

유전알고리즘과 진화프로그래밍을 이용한 퍼지제어기의 성능 향상에 관한 연구

A Study on the Performance Improvement of Fuzzy Controller Using Genetic Algorithm and Evolution Programming

이상부 · 임영도*

Sang Boo Lee and Young Do Lim*

한라전문대학 전산정보처리과
*동아대학교 공과대학 전자공학과

요 약

FLC(퍼지 제어기: Fuzzy Logic Controller)는 고전적인 제어기보다 외란(disturbance)에 강하고 초기치의 과도특성(overshoot)이 우수하다. 그리고 미지의 프로세스(process)나 복잡한 시스템의 수학적 모델링이 불가능한 경우에도 퍼지 추론에 의하여 적절한 제어량을 얻을 수 있다. 그러나 퍼지변수의 양자화 단계 크기에 의해 출력값이 항상 미세한 오차를 가지므로 목표치에 정확히 수렴하지 못한다 [1]. 이 미세한 오차를 제거하기 위한 여러 가지 방법이 [2~4] 있지만 본 논문에서는 FLC에 GA(유전알고리즘: Genetic Algorithm)와 EP(진화프로그래밍: Evolution programming)를 결합한 GA-FLC, EP-FLC Hybrid 제어기를 제안한다. 이 Hybrid 제어기의 출력 특성과 FLC의 출력 특성을 비교 분석하고, 이 Hybrid 제어기가 오차없이 목표치에 잘 수렴하는 것을 보이고자 한다. 또한 이 두 종류의 Hybrid 제어기 수렴 속도 성능도 비교한다.

ABSTRACT

FLC(Fuzzy Logic Controller) is stronger to the disturbance than a classical controller and its overshoot of the intialized value is excellent. In case an unknown process or the mathematical modeling of a complicated system is impossible, a fit control quantity can be acquired by the Fuzzy inference. But FLC can not converge correctly to the desirable value because the FLC's output value by the size of the quantization level of the Fuzzy variable always has a minor error. There are many ways to eliminate the minor error, but I will suggest GA-FLC and EP-FLC Hybrid controller which combines FLC with GA(Genetic Algorithm) and EP(Evolution Programming). In this paper, the output characteristics of this Hybrid controller will be compared and analyzed with those of FLC, it will be showed that this Hybrid controller converge correctly to the desirable value without any error, and the convergence speed performance of these two kinds of Hybrid controller also will be compared.

1. 서 론

종래의 통상적인 PID 제어는 상태에 따라 PID 계수를 동조시켰으나, 내·외부의 환경변화나 시스템의 비선형성을 추종하지는 못하게 된다. 이 경우 모델규범 적응제어 기법을 사용하여 실제 시스템에 추종시키고 있다. 이는 프로세스의 출력과 기존모델의 출력을 비교하는 것으로서 그 시스템에 알맞은 적응 알고리즘을 통해 제어기의 파라미터(paramater)를 조정하여야 한다. 이 기법은 전문가에 의해서만 가능하나, 대단히 복잡하며 많은 계산시간이 요하게 되므로 보다 정밀하고 빠른 응답을 요구하는 시스템에는 적용

한계가 따른다[5]. 이러한 문제의 해결방법으로서, 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 제어기에 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 정상편차는 적지만 부하변동에 의한 과도응답이 문제가 된다[5]. 이에 비해 FLC는 부하변동에 대해 과도특성은 양호하지만 목표치에서 약간의 오차가 존재하는

결점이 있다[1]. 이와 같은 단점을 상호 보완하기 위해 PID 제어와 FLC를 결합한 Hybrid 제어가 제안되고 있으나 실현을 위한 설계가 복잡해진다. FLC의 목표치에서 오차를 제거하기 위하여 본 논문에서는 FLC에 GA와 EP를 결합한 Hybrid 제어를 제안하고 이들의 성능을 비교 분석하고자 한다.

2. FLC, GA, EP

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

2.1 퍼지 제어 규칙과 추론

FLC의 제어 규칙(Rule)은 전문가(Expert)의 지식이나 경험에 의해서 얻어진다[6]. 규칙 "IF-THEN" 형식으로 구성되어 있기 때문에 전문가의 경험과 지식을 자연언어로 사용할 수 있게 한다. 제어규칙의 예는 다음과 같다.

- R₁: IF e₁ is PB and ce₁ is PS THEN co₁ is PB.
- R₂: IF e₂ is NM and ce₂ is PS THEN co₂ is NM.
- R_n: IF e_n is ZO and ce_n is ZO THEN co_n is ZO.

여기서 e₁, e₂, e_n, ce₁, ce₂, ce_n: 제어기 입력, PB, PS, ZO: 언어적인 값, co₁, co₂, co_n: 퍼지 제어기 출력

추론이란 어떤 주어진 사실이나 관계로부터 새로운 사실이나 관계를 유추하는 과정이다[3]. 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 퍼지 집합 co': 추론된 제어기 출력값 μ: 소속함수값

본 논문에서는 e와 ce에 대해서 어떤 측정값 e'와 ce'가 입력되면 그 값에 대한 퍼지 집합의 소속함수를 모두 표현하지 않고 식 (3)과 같이 그때 그 부분값만을 취하는 결정론적인 입력형태 e^o와 ce^o를 사용한다[3].

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

그러면 식 (2)는 식 (3)에 의해서 식 (4)와 같이 된다.

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

규칙이 n개 있고 이 규칙으로부터 추론하여 결과값을 구하면 식 (5)와 같다. 즉 n개의 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 프로세스의 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, Σ: 값의 합, μ: 소속함수값, μ_s: 소속함수값을 갖는 대집합, μ_v: 해당 대집합의 소속 함수값

2.2 GA, EP

GA와 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, 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[7-9]. GA, EP의 성능은 교배율, 돌연변이율, 해집단의 크기, 각 파라미터의 값, 목적함수 설정과 적합도 값의 scaling 등에 좌우된다[7, 8]. 특히 GA에서는 교배율이 큰 영향을 미치고 EP에서는 돌연변이율이 큰 영향을 미친다. GA에서 전역적인 최적의 해를 구하기 위한 순서는 다음과 같다[7, 8].

1) 부호화 및 초기화: 생물의 유전정보인 염색체(chromosome)와 같이 GA에서는 염색체에 해당하는 파라미터를 부호화 한다. 보통의 부호화 방법은 유한 길이의 이진 염색체로 부호화하며 랜덤하게 N개의 문자열을 생성하여 초기 해집단(population)을 구성한다.

2) 평가: 각 문자열을 디코딩하여 목적함수에 대한

적합도(fitness)를 계산한다.

- 3) 최적해를 구하였으면 탐색 과정을 중단한다.
- 4) 재생산: 자연 생태계의 적자생존 원리를 이용하여 높은 적합도를 가진 문자열에 대하여 다음 세대로 복제될 확률을 높게 한다.
- 5) 교배와 돌연변이: 문자열을 쌍으로 구성하여 문자열 안에 있는 유전자 정보를 서로 교환하고 유전자의 돌연변이 형태로 새로운 유전 정보를 갖는 문자열을 만든다.
- 6) 새로 생성된 자손의 문자열에 대하여 적합도를 평가하고 최적해를 구할때까지 3) 이하의 과정을 반복한다.

EP에서 전역적인 최적의 해를 구하기 위한 순서는 다음과 같다[10].

- 1) 부호화 및 초기화: 염색체에 해당하는 파라미터를 랜덤한 실수값의 문자열로 코드화 한 후 N개의 문자열로 구성된 초기 집단을 생성하여 초기 부모 집단을 형성한다.
- 2) 재생산: 각 문자열들을 디코딩(decoding)하여 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, 각 문자열에서 표현되는 bit (or chromosome)을 돌연변이 확률에 의해 각 bit를 변화시킨다. 이와같은 동작은 과거에 잃어 버렸던 정보를 되찾을 수도 있으며, 탐색의 방향이 지역 최솟치에 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 6)에서 만들어진 순위에서 상위 반을 선택하여 부모 집단으로 교체한다. 목적함수가 최대가 될 때까지 단계 2)로 진행한다.

GA에서 유전자 문자열은 2진수로 코드화하고 교배를 주연산자로 하며 돌연변이는 보조적으로 사용한다. EP에서는 유전자 문자열을 실수값으로 코드화하며 교배없이 돌연변이만을 사용하거나, 돌연변이를 주연산자로 사용하고 교배는 보조적으로 사용할 수도 있다.

3. GA를 결합한 FLC

FLC에서 어려운것은 지식 기반을 토대로 한 퍼지 소속 함수의 형태 또는 퍼지 규칙을 설계하는 작업이다. 본 논문에서는 지식 기반의 토대를 GA를 이용해서 소속함수의 형태를 최적화 한다. FLC에 GA를 결합하여 최적의 퍼지소속 함수를 구하는 GA-FLC 구조는 그림 1과 같다.

이 제어기는 먼저 GA를 실행시켜 최적화된 퍼지소속 함수값을 구한 후 FLC를 동작시킨다. 유전자 구조는 그림 2의 (a)와 같이 설정하였다. 여기서 각 유전자는 m개의 염색체를 갖는다. 오차 (e)와 오차변화 (ce)의 모든 퍼지 소속함수에 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.

- 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

그리고 각 염색체는 10비트 2진코드로 구성한다. 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, 우폭값은 PB의 중심에서 PB의 좌폭값을 뺀 것으로 처리하였다. 시스템의 안정화를 위하여 PM의 좌폭은 PM의 중심값보다 크지 않도록 하였다. 초기 세대 유전자는 안정화된 세대진화를 위하여 각 퍼지수에 적절한 영역을 설정하여 다음 세대를 생성하도록 하였다. 세대를 진화시키는 요소는 앞의 염

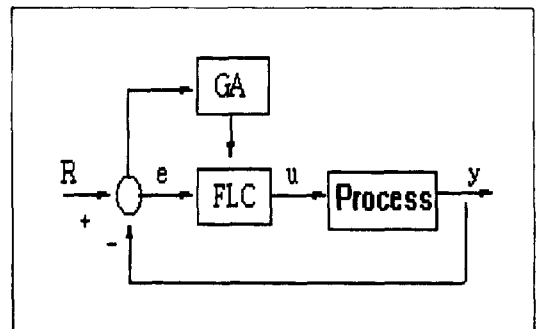
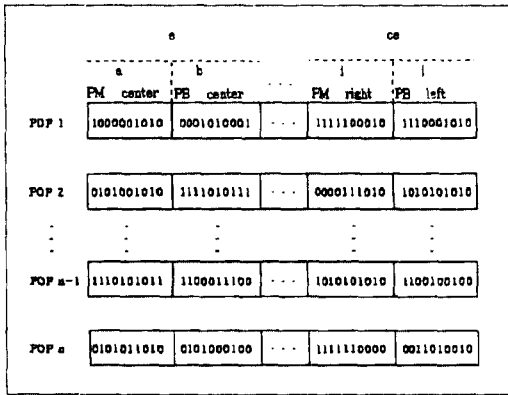
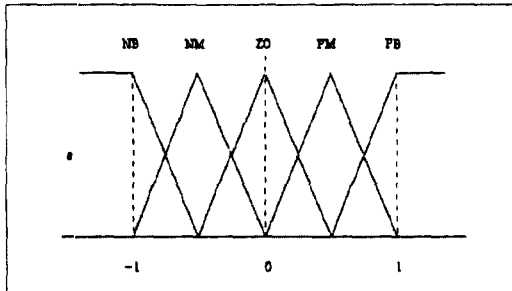


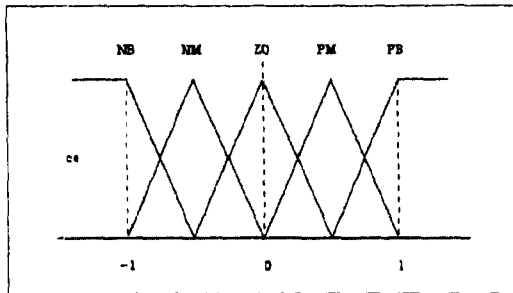
그림 1. GA-FLC의 구조
Fig. 1. Structure of GA-FLC



(a) 2진 형태의 유전자 구조
(a) Gene structure of binary type



(b) e의 퍼지 소속함수 형태
(b) Fuzzy membership type of e



(c) ce의 퍼지 소속함수
(c) Fuzzy membership type of ce

그림 2. 2진 형태의 유전자 구조와 FLC의 삼각소속함수 형태

Fig. 2. Gene structure and FLC tringle membership type of binary type

색채 구조와 다음의 평가함수로 이루어진다. 이 평가함수는 시스템에 있어서 적합한 유전자인가를 평가할 수 있는 적합도 함수이다. 여기서 사용한 평가함수는 오차와 오차변화를 최소화 하는 것을 목적으로 한다. 따라서 식 (7)과 같이 오차와 오차변화의 제곱 누

적을 평가함수의 항으로 사용하였다.

$$F(pop_i) = \frac{K_1}{1 + \sum_k e_k^2} + \frac{K_2}{1 + \sum_k ce_k^2} \quad (7)$$

여기서, $K_1 + K_2 = 1$ 로 하여 K_1, K_2 를 적절하게 설정한다.

4. EP를 결합한 FLC

FLC에 EP를 결합하여 최적화된 퍼지소속함수를 구하는 EP-FLC 구조는 그림 3과 같다. 유전자 구조와 삼각소속함수 형태는 그림 2와 같이 하였다.

본 논문에서는 그림 2(a)에 해당하는 2진 형태의 유전자 구조를 그림 4와 같이 각각의 소속함수에 해당하는 스트링들은 각 멤버십 함수에 3개씩의 실수 스트링을 배당하였다. 이렇게 함으로써 2진 bit 에 대한 디코딩 작업을 생략할 수 있다. 이렇게 하여 얻어진 각 염색체는 일정한 범위 내에서의 정보를 무작위로

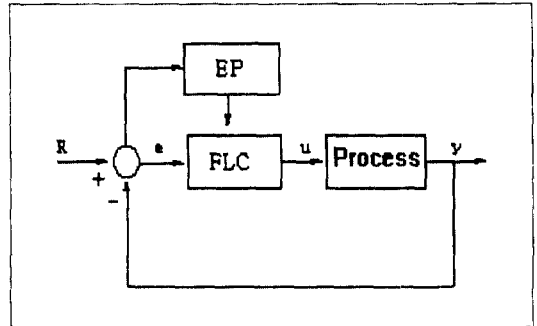


그림 3. EP-FLC의 구조
Fig. 3. Structure of EP-FLC

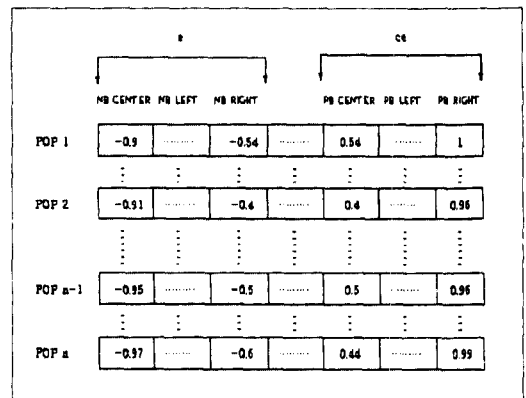


그림 4. 10진 형태의 유전자 구조
Fig. 4. Gene structure of decimal type

얻어오게 하였다. 이렇게 할 경우 소속함수 형태의 변화폭이 너무 크게 변화되므로, 초기 세대 유전자는 무작위 설정보다는 안정된 세대 진화를 위하여 각 퍼지수에 적절한 영역을 설정하여 다음 세대를 생성하도록 하였다. 세대를 진화시키는 요소는 앞의 염색체 구조와 다음의 평가함수에 교배율, 돌연변이율에 의해 크게 좌우된다.

그림 3의 제어기 구조에서 사용한 평가함수는 식 (7)를 이용하였다.

그림 3의 구조에서 먼저 EP를 이용하여 최적화된 FLC 소속함수값을 얻은후 FLC를 동작시킨다.

추론 방법은 MAX-MIN 방법을 이용하고 무게 중심법으로 비퍼지화 값을 구했다.

5. 실험 및 결과

모의(simulation) 실험에서 사용한 가상 2차계의 프로세스 차분 방정식은 식 (8)과 같다.

$$y(k) = 1.3679 * Y_1 - 0.3679 * Y_2 + 0.3679 * U_1 + 0.2642 * U_2 \quad (8)$$

여기서 Y_1 : 현재의 프로세스 출력, Y_2 : 현재보다 한 샘플링 시점 이전의 프로세스 출력, U_1 : 현재의 제어기 출력, U_2 : 현재보다 한 샘플링 시점 이전의 제어기 출력

설정치 R은 100, 샘플수 K는 500으로 하였다. 그림 5는 FLC의 출력이 목표치에 잘 수렴하도록 각각의 이득값을 설정한 것이다. 이때의 각각 이득값은 $Ge=0.605$, $Gce=1$, $Go=0.348$ 이다. 여기서 Ge 와 Gce 는 각각의 입력 이득이 곱해진 후의 오차와 오차 변

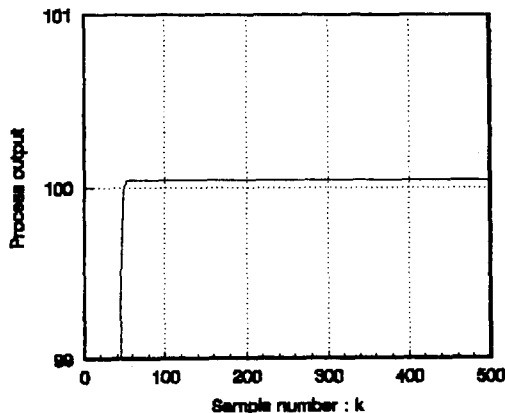


그림 5. $Ge=0.605$, $Gce=1$, $Go=0.348$ 의 FLC 출력 특성
Fig. 5. FLC output characteristics of $Ge=0.605$, $Gce=1$, $Go=0.348$

화량 값이고 Go 는 출력 이득을 곱한후의 비퍼지화 값이다. 아무리 이득값을 조정해도 목표치와 미세한 오차를 가지게 된다. 여기서는 목표치에 거의 가깝게 수렴하지만 초기의 출력 응답속도가 목표치에 너무 늦게 수렴하는 단점이 생긴다. 그림 6은 FLC 출력이 목표치에 빨리 수렴하도록 각각의 이득을 설정한 출력 특성 그래프이다. 이때 각각의 이득값은 $Ge=0.01$, $Gce=9$, $Go=0.128$ 이다. 이 그림의 출력 응답이 그림 5와 비교해서 목표치에 빨리 도달 하지만 출력 응답값이 목표치와 오차가 크게된다. 그림 7은 GA-FLC에서 71세대를 진화 시킨후의 출력 응답을 나타낸 것이다. 이때 사용된 파라미터값 유전율은 0.7, 돌연변이율은 0.01, 한세대의 유전자는 30개로 하였다. 그림 8은 EP-FLC에서 14세대를 진화 시킨후의 출력 응답 그래프이다. 여기서 사용한 유전율은 0.004, 돌연변이

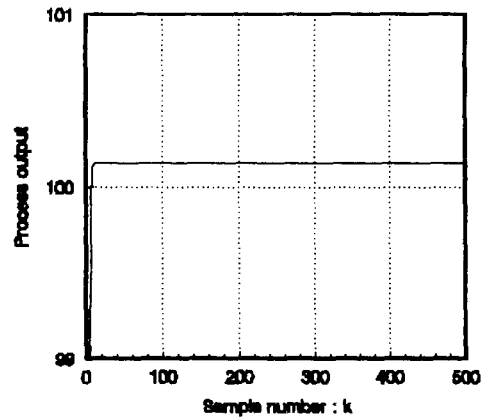


그림 6. $Ge=0.01$, $Gce=9$, $Go=0.128$ 의 FLC 출력 특성
Fig. 6. FLC output characteristics of $Ge=0.01$, $Gce=9$, $Go=0.128$

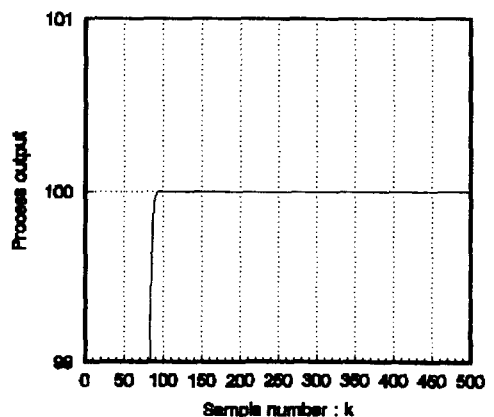


그림 7. GA-FLC의 출력 특성
Fig. 7. Output characteristics of GA-FLC

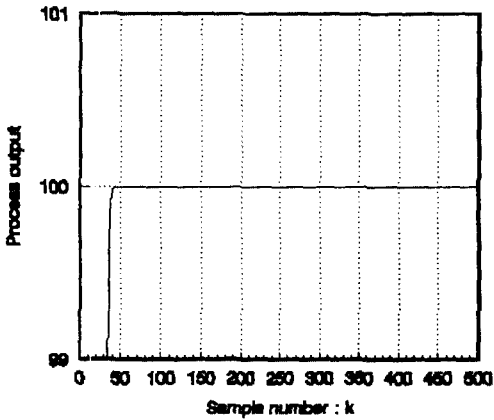


그림 8. EP-FLC의 출력 특성
Fig. 8. Output characteristics of EP-FLC

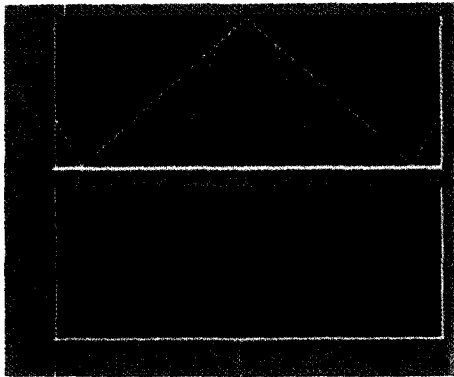


그림 9. 그림 8의 1세대 소속함수 형태
Fig. 9. 1 generation membership function type of fig. 8

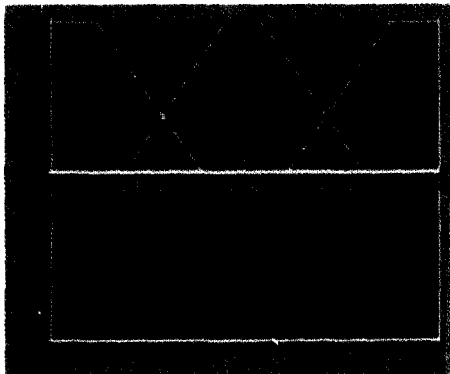


그림 10. 14 세대 진화후 그림 8의 소속함수 형태
Fig. 10. Membership function type of fig. 8 after 14 generation evolution

율은 0.8, 한세대의 유전자는 30개이다. 그림 7과 그

림 8의 출력 응답은 오차없이 목표치에 정확히 수렴하며 EP-FLC가 GA-FLC보다 진화와 수렴 속도가 훨씬 빠르다는 것을 알 수 있다. 그림 8의 EP-FLC 제어기에 대한 1세대 소속함수의 모양은 그림 9이고 14세대 진화후 7개체의 소속함수 모양은 그림 10이다.

6. 결 론

본 논문에서는 FLC의 성능을 향상시키기 위하여 FLC에 GA와 EP를 결합한 Hybrid 제어기를 제안하였다. 실험 결과 FLC는 목표치에서 미세한 오차를 가지고 있었지만 이들 Hybrid 제어기는 목표치에 오차 없이 정확히 수렴하였다. 또한, EP-FLC와 GA-FLC를 비교하면 EP-FLC의 출력 응답이 빠른 것을 실험적으로 알 수 있었다. 그러므로 EP가 GA보다 더 우수하다. GA와 EP의 문제점으로는 해집합 선택 방법에서 해집합이 클 경우 최적화 하는데 많은 시간이 소모된다. 또한 적당한 교배율과 돌연변이율을 찾는 데도 많은 시간이 소모되므로 짧은 시간 내에 이들의 파라미터 값을 찾는 방법도 과제이다. GA나 EP는 off-line 구조이므로 제어 system에 응용할 경우 학습의 능력을 가지고 있는 신경망을 이용하여 보다 효율적인 on-line 제어 방식을 찾는 것도 앞으로의 연구 과제이다. 최근에는 GA와 EP가 다양한 방법들로 발표되므로 서로의 특징을 세세한 차이로 구별하기 힘들다. 단지 실수 코딩, 2진 코딩, 재생산, 교배, 돌연변이를 문제에 알맞게 선택하는 것이 중요하다.

참고문헌

- [1] 이영도, 김태완, 서일홍, "록업 테이블을 이용한 자동학습 퍼지제어기의 설계에 관한연구," 전자공학회 논문지, 제 29권, B편, 제 9호, pp. 723-734, 1992.
- [2] 이상부, 김현정, 임영도, "퍼지와 비례제어기를 결합한 Hybrid 제어기의 성능향상에 관한연구," 인공지능 신경망 및 퍼지시스템 추계종합학술대회 논문집, pp. 183-188, 1993.
- [3] 이상부, 김형수, "퍼지제어기와 신경망제어기의 응답특성에 관한연구," 정보처리학회 논문지, Vol. 3, No. 6, pp. 1473-1482, 11, 1996.
- [4] James J. Buckley, Yoichi Hayashi, "Fuzzy neural networks: A survey," *Fuzzy Sets and Systems*, No. 66, pp. 1-11, 1994.
- [5] Katsuhiko Ogata, *Discrete-time Control System*, Prentice-Hall, INC., pp. 198-230, 1987.
- [6] 李勇, 米澤洋, "DC 모-타 사-보系の ファジイ制御," *日本ファジイ 學會誌*, Vol. 2, No. 4, pp. 146-151,

1990.

- [7] David E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, INC., PP. 27-82, 1989.
- [8] 임영도, 이상부 공저, 퍼지 신경망 유전진화, 도서출판 영과일, PP. 193-215, 1996.

- [9] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.
- [10] Zbigniew Michalewicz, *Genetic Algorithms+Data Structures=Evolution Programs*, Springer-Verlag, INC., pp. 65-69, 1992.



이상부 (Sang Boo Lee) 정회원

1956년 3월 11일생
1983년: 동아대학교 공과대학 전자공학과 졸업 (공학사)
1985년: 동아대학교 대학원 전자공학과 졸업 (공학석사)
1997년: 동아대학교 대학원 전자공학과 졸업 (공학박사)

1994년~현재: 한라전문대학 전산정보처리과 교수
주관심분야: 퍼지 및 신경망 제어, 유전·진화 알고리즘, 카오스 이론, 적응제어, 컴퓨터 구조 및 마이크로프로세서 응용



임영도 (Young Do Lim) 종신회원

1949년 9월 20일생
1973년: 동아대학교 전자공학과 졸업 (공학사)
1978년: 동아대학교 대학원 전자공학과 졸업 (공학석사)
1987년: 동아대학교 대학원 전자공학과 졸업 (공학박사)

1988년~현재: 동아대학교 전자공학과 교수
주관심분야: 신경망제어, 퍼지제어, 전력전자, 고주파유도 가열