이득의 변화

그림 5  $L = 0.2[\text{sec}]$ 일 때 2차 선형시스템에 대한 응답특성

Fig. 5 Responses for second-order linear system when  $L = 0.2[\text{sec}]$

(a) 스텝응답 ( $L = 0.3[\text{sec}]$ )

(b) 스텝응답 (부하 변화)

그림 6 2차 선형시스템에 대한 응답특성  
Fig. 6 Responses for second-order linear system

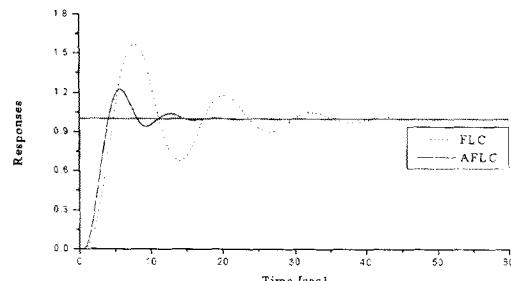
## 4.2 Marginally Stable System의 시뮬레이션 결과

실수형 유전알고리즘을 이용하여 적응 퍼지제어기의 설계에 사용된 조건은 제어시스템에 대한 시지연 상수가  $L = 0.0[\text{sec}]$ ,  $L = 0.2[\text{sec}]$ 을 이용하였다. 적응 퍼지제어기의 강인성 평가에 사용된 조건은 시지연 상수가  $L = 0.1[\text{sec}]$ ,  $L = 0.3[\text{sec}]$ 과  $L = 0.3[\text{sec}]$ 일 때 부하가 감소한 경우를 사용하였다.

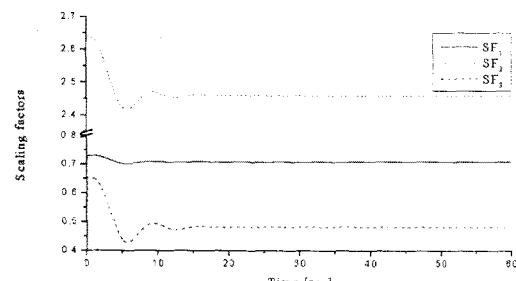
그림 7은 Marginally Stable 시스템에 대해 실수형 유전알고리즘으로 퍼지제어기의 입·출력 이득과 신경회로망의 가중치 튜닝에 사용된 시지연 상수가  $L = 0.2[\text{sec}]$ 일 때 스텝응답과 입·출력 이득의 변화를 보였다. 그림 7에서 보는 것처럼 적응 퍼지제어기가 오버슈트, 정정시간, 상승시간면에서 기존의 퍼지제어기보다 더 우수한 제어성능을 보였고, 적응 퍼지제어기의 입·출력 이득도 적응적으로 변경되고 있음을 알 수 있다. 그림 8의 (a)는 Marginally Stable 시스템에서 시지연 상수가  $L = 0.3[\text{sec}]$ 으로 증가했을 때의 응답특성을 보였고, 그림 8의 (b)는  $L = 0.1[\text{sec}]$ 이고, 부하가 변화했을 때의 응답특성을 보였다. 그림 8에서 보는 바와 같이 적응 퍼지제어기가 오버슈트, 정정시간, 상승시간면에서 기존의 퍼지제어기보다 더 좋은 제어성능을 보였다. 이상의 결과를 요약하면 표 3과 같다.

표 3 Marginally Stable 시스템에 대한 제어 성능 분석  
Table 3 Performance analysis for the marginally stable system

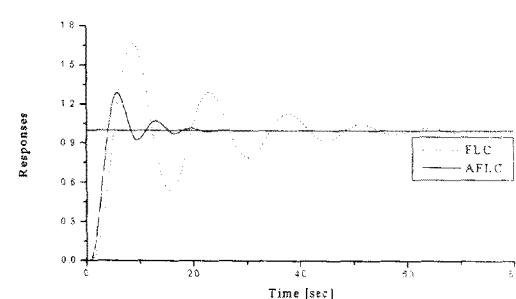
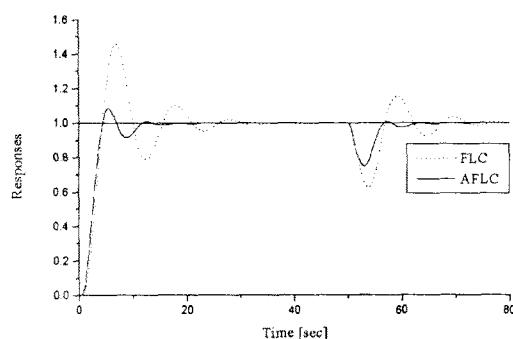
$L$	제어기 종류	최대오버슈트 [%]	상승시간 [sec]	정정시간 [sec]
0.0	FLC	31.8	2.34	16
	AFLC	0	3.07	11
0.1	FLC	45.9	2.53	25
	AFLC	8.63	2.62	11
0.2	FLC	56.9	2.65	39
	AFLC	22.7	2.33	14
0.3	FLC	66.4	2.84	59
	AFLC	29.0	2.25	17



(a) 스텝응답



(b) 적응 퍼지제어기의 입·출력 이득의 변화

그림 7  $L = 0.2[\text{sec}]$  일 때 marginally stable 시스템에 대한 응답특성Fig. 7 Responses for marginally stable system when  $L = 0.2[\text{sec}]$ (a) 스텝응답 ( $L = 0.3[\text{sec}]$ )

(b) 스텝응답 (부하 변화)

그림 8 Marginally stable 시스템에 대한 응답특성  
Fig. 8 Responses for marginally stable system

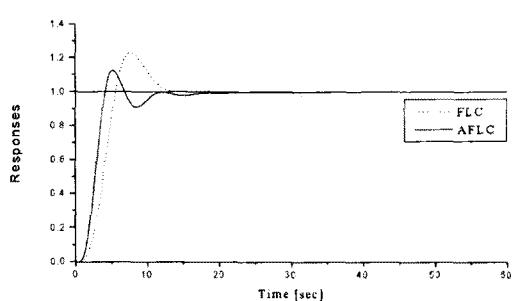
#### 4.3 Second-Order Nonlinear System의 시뮬레이션 결과

실수형 유전알고리즘을 이용하여 적응 퍼지제어기의 설계에 사용된 조건은 제어시스템에 대한 시지연 상수가  $L = 0.0[\text{sec}]$ ,  $L = 0.2[\text{sec}]$ ,  $L = 0.4[\text{sec}]$ 을 이용하였다. 강인성 평가에 사용된 조건은 시지연 상수가  $L = 0.3[\text{sec}]$ ,  $L = 0.5[\text{sec}]$ 와  $L = 0.3[\text{sec}]$ 일 때 부하가 감소한 경우를 사용하였다.

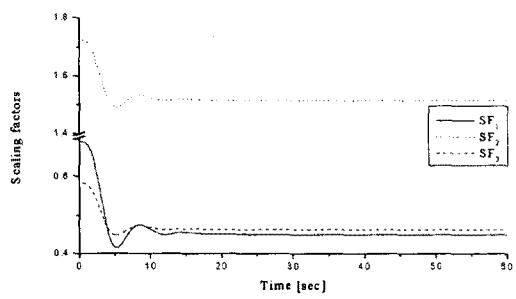
그림 9는 2차 비선형 시스템에 대해 실수형 유전알고리즘으로 퍼지제어기의 입·출력 이득과 신경회로망의 가중치 튜닝에 사용된 시지연 상수가  $L = 0.2[\text{sec}]$ 일 때 스텝응답과 입·출력 이득의 변화를 보였다. 그림 9에서 보는 것처럼 적응퍼지제어기가 오버슈트, 정정시간, 상승시간면에서 기존의 퍼지제어기보다 더 우수한 제어성능을 보였고, 적응 퍼지제어기의 입·출력 이득도 적응적으로 변경됨을 알 수 있다. 그림 10의 (a)는 2차 비선형 시스템에서 시지연 상수가  $L = 0.5[\text{sec}]$ 으로 증가했을 때의 응답특성을 보였고, 그림 10의 (b)는  $L = 0.3[\text{sec}]$ 이고, 부하가 변화했을 때의 응답특성을 보였다. 그림 10에서 보는 바와 같이 적응 퍼지제어기가 오버슈트, 정정시간, 상승시간면에서 기존의 퍼지제어기보다 더 좋은 제어성능을 보였다. 이상의 결과를 요약하면 표 4와 같다.

표 4 2차 비선형 시스템에 대한 제어 성능 분석  
Table 4 Performance analysis for the second-order nonlinear system

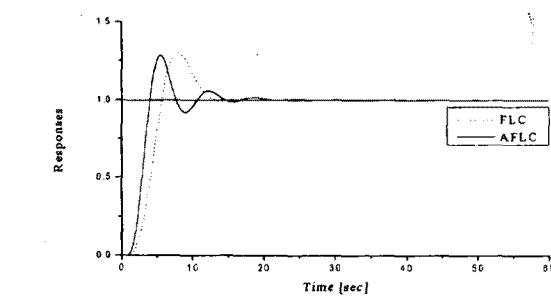
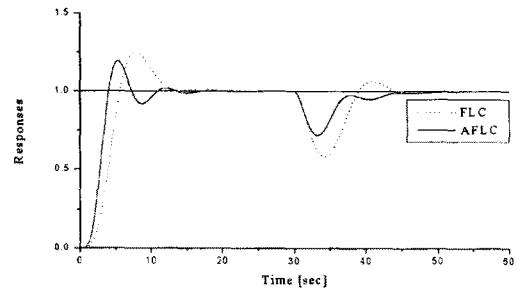
$L$	제어기종류	최대 오버슈트 [%]	상승시간 [sec]	정정시간 [sec]
0.2	FLC	23.1	3.11	12
	AFLC	12.7	2.29	10
0.3	FLC	25.5	3.06	12
	AFLC	19.5	2.15	10
0.5	FLC	30.0	2.97	12
	AFLC	28.5	2.01	13



(a) 스텝응답



(b) 적응 퍼지제어기의 입·출력 이득의 변화

그림 9  $L = 0.2[\text{sec}]$  일 때 2차 비선형 시스템에 대한 응답특성Fig. 9 Responses for second-order nonlinear system when  $L = 0.2[\text{sec}]$ (a) 스텝응답 ( $L = 0.5[\text{sec}]$ )

(b) 스텝응답 (부하 변화)

그림 10 2차 비선형 시스템에 대한 응답특성  
Fig. 10 Responses for second-order nonlinear

## 5. 결 론

본 논문에서 실수형 유전알고리즘과 전방향 신경회로망을 이용한 적용 퍼지제어기를 제안하였다. 적용 퍼지제어기는 실시간으로 매샘플링 시간마다 퍼지제어기의 입·출 이득이 적응적으로 변경된다. 제안한 방법의 유용성을 평가하기 위해 참고문헌[14]에 있는 제어시스템에 대한 기존의 퍼지제어기와 제어성능을 비교하였고, 그 결과를 요약하면 다음과 같다.

- 1) 2차 선형 시스템에 대해 컴퓨터 시뮬레이션 해본 결과, 제안한 적용 퍼지제어기가 기존의 퍼지제어기보다 약 26 ~ 30 [%] 정도 더 적게 오버슈트가 발생하였고, 상승시간면에서 약 0.37 ~ 0.8 [sec] 정도 더 빠르고, 정정시간면에서 약 14 ~ 22 [sec] 정도 더 빨리 정상상태에 수렴함을 알 수 있다.
- 2) Marginally stable 시스템에 대해서는 제안한 적용 퍼지제어기가 기존의 퍼지제어기보다 약 31 ~ 37 [%] 정도 더 적게 오버슈트가 발생하였고, 상승시간면에서 약 0.3 ~ 0.5 [sec] 정도 더 빠르고, 정정시간면에서 약 4 ~ 42 [sec] 정도 더 빨리 정상상태에 수렴함을 알 수 있다.
- 3) 2차 비선형 시스템에 대해서는 제안한 적용 퍼지제어기가 기존의 퍼지제어기보다 약 1.5 ~ 10 [%] 정도 더 적게 오버슈트가 발생하였고, 상승시간면에서 약 0.09 ~ 0.9 [sec] 정도 더 빠르고, 정정시간면에서 약 1 ~ 2 [sec] 정도 더 빨리 정상상태에 수렴함을 알 수 있다.

## (참 고 문 헌)

- [1] Emmanuel G. Collins, Jr. and Chun Fan, Richard Millett, "Automated PI Tuning for a Weigh Belt Feeder via unfalsified Control", Proceedings of the 38th IEEE Conference on Decision and Control, Vol. 1, pp. 785-790, 1999
- [2] Dragan Kukolj, Filip Kulic and Emil Levi, "Artificial Intelligence Based Gain Scheduling of PI Speed Controller in DC Motor Drives", Proceedings of the IEEE International Symposium on Industrial Electronics, Vol. 1, pp. 425-429, 1999
- [3] Fan JC, and Kobayashi T, "A SIMPLE ADAPTIVE PI CONTROLLER FOR LINEAR SYSTEMS WITH CONSTANT DISTURBANCES", IEEE Transactions on Automatic Control, Vol. 43, No.5, pp. 733-736, May, 1998
- [4] Lian KY and Lin CR, "Sliding-mode motion/force control of constrained robots", IEEE Transactions on Automatic Control, Vol. 43, No. 8, pp. 1101-1103, Aug. 1998
- [5] Benchaib A, Rachid A, Audrezet E and Tadjine M, "Real-time sliding-mode observer and control of an induction motor", IEEE Transactions on Industrial Electronics, Vol. 46, No. 1, pp. 128-138, 1999
- [6] Abraham Kandel, Gideon Langholz, "Fuzzy Control Systems", CRC Press, 1994
- [7] Siri Weerasoriya and M. A. El-Sharkawi, "Identification and Control of a dc motor using back-propagation neural networks", IEEE Trans. on Energy Conversion, Vol. 6, No. 4,

pp. 663 - 669, Dec., 1991

- [8] Y. F. Li and C. C. Lau, "Development of Fuzzy Algorithms for Servo Systems", IEEE International Conference on Robotics and Automation, pp. 65 - 72, April 24 - 29, 1988
- [9] Siri Weerasoriya and Mohamed A. El-Sharkawi, "Laboratory Implementation of a Neural Network Trajectory Controller for a DC motor", IEEE Trans. on Energy Conversion, Vol. 8, No. 1, pp. 107 - 113, March, 1993
- [10] K. Krishnakumar, David E. Goldberg, "Control System Optimization Using Genetic Algorithms", Journal of Guidance, Control, and Dynamics Vol. 15, No. 3, pp. 735 - 740, May - June, 1992
- [11] Vittorio Maniezzo, "Genetic Evaluation of the Topology and Weight Distribution of Neural Networks", IEEE Trans. on Neural Networks, Vol. 5, No. 1, pp. 39 - 53, Jan., 1994
- [12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley publishing Company, INC., 1989
- [13] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 1992
- [14] R. K. Mudi and N. R. Pal, "A Robust Self-Tuning Scheme for PI- and PD-Type Fuzzy Controllers", IEEE Transactions on Fuzzy Systems, Vol. 7, No. 1, pp. 2 -16, Feb. 1999[1]