

論文98-35C-12-10

학습과 진화의 Lamarckian 상호 적응에 의한 뉴로-퍼지 제어기의 최적 설계

(An Optimal Design of Neuro-Fuzzy Logic Controller Using Lamarckian Co-adaptation of Learning and Evolution)

金大鎭*, 李한별*, 姜大星*

(Daijin Kim, Hanpyul Lee, and Dae Sung Kang)

요 약

본 논문은 특정 응용에 적합한 퍼지 제어기의 최적 설계 파라미터 (퍼지 규칙과 소속 함수)를 찾는 데 역전과 학습 과정과 유전 알고리즘을 결합한 Lamarckian 상호적응 기법을 이용한 뉴로-퍼지 제어기의 새로운 설계 방법을 제안한다. 설계 파라미터들은 진화에 의한 전역적 탐색을 통해 높은 포함값과 유용한 퍼지 규칙들을 갖는 규칙 베이스와 작은 근사화 오차와 좋은 제어 성능을 갖는 소속 함수들을 얻도록 제어기간 파라미터 조절을 수행하며, 학습에 의한 국부적 탐색을 통해 각 퍼지 제어기가 원하는 제어 결과를 나타내도록 제어기내 파라미터 조절을 수행한다. 제안한 상호적응 설계 방법은 유전 알고리즘의 모든 세대에서 역전과 학습이 이루어지므로 보다 좋은 근사화 능력을 나타내고, 사용한 무게 중심 비퍼지화기가 정확한 비퍼지화값을 계산하므로 보다 좋은 제어 성능을 가지며, 퍼지 규칙 베이스와 소속 함수들의 최적화 탐색 과정이 입출력 공간의 같은 퍼지 분할 상에서 통합된 적응 함수에 의하여 동시에 수행되므로 탐색을 위한 작업 공간이 아주 작아지는 장점이 있다. 시뮬레이션 결과는 Lamarckian 상호 적응에 의해 얻어진 FLC가 퍼지 규칙수, 근사화 능력, 제어 성능 등 모든 면에서 다른 방법에 의해 얻어진 FLC보다 가장 우수함을 보여준다.

Abstract

This paper proposes a new design method of neuro-FLC by the Lamarckian co-adaptation scheme that incorporates the backpropagation learning into the GA evolution in an attempt to find optimal design parameters (fuzzy rule base and membership functions) of application-specific FLC. The design parameters are determined by evolution and learning in a way that the evolution performs the global search and makes inter-FLC parameter adjustments in order to obtain both the optimal rule base having high covering value and small number of useful fuzzy rules and the optimal membership functions having small approximation error and good control performance while the learning performs the local search and makes intra-FLC parameter adjustments by interacting each FLC with its environment. The proposed co-adaptive design method produces better approximation ability because it includes the backpropagation learning in every generation of GA evolution, shows better control performance because the used COG defuzzifier computes the crisp value accurately, and requires small workspace because the optimization procedure of fuzzy rule base and membership functions is performed concurrently by an integrated fitness function on the same fuzzy partition. Simulation results show that the Lamarckian co-adapted FLC produces the most superior one among the differently generated FLCs in all aspects such as the number of fuzzy rules, the approximation ability, and the control performance.

* 正會員, 東亞大學校 電氣電子컴퓨터工學部
(School of Electrical, Electronic, and Computer Eng.,
DongA University)

※ 본 논문은 과학 재단의 1998년도 핵심전문 연구 사업의 지원(981-0920-104-2)에 의해 수행되었음.
接受日字:1998年7月20日, 수정완료일:1998年11月23日

I. 서론

퍼지 논리 제어기는 가전 및 산업 분야의 공정 제어에 폭넓게 응용되고 있다. 특히 시스템의 특성이 복잡하여 기존의 정량적인 방법으로는 해석할 수 없거나, 얻어지는 정보가 정성적이며, 부정확하고 불확실한 경우에 있어서 기존의 제어기보다 우수한 제어결과를 나타낸다^[1].

퍼지 제어기는 크게 4가지 구성 요소 - 퍼지화부, 추론 엔진부, 퍼지 규칙 베이스부, 그리고 비퍼지화부로 나뉜다. 각 부분의 동작 설명은 다음과 같다. 퍼지화부에서는 입력 변수의 값을 측정하고, 입력 변수의 영역을 전체 집합범위에 맞게끔 크기 변환한 뒤 입력 값을 적절한 언어적인 값으로 변환시키고, 추론부에서는 퍼지 관계와 퍼지 논리의 추론 규칙을 사용하여 퍼지 제어 출력을 결정하며, 퍼지 규칙 베이스부에서는 퍼지 논리 제어에서의 퍼지 자료를 조작하고 언어적 제어 규칙을 정의하는데 필요한 사항들을 정의한 데이터 베이스와 제어 전문가가 수행하는 일련의 제어 과정을 언어적 제어 규칙들로 나타낸 제어 규칙부로 구성되고, 그리고 비퍼지화부에서는 퍼지 출력값을 실제 제어 입력에 맞게끔 변환시켜 실제 제어 입력으로 사용할 수 있는 확정된 출력값으로 변환시켜 준다.

FLC는 프로세스를 처리하는 IF-THEN 규칙과 퍼지 언어항을 정확한 산술치로 변환하는 소속함수를 사용한다. 따라서, FLC의 설계 문제는 퍼지 규칙 베이스와 입출력 변수가 갖는 퍼지 언어항의 소속 함수를 결정할 것이 요구된다. FLC 설계자는 전문가의 지식과 경험이 반영되어지도록 퍼지 규칙 베이스를 선택하고, 퍼지 언어항에 대한 전문가의 이해가 반영되도록 소속 함수를 결정한다. 그러나, 이러한 결정은 일반적으로 전문가의 주관적 판단이나 조작자의 행위적 지식에 의하여 이루어진다. 이러한 ad-hoc 방법으로 얻어진 FLC의 제어 성능은 항상 최적은 아니다. 본 연구의 동기는 전문가의 도움 없이 복잡하고 비선형적인 문제를 해결할 수 있는 최적 FLC의 설계 방법을 찾는 방안을 얻는 것이다.

FLC를 설계하는 한 방법은 신경망 구조와 학습 능력을 이용한 뉴로-퍼지 방법^[2,3]이다. 이 방법은 1) 산술적 데이터만 이용할 수 있고, 2) 국부해에 빠지기 쉽기 때문에 항상 최적 시스템 성능을 기대할 수 없으며, 3) 가중치의 내부적 표현 성질 때문에 생성된 퍼

지 규칙을 해석하기가 어려운 단점이 있다. FLC의 또 다른 설계 방법은 최적 퍼지 규칙 베이스와 소속 함수를 결정하는데 유전 알고리즘을 이용하는 진화적 접근 방법^[4,5]이다. 이 방법은 퍼지 규칙 베이스를 생성할 때 산술 데이터와 언어 퍼지 규칙들을 모두 사용할 수 있고, 유전 알고리즘이 갖는 내재적 병렬성 때문에 최적해를 찾을 가능성이 더 큰 장점이 있다. 그러나, 이 방법은 만족할 수준의 해에 도달하는데 걸리는 시간이 많이 소요되고, 요구하는 작업 공간이 큰 단점이 있다.

최근에, 생물학적 개념에 기초하여 진화와 학습을 결합하는 방안이 점차 주목을 끌고 있다^[6,7]. 진화는 해집단 수준에서 유전자형을 변화시키는 반면, 학습은 개체 수준에서 개체의 행동 양식을 변화시킨다. 진화에 의한 변화는 개체의 선택적인 재생산성 및 변이성을 유지하기 위한 유전 연산(재결합 and/or 변이)을 수행함으로써 얻어진다. 진화에 의한 변화는 특정한 세대에서 일어난 변화가 그 이전 세대에서 생긴 다른 변화들에 중첩되어 세대가 지날수록 누적된다. 학습에 의한 변화는 일생동안의 특정 환경과의 상호 작용과 경험을 통해 그 내부 구조에 환경의 영향을 반영함으로써 얻어진다. 그리고, 학습에 의한 변화는 한 개체의 일생 중 특정 시점에서 일어난 변화가 그 이전 시점에 일어난 변화에 의해 영향을 받아 개체적으로 누적된다.

진화와 학습이 두 가지 다른 종류의 수준(각각 해집단과 개체수준)에서 일어나는 서로 다른 종류의 변화이지만, 서로 영향을 주고받는다. 진화가 학습에 미치는 영향은 진화적 변화가 유전자형(genotype)에 그 흔적을 남기고 한 개체의 계승은 표현형(phenotype) 특성을 부분적으로 결정한다는 사실로부터 쉽게 이해할 수 있고, 그러한 영향은 개체가 앞으로 어떻게 행동할 것인가와 무엇을 학습할 것인가를 결정짓는다. 진화가 학습에 미치는 영향은 유전 알고리즘을 신경망의 해집단에 적용하는 많은 실험에서 발견할 수 있다. 진화는 신경망이 특정 과제를 학습하려는 경향을 가지도록 선택한다. 특정 과제를 학습하려는 경향은 여러 다양한 방법으로 신경망에 결합되어질 수 있다. 예를 들어, 진화는 좀 더 나은 학습이 이루어질 수 있도록 초기 가중치 벡터^[8]나 망 구조^[9]를 선택할 수 있다.

학습이 진화에 어떻게 영향을 미치는가에 대한 하나의 가설은 한 생명체가 일생 동안 얻은 표현형의 특성이 유전될 수 있는 유전자형에 복사되어 한 생명체의

자손에 직접 전달된다는 Lamarckian 가설^[10]이다. D. Ackley와 M. Littman^[11]은 Lamarckian 가설이 컴퓨터 상에 쉽게 구현될 수 있고 동시에 그것이 갖는 빠른 수렴 특성으로 인해 최적화 문제에 훨씬 더 효율적이라는 것을 보였다. 현대 생물학의 관점을 따르면, Lamarckian 가설은 더 이상 진실이 아니다. 그러나, 컴퓨터 과학자의 관점에서 보면 Lamarckian 모델은 인공적인 모의 시스템 상에서는 구현 가능하다. 따라서, 본 논문에서는 최적 FLC 설계를 위하여 학습을 진화에 결합시킨 Lamarckian 상호 적응을 사용할 것을 제안한다. 제안한 상호 적응 방법은 유전 알고리즘으로 많은 FLC들을 진화시키고 역전파 학습 규칙을 이용하여 개개의 FLC를 학습하여 설계 파라미터들을 조정한다.

본 논문은 다음과 같이 구성된다. II장에는 connectionist 구조와 정확한 COG 비퍼지화기를 갖는 뉴로-퍼지 제어기가 기술되고, 제안한 퍼지 제어기에 적당한 학습 알고리즘을 설명한다. III장에서는 유전 알고리즘에 기초한 진화에 의한 뉴로-FLC의 설계 방법이 설명된다. IV장에서는 학습을 진화에 포함시킨 새로운 뉴로-FLC의 설계 방법이 설명된다. V장에서는 시뮬레이션을 통해 제안한 뉴로-FLC가 보다 빠른 수렴 속도와 보다 좋은 제어 성능을 나타냄을 보인다. 마지막으로, 결론이 뒤따른다.

II. 제안한 뉴로-퍼지 제어기와 학습

1. 제안한 뉴로-FLC

기존의 FLC는 여러 가지 방법으로 재구성될 수 있다. 본 논문에서는 5층의 전방향 신경망 구조^[2]로 모형화된 Lin과 Lee의 connectionist FLC를 채택하였다. 1층에 있는 노드는 입력 언어 변수를 기술하는 입력 노드(언어 노드)이다. 5층은 출력층이다. 2층과 4층에 있는 노드는 각기 언어 변수의 항들을 기술하기 위한 소속 함수들의 역할을 하는 항 노드(term nodes)이다. 3층에 있는 각 노드는 퍼지 규칙을 기술하는 규칙 노드으로써 퍼지 규칙 베이스를 형성한다. 3층에 있는 링크들은 퍼지 규칙의 전단부를 정의하고 4층에 있는 링크들은 퍼지 규칙의 후단부를 정의한다. 2층과 5층의 링크들은 각각 입출력 변수에서 언어 노드와 대응하는 항 노드사이를 완전 연결 형태로 연결되어 있다. 그림 1은 일반적인 뉴로-FLC의 구조를 설명한다.

제안한 뉴로-FLC의 다섯 층들의 각각에 있는 함수 표기는 [2]에서 잘 요약되어 있다.

본 연구에서는 기존의 COG 비퍼지화기가 문제점을 해결하고, COG 비퍼지화기의 최적 설계는 다른 기능 유닛보다 제어 성능을 더욱 향상시킬 수 있기 때문에 FLC가 갖는 여러 기능 유닛중 COG 비퍼지화기를 변경하고자 한다. 기존의 COG 비퍼지화기가 잘 못 사용되고 있는 문제점은 다른 논문 [12]에 잘 기술되어 있다. 이러한 잘못을 바로 잡기 위하여 본 논문은 비퍼지화값을 다음과 같이 구하는 새로운 COG 비퍼지화 방법을 제안한다.

$$y_c = \frac{\sum_{i=1}^n A_Y(y_i) \cdot y_i}{\sum_{i=1}^n A_Y(y_i)} \quad (1)$$

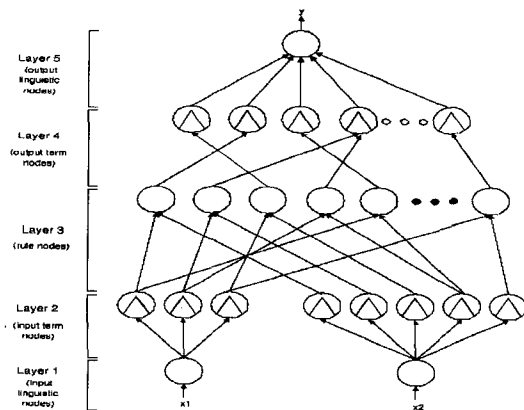


그림 1. 뉴로-퍼지 제어기의 구조
Fig. 1. Structure of neuro-FLC.

여기서 $A_Y(y)$ 는 i 번째 추론된 소속함수의 절단된 소속 함수값의 아래 영역, n 은 퍼지항의 개수, y_i 는 i 번째 퍼지항의 단일 지지값을 나타낸다.

일반화된 소속 함수 $\mu_Y^r(y)$ 의 경우에 면적 $A_Y(y_i)$ 의 정확한 식의 상세한 유도는 [12]에서 나타나 있다. 여기서 r 은 소속 함수를 수축($r > 1$)시키거나 팽창($r < 1$)시키는 모양 상수이다. 유도 결과는 $\mu_Y^r(y)$ 의 절단된 소속 함수 아래의 면적 $A_Y(y_i)$ 이 다음과 같이 계산됨을 보여준다.

$$\begin{aligned} A_Y(y_i) &= \mu_Y(y_i) \cdot s_i - \frac{r}{r+1} \mu_Y^{\frac{r+1}{r}}(y_i) \cdot s_i \\ &= \mu_Y(y_i) \cdot s_i \cdot \left(1 - \frac{r}{r+1} \mu_Y^{\frac{1}{r}}(y_i)\right) \\ &= \mu_Y^{\frac{r+1}{r}}(y_i) \cdot s_i \end{aligned} \quad (2)$$

여기서, r , $\mu_Y(y_i)$ 와 s_i 는 각각 i 번째 퍼지항의 소속 함수의 모양 상수, 소속 함수, 그리고 i 번째 소속 함수의 폭이다. $\mu_Y^e(y)$ 는 $\mu_Y(y)\left(1 - \frac{r}{r+1} \cdot \mu_Y^{\frac{1}{r}}(y_i)\right)$ 로 계산되는 유효 소속 함수값이다. 따라서 비퍼지화값 y_c 는 다음과 같이 계산된다.

$$y_c = \frac{\sum_{i=0}^n A_Y(y_i) \cdot y_i}{\sum_{i=0}^n A_Y(y_i)} \quad (3)$$

$$= \frac{\sum_{i=0}^n \mu_Y(y_i) \cdot s_i \cdot \left(1 - \frac{r}{r+1} \cdot \mu_Y^{\frac{1}{r}}(y_i)\right) \cdot y_i}{\sum_{i=0}^n \mu_Y(y_i) \cdot s_i \cdot \left(1 - \frac{r}{r+1} \cdot \mu_Y^{\frac{1}{r}}(y_i)\right)}$$

2. 배치 학습

뉴로-FLC를 학습시키기에 앞서, 외부 세계에서 적당한 입출력 퍼지 분할 정보(예를 들면, 입출력 퍼지 변수들의 퍼지항 크기)를 제공해야 한다. 퍼지 분할에서 각 퍼지항은 소속 함수에 의하여 특성 지어진다. 본 논문에서는, 소속 함수가 이등변 삼각형 모양(모양 상수 $r=1$)을 갖는다고 가정한다. 입출력 소속 함수들은 두 파라메타(중심과 폭)에 의하여 정해진다.

기본적으로, 입출력 소속 함수들의 최적 파라메타의 학습 과정은 일반화된 경도 하강법의 한 종류인 교차형 역전파 학습에 기초한다. N 개의 샘플로 구성된 학습예 집합 $E = \{e^1, e^2, \dots, e^m\}$ 가 미리 알려져 있다고 가정한다. 한 학습예 $e^t (t=1, 2, \dots, N)$ 는 $(x_1^t, x_2^t; y^t)$ 로 구성된다. 여기서 x_1^t 와 x_2^t 는 t 번째 샘플의 두 입력값이고, y^t 는 t 번째 샘플의 출력값이다.

뉴로-FLC를 통한 입력값의 전방향 연산은 다음과 같이 수행된다. 두 번째 층에서 입력항 노드들은 학습예 e^t 의 입력값을 받는다. 세 번째 층에서 규칙 노드들은 들어오는 소속 함수값들중 최소값을 취한다. 네 번째 층에서 출력항 노드들은 들어오는 소속 함수값들중에서 최대값을 제공한다. 규칙 노드의 개수와 연결 패턴은 다음절에서 설명하는 포함 관계 테스트(coverage test)에 의해 결정된다. 다섯 번째 출력 언어 노드는 위의 식 (3)에 의해 정의된 새로운 COG 비퍼지화 연산식에 의해 비퍼지화값을 계산한다.

학습예 집합 E 에 대해서 제안한 COG 비퍼지화기를 사용한 FLC의 근사화 오차를 E 라고 하면 시스템 오차는 다음과 같이 계산된다.

$$E = \frac{1}{2} \sum_{i=1}^N (y^t - y_c^t)^2$$

$$= \frac{1}{2} \sum_{i=1}^N \left(y^t - \frac{\sum_{j=1}^n \mu_Y^e(y_j^t) \cdot s_j \cdot y_j}{\sum_{j=1}^n \mu_Y^e(y_j^t) \cdot s_j} \right)^2 \quad (4)$$

$$= \frac{1}{2} \sum_{i=1}^N \left(y^t - \sum_{j=1}^n a_j^t \cdot y_j \right)^2$$

여기서, n , N , 그리고 $\mu_Y^e(y_j^t)$ 는 각각 출력 퍼지항의 개수, 학습예의 개수, i 번째 학습예에 대한 i 번째 출력 퍼지항의 잘려진 유효 소속값이다. 여기서 $a_j^t = \left(\frac{\mu_Y^e(y_j^t) \cdot s_j}{\sum_{j=1}^n \mu_Y^e(y_j^t) \cdot s_j} \right)$ 은 식을 간단하게 하기 위하여 사용하였다.

뉴로-FLC를 통한 근사화 오차의 역방향 전파는 층과 층간에 다음과 같은 연산으로 이루어진다.

1) 5층

경도 하강법의 기본 과정은 초기 파라메타 벡터 (y^0, s^0) 에서 시작하여 다차원 오차 표면 $E(y, s)$ 의 기울기 $\left(\frac{\partial E}{\partial y}, \frac{\partial E}{\partial s} \right)$ 를 계산한 다음, 파라메타 벡터 (y, s) 의 다음 값들은 아래 식과 같이 다차원 오차 표면을 따르는 기울기 방향으로 이동하면서 얻어진다.

$$y_i(T+1) = y_i(T) - \eta_i^y \frac{\partial E}{\partial y_i}, \quad (5)$$

$$s_i(T+1) = s_i(T) - \eta_i^s \frac{\partial E}{\partial s_i},$$

여기서, η_i^y 와 η_i^s 는 각각 i 번째 퍼지항의 중심과 폭에 대한 학습율이다. 간단한 미분 공식과 연쇄 규칙을 이용하여, 두 파라메타의 기울기 값을 구하면 다음과 같다.

$$\frac{\partial E}{\partial y_i} = - \sum_{t=1}^N (y^t - y_c^t) a_i^t, \quad (6)$$

$$\frac{\partial E}{\partial s_i} = - \sum_{t=1}^N (y^t - y_c^t) \frac{a_i^t}{s_i} (y_i - y_c^t),$$

여기서, y^t 와 y_c^t 는 각각 t 번째 학습예에 대한 원하는 출력값과 실제 비퍼지화기에 의해 얻어진 출력값이다. 따라서, 두 출력 파라메타에 대한 학습 규칙은 다음과 같이 요약된다.

$$y_i(T+1) = y_i(T) + \frac{\sum_{t=1}^N a_i^t (y^t - y_c^t)}{\sum_{t=1}^N a_i^t}, \quad (7)$$

$$s_i(T+1) = s_i(T) + \frac{\sum_{t=1}^N \beta_i^t (y^t - y_c^t) (y_i - y_c^t)}{\sum_{t=1}^N \beta_i^t},$$

여기서 β_i^t 는 $\frac{a_i^t}{s_i}$ 이고, 학습을 η_i^t 과 η_i^s 는 각각 $\eta_i^t = \frac{1}{\sum_{i=1}^N a_i^t}$, $\eta_i^s = \frac{1}{\sum_{i=1}^N \beta_i^t}$ 로 정하였다. 앞 층으로 전파되는 오차는 다음과 같이 주어진다.

$$\delta^5 = -\frac{\partial E}{\partial \text{net}^5} = \sum_{i=1}^N (y_i' - y_i') \quad (8)$$

여기서, net^5 는 가중 입력값 $\sum_{i=1}^N a_i^t \cdot y_i$ 로 정의된다.

2) 4층

이 층에서 조정해야 되는 파라메타는 없다. 앞 층으로 전파되는 오차 (δ_i^4)을 다음과 같이 계산한다.

$$\delta_i^4 = -\frac{\partial E}{\partial \text{net}_i^4} = -\frac{\partial E}{\partial \text{net}^5} \frac{\partial \text{net}^5}{\partial \mu_{ij}^4(y_j)} = -\sum_{i=1}^N (y_i' - y_i') \times \left(s_i \cdot \frac{(y_i \cdot \sum_{j=1}^N \mu_{ij}^4(y_j) \cdot s_j - \sum_{j=1}^N \mu_{ij}^4(y_j) \cdot s_j \cdot y_j)}{(\sum_{j=1}^N \mu_{ij}^4(y_j) \cdot s_j)^2} \right) \quad (9)$$

여기서, net_i^4 은 $\mu_{ij}^4(y_j)$ 와 같은 값을 갖는다.

3) 3층

4층과 유사하게, 앞 층으로 전파되는 오차 (δ_i^3)를 다음과 같이 계산한다.

$$\delta_i^3 = -\frac{\partial E}{\partial \text{net}_i^3} = -\frac{\partial E}{\partial \text{net}^4} \frac{\partial \text{net}^4}{\partial \text{net}^3} = \delta_i^4 \quad (10)$$

여기서, 들어오는 신호들이 그대로 규칙 노드를 통과하므로, net_i^4 의 값은 net_i^3 의 값과 같다. 출력이 여러 개인 경우, 오차 신호는 $\delta_i^3 = \sum_k \delta_k^4$ 으로 나타내어지는데, 여기서 합은 한 출력 노드의 후단부에 대해서 수행되어진다.

4) 2층

2층에서는 입력 퍼지 변수의 소속 함수를 갱신시키는데, i 번째 입력 변수의 j 번째 소속 함수 $\mu_{X_i}(x_i)$ 은 다음과 같이 나타내어진다.

$$\mu_{X_i}(x_i) = 1 - \frac{|x_i - x_{ij}|}{s_{ij}} \quad (11)$$

여기서, x_{ij} 와 s_{ij} 는 각각 i 번째 입력 변수 x_i 의 j 번째 소속 함수의 중심과 폭으로 다음과 같이 갱신된다.

$$\begin{aligned} x_{ij}(T+1) &= x_{ij}(T) - \eta_{x_{ij}} \frac{\partial E}{\partial x_{ij}}, \\ s_{ij}(T+1) &= s_{ij}(T) - \eta_{s_{ij}} \frac{\partial E}{\partial s_{ij}}, \end{aligned} \quad (12)$$

여기서, $\eta_{x_{ij}}$ 와 $\eta_{s_{ij}}$ 는 각각 중심 x_{ij} 과 폭 s_{ij} 의 학습율이다. 출력 파라메타를 갱신하는 경우와 유사한 계산을 통해 입력 파라메타에 대한 오차의 기울기 값을 구하면 다음과 같다.

$$\frac{\partial E}{\partial x_{ij}} = \frac{\partial E}{\partial \mu_{X_i}(x_i)} \frac{\partial \mu_{X_i}(x_i)}{\partial x_{ij}} = -\sum_{i=1}^N (y_i' - y_i') \times \frac{s_i \cdot (\sum_{j=1}^N (x_{ij} - x_{ik}) \cdot \mu_{X_i}(x_i) \cdot s_{ik})}{(\sum_{j=1}^N \mu_{X_i}(x_i) \cdot s_{ik})^2} \cdot \text{sgn}(x_{ij}) \cdot \frac{1}{s_{ij}} \quad (13)$$

$$\frac{\partial E}{\partial s_{ij}} = \frac{\partial E}{\partial \mu_{X_i}(x_i)} \frac{\partial \mu_{X_i}(x_i)}{\partial s_{ij}} = -\sum_{i=1}^N (y_i' - y_i') \times \frac{s_i \cdot (\sum_{j=1}^N (x_{ij} - x_{ik}) \cdot \mu_{X_i}(x_i) \cdot s_{ik})}{(\sum_{j=1}^N \mu_{X_i}(x_i) \cdot s_{ik})^2} \cdot \frac{|x - s_{ij}|}{s_{ij}^2}$$

여기서, $x \geq 0$ 인 경우 $\text{sgn}(x) = 1$ 이고, $x < 0$ 인 경우 $\text{sgn}(x) = -1$ 이다. 따라서, 두 입력 파라메타의 갱신은 다음 식과 같이 요약된다.

$$\begin{aligned} x_{ij}(T+1) &= x_{ij}(T) - \text{sgn}(x - x_{ij}) \eta_{x_{ij}} \sum_{i=1}^N (y_i' - y_i') \times \\ &\quad \frac{(\sum_{j=1}^N (x_{ij} - x_{ik}) \cdot \mu_{X_i}(x_i) \cdot s_{ik})}{(\sum_{j=1}^N \mu_{X_i}(x_i) \cdot s_{ik})^2}, \\ s_{ij}(T+1) &= s_{ij}(T) - \eta_{s_{ij}} \frac{|x - s_{ij}|}{s_{ij}} \sum_{i=1}^N (y_i' - y_i') \times \\ &\quad \frac{s_i \cdot (\sum_{j=1}^N (x_{ij} - x_{ik}) \cdot \mu_{X_i}(x_i) \cdot s_{ik})}{(\sum_{j=1}^N \mu_{X_i}(x_i) \cdot s_{ik})^2}. \end{aligned} \quad (14)$$

III. 진화에 의한 퍼지 제어기 설계

일반적으로, 교사형 역전과 학습의 수렴은 초기 파라메타들의 선택에 좌우되어, 얻어진 FLC의 제어 성능이 최적이라고 보장할 수 없다. 이러한 단점을 극복하기 위하여, 많은 연구자들은 유전 알고리즘들^[13,14]에 기초한 진화 기법을 사용할 것을 제안한다. 유전 알고리즘(GA)^[13]은 자연계의 적자 생존과 유전 현상을 모방한 선택과 재생 연산을 사용하는 해집단에 근거한 반복적 적응 알고리즘의 일종으로 탐색, 최적화 그리고 기계 학습에서 매우 유용한 방법으로 판명되었다^[14]. 유전 알고리즘은 특정 문제에 대한 가능한 해를 간단한 해개체 형태의 자료구조로 표현하고 여기에 유전 연산을 적용하여 얻어지는 새로운 해를 점차 최적해로 접근하도록 한다.

GA의 동작 과정은 다음과 같다. N 개의 해개체로

구성된 해집단을 P 라고 한다. 무작위로 생성된 초기 해집단을 $P(0)$ 라하고 시간 t 의 해집단을 $P(t)$ 라 한다. $P(t)$ 에 유전 연산(재생, 교차, 변이)을 적용하여 새로운 해집단 $P(t+1)$ 를 얻는다. $P(t+1)$ 에서의 각 해개체들은 시간 t 에서의 목적 함수값에 비례하여 재생된 것들이다. 교차는 두 해개체를 임의의 위치에서 절단하고, 첫 번째 해개체의 일부가 다른 하나의 해개체로 옮기는 등의 방법으로 유전 물질을 서로 교환하여 재결합한다. 변이는 임의로 선택된 유전자의 값의 일부를 변경한다. GA의 전체적 효과는 더 높은 목적 함수값을 갖는 더 나은 해를 갖도록 새로운 해집단을 생성해 나가는 것이다. 그림 2는 GA의 동작과정을 pseudo-code로 표현한 것이다.

```

Begin
  t = 0
  Initialize P(t);
  Evaluate fitness functions in P(t);
  while ~(stop_conditions) do
    Begin
      t = t + 1;
      Select P(t) from P(t-1);
      Recombine chromosomes in P(t);
      Evaluate fitness functions in P(t);
    End
  End

```

그림 2. 유전 알고리즘의 일반적인 동작 과정
Fig. 2. A general procedure of GAs.

최적 퍼지 제어기를 찾는데 GA를 적용할 때, 다음의 다섯 가지 문제를 고려하여야 한다.

1. 해개체 표현

입출력 변수가 $(x_1, x_2, \dots, x_m; y_1, y_2, \dots, y_n)$ 으로 나타내어지는 FLC를 생각한다. 여기서, 입력 변수 x_i 는 N_x 퍼지항으로 분할되고, 출력 변수 y_j 는 N_y 퍼지항으로 분할된다. 여기서는 두 입력과 하나의 출력 $(x, \phi; \theta)$ 를 갖고 각 변수가 각각 $(N_x, N_\phi; N_\theta)$ 개의 퍼지항으로 분할된 FLC를 생각한다. FLC를 표현하는 해개체는 $2 \times (N_x + N_\phi + N_\theta)$ 의 연속적인 실수로 구성된다. 여기서, 첫 $2 \times N_x$ 실수는 입력 변수 x 의 중심과 폭에 대응하고, 다음 $2 \times N_\phi$ 실수는 입력 변수 ϕ 의 중심과 폭에 대응하고, 마지막 $2 \times N_\theta$ 실수는 출력 변수 θ 의 중심과 폭에 대응한다. 본 논문에서는 퍼지항의 각 구성 요소가 실수이기 때문에 해개체가 실수

표현을 따른다.

2. 초기 해집단 생성

초기 해집단은 유전 알고리즘의 수렴 속도와 언어지는 FLC의 제어 성능을 구현하는데 중요하다. 초기 해집단에서 해개체 C_i 은 입력 설계 파라메타 (c_i, s_i) 쌍을 $(N_x + N_\phi)$ 번, 출력 설계 파라메타 (c_i, s_i) 쌍을 N_θ 번 임의로 발생시켜서 만들어진다. 퍼지항의 중심은 전체 영역에 걸쳐 균일하게 위치하도록 정해지며, 퍼지항의 폭은 가능하면 많은 학습예를 포함하도록 정해진다. 이 과정을 M 번 반복 실행하여 초기 해집단을 형성한다. 여기서 M 은 해집단의 크기이다.

3. 목적 함수

해개체의 우수성은 목적 함수에 의하여 평가된다. 본 논문에서는, 여러 다음과 같이 목적 함수에 의해 여러 요인들을 고려하였다.

첫째로, 좋은 FLC는 원하는 입출력 특성을 잘 근사화할 수 있도록 가능하면 학습에 집합 T 내의 많은 입출력 데이터 쌍들을 포함하는 퍼지 규칙 베이스를 갖는다^[4]. 한 학습예 e' 에 대한 퍼지 규칙 R_k 의 포함값은 다음과 같은 일치도 (compatibility degree) $R_k(e')$ 에 의하여 결정된다.

$$R_k(e') = (\wedge_{i=1}^M M_{x_i}^k(e')) \wedge (\wedge_{j=1}^N M_{y_j}^k(e')), \quad (15)$$

여기서, \wedge 은 대수적 곱과 같은 퍼지 AND이고, $M_{x_i}^k(e')$ 는 k 번째 퍼지 규칙에서 e' 의 단일 지지값에서의 x_i 의 절단된 소속 함수값이다. 학습예 집합 T 에 대한 퍼지 규칙 R_k 의 포함값 $R_k(T)$ 는 다음과 같이 T 에서 모든 학습예에 의한 일치도의 합으로 정의된다.

$$R_k(T) = \sum_{e' \in C_{RB}} R_k(e'). \quad (16)$$

학습예 집합 T 에 의한 퍼지 규칙 베이스 RB 의 포함값 $C_{RB}(T)$ 는 다음과 같이 규칙 베이스내 모든 퍼지 규칙들에 의한 포함값 $R_k(T)$ 의 전체 합으로 정의된다.

$$C_{RB}(T) = \sum_{k=1}^{|RB|} R_k(T) = \sum_{k=1}^{|RB|} \sum_{i=1}^N R_k(e') \quad (17)$$

여기서, $|RB|$ 와 N 은 각각 RB 내 퍼지 규칙의 수와 T 내 학습예의 수이다. 학습예 e' 가 RB 내 하나 이상의 퍼지 규칙에 포함될 때, 최대 포함값을 나타내는

퍼지 규칙에 포함시킨다. 따라서, 포함값을 고려한 목적 함수 $F_c(T)$ 는 $F_c(T) = \frac{1}{1 + C_{RB}(T)^{-1}}$ 로 나타내어진다.

이러한 목적 함수는 높은 포함값을 갖는 해개체가 더 좋은 해개체라는 것을 반영한다.

둘째로, 좋은 FLC는 가능하면 퍼지 규칙의 수가 적은 규칙 베이스를 갖는다. 포함값이 너무 적은 ($\leq \delta$) 퍼지 규칙은 유용한 퍼지 규칙 베이스에서 제거된다. 따라서, 주어진 퍼지 분할 하에서 유용한 퍼지 규칙의 수 $N_{RB}(T)$ 는 다음과 같이 정의된다.

$$N_{RB}(T) = (\#(R_k(T)))(\sum_{e' \in R_k} R_k(e') > \delta). \quad (18)$$

따라서, 유용한 퍼지 규칙의 수를 $N_{RB}(T)$ 라고 할 때 이를 고려한 목적 함수 $F_n(T)$ 는 $\frac{1}{1 + N_{RB}(T)}$ 로 나타내어진다. 이러한 목적 함수는 유용한 퍼지 규칙의 수가 적은 FLC가 더 좋은 해개체라는 것을 반영한다.

셋째로, 좋은 FLC는 가능하면 적은 시스템 근사화 오차를 나타낸다. 이 요구 조건은 근사화 오차를 고려한 목적함수 $F_e(T) = \frac{1}{1 + E(T)}$ 을 정의함으로써 만족된다. 여기서, $E(T)$ 는 식(4)에서 정의된 시스템 오차이다. 이러한 목적 함수는 보다 적은 근사화 오차를 나타내는 FLC가 더 좋은 해개체라는 것을 반영한다.

넷째로, 좋은 FLC는 좋은 제어 성능을 보여준다. 제안한 FLC 설계 방법의 타당성을 보이하고자 사용된 트럭 후진 주차 제어 문제에서, 좋은 제어 성능은 적은 주행 거리를 의미한다. 초기 위치 $x^0 = (x_1^0, x_2^0, \dots, x_m^0)$ 에서 x^1, x^2, \dots, x^{L-1} 을 통하여 목적 위치 $x^L = (x_1^L, x_2^L, \dots, x_m^L)$ 까지 주행하는데 L 스텝이 걸린다고 하면, 학습예 e' 에 대한 주행 거리 $d(e')$ 는 다음과 같다.

$$d(e') = \sqrt{\sum_{i=0}^{L-1} \sum_{j=1}^m (x_j^{i+1} - x_j^i)^2}. \quad (19)$$

학습예 집합 T 내 모든 학습예들에 의한 평균 주행 거리 $d(T)$ 는 다음과 같이 모든 학습예들을 사용한 주행 거리들의 평균으로 정의된다.

$$d(T) = \frac{1}{N} \sum_{i=1}^N d(e'), \quad (20)$$

여기서, N 은 T 에서 학습예들의 수이다. 트럭 후진 주차 제어 문제에 대하여 제어 성능을 고려한 목적 함수

는 $F_f(T)$ 는 $\frac{1}{1 + d(T)}$ 로 나타내어진다. 이러한 목적 함수는 보다 짧은 평균 주행 거리를 갖는 FLC가 더 좋은 해개체라는 것을 반영한다.

위의 네 가지 구성 요소들은 서로 다른 동적 범위를 가지기 때문에, 본 논문에서는 $f_i = \frac{f_i - m_i}{\delta_i}$ 과 같은 변수 변환을 통해 각 항이 똑 같은 정규분포 $N(0,1)$ 를 갖도록 Gaussian 정규화^[16]를 수행하였다. 여기서, m_i 와 δ_i 는 개체군에 대한 변수 f_i 의 평균과 표준편차이다. Gaussian 정규화 과정을 거친 네 가지 목적 함수들은 다음과 같이 하나의 목적 함수 $F(T)$ 로 통합된다.

$$F(T) = w_1 \cdot F_c(T) + w_2 \cdot F_n(T) + w_3 \cdot F_e(T) + w_4 \cdot F_f(T) \\ = \frac{w_1}{1 + C_{RB}(T)^{-1}} + \frac{w_2}{1 + N_{RB}(T)} + \frac{w_3}{1 + E(T)} + \frac{w_4}{1 + d(T)}. \quad (21)$$

여기서, $w_1, w_2, w_3,$ 와 w_4 는 각 목적 함수의 중요성을 결정하는 적절한 가중 상수로서 정규화 조건 $\sum_{i=1}^4 w_i = 1.0$ 을 만족한다.

4. 유전 연산

FLC의 초기 해집단은 주어진 제어 목적들에 대한 최적 FLC를 찾기 위한 적절한 유전 연산들에 의하여 진화된다. 진화에 의한 FLC 설계에서 사용된 유전 연산들에 대한 상세한 설명 다음과 같다.

1) 재생산

해개체들을 재생하기 위하여 여러 가지 선택 방법을 혼합하여 사용한다. 첫 번째 선택 방법은 elitism으로 가장 높은 목적 함수값을 갖는 해개체가 새로운 해집단에 그대로 전달되는 것이다. 두 번째 선택 방법은 수정된 k -토너먼트 방법^[17]으로 높은 목적 함수값을 갖는 상위 해개체군에서 임의로 선택한 k 개의 해개체 중에서 가장 높은 목적 함수값을 갖는 해개체가 재생산을 위하여 선택된다. 위 과정을 계속 반복하여 얻어진 두 FLC 해개체 C 와 C' 는 뒤에서 설명될 교차와 돌연변이 연산을 적용하여 새로운 해개체 C 를 만든다. 새로운 해개체 C 는 목적 함수값의 낮은 부분에서 임의로 선택한 k 개체들 중에서 가장 나쁜 목적 함수값을 갖는 개체를 대체한다. 이상의 재생산 과정은 $pselect \times |P|$ 번 반복되는데, 여기서 $|P|$ 는 해집단의 크기이다. 마지막으로, 해집단의 나머지 부분은 목적 함수값의 크기에 따라서 상위의 속하는 해개체를 차례로

복사하여 채워진다. 그림 3은 세 가지 다른 재생산 방법들을 갖는 혼합 재생 방법을 보여 준다.

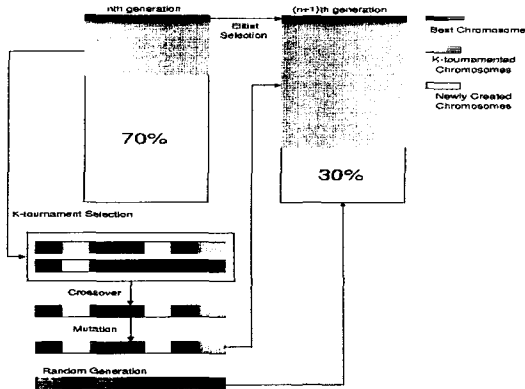


그림 3. 혼합 재생산 방법
Fig. 3. A hybrid reproduction scheme.

2) 교배

기본적으로, 교차 연산은 빈도수 P_c 로 일어나는 목적 함수값의 가중 평균 연산에 기초한다. 선택된 두 부모 개체와 대응하는 목적 함수값을 각각 C^1 과 C^2 및 F^1 과 F^2 라고 하자. 선택된 두 부모 개체들 사이의 모든 대응하는 설계 파라미터들은 같은 방법으로 처리되기 때문에, 여기서는 두 선택된 개체 중에서 i 번째 퍼지항의 특정 설계 파라미터만을 설명하기로 한다. 두 선택된 개체들의 i 번째 퍼지항들의 설계 파라미터를 각각 (x_i^1, s_i^1) 과 (x_i^2, s_i^2) 라고 하자. 그러면, 교차 연산을 적용하여 생성된 새로운 개체 C 의 설계 파라미터 (x_i, s_i) 은 다음의 식에 의해서 얻어진다.

$$\begin{aligned} x_i' &= \frac{F^1 \cdot x_i^1 + F^2 \cdot x_i^2}{F^1 + F^2} \\ s_i' &= \frac{F^1 \cdot s_i^1 + F^2 \cdot s_i^2}{F^1 + F^2} \end{aligned} \quad (22)$$

3) 변이

새로운 개체 C 에 대한 돌연 변이 연산은 다음과 같이 수행된다. 먼저, 빈도수 P_m 의 확률로 선택된 한 해개체에 대해 한 돌연 변이점을 임의로 선택한다. 다음, 돌연변이된 해개체 C 의 설계 파라미터값 (x_i'', s_i'') 은 다음 식에 의하여 연산된다.

$$\begin{aligned} x_i'' &= x_i' + \text{random}(-\Delta x', \Delta x') \\ s_i'' &= s_i' + \text{random}(-\Delta s', \Delta s'), \end{aligned} \quad (22)$$

여기서, $\Delta x'$ 와 $\Delta s'$ 는 각각 돌연 변이가 일어날 때 중

심값 및 폭의 최대 변동량이다.

IV. Lamarckian 상호 적응에 의한 FLC 설계

역전과 학습에서 얻어진 뉴로-FLC의 제어 성능은 초기 파라메타들의 선택에 매우 민감하게 반응하고 최적 제어 성능을 보장하지는 못한다. 또한, 최적 FLC를 결정하기 위한 진화적 접근은 일반적으로 만족할만한 제어성능의 수준에 도달하는데 많은 세대를 지나야 한다. 이러한 문제들을 해결하기 위하여, 본 논문에서는 생물학적 가설에서 얻어진 진화와 학습의 Lamarckian 상호 적응에 기반한 FLC의 새로운 설계 방법을 소개한다. 기본적으로, Lamarckian 상호 적응 설계 기법은 학습을 진화에 포함시킨다. 이 기법에서는 각 개체는 역전과 규칙에 학습되며, 개체들끼리는 유전 알고리즘을 통한 진화를 수행한다. 그림 4는 최적 FLC 설계를 위한 제안한 Lamarckian 상호 적응 방법을 보여준다. 제안한 설계 절차의 상세한 설명은 다음과 같다.

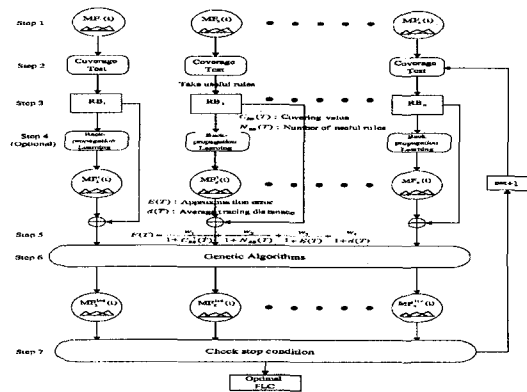


그림 4. 최적 FLC 설계를 위한 라마키안 상호 적응
Fig. 4. Lamarckian co-adaptation for an optimal FLC design.

Step 1: 초기화

학습예 $T = \{e_1, e_2, \dots, e_N\}$ 을 준비하고 시간 인덱스 $t=0$ 으로 한다.

주어진 N 에 대해, M 개의 해개체 $C(t) = \{C_1(t), C_2(t), \dots, C_M(t)\}$ 을 만든다. 여기서, 각 해개체는 하나의 FLC에 대응한다. 각 FLC는 m 개의 입력과 n 개의 출력 $(x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_n)$ 으로 구성되고 각 입출력 x_i 와 y_j 는 각각 N_{x_i} 개와 N_{y_j}

개의 퍼지항으로 구성되므로, 각 해개체는 $2 \times (\sum_{i=1}^m N_{x_i} + \sum_{j=1}^m N_{y_j})$ 개의 실수의 연속적인 나열로 이루어진다.

Step 2: 각 FLC의 포함 정도 테스트를 수행한다.

식 (17)과 (18)을 이용하여 학습에 집합 T 에 대한 각 해개체에 의한 포함값 $C_{RB}(T)$ 와 퍼지 규칙의 수 $N_{RB}(T)$ 를 각각 구한다.

Step 3: 각 FLC의 퍼지 규칙 베이스를 구축한다.

식 (18)을 이용하여 원 퍼지 규칙 베이스로부터 유용한 퍼지 규칙만을 택한다.

Step 4: (optional) 각 FLC를 학습시킨다.

설계 파라미터가 더 이상 변하지 않을 때까지 3장에서 설명한 역전파 배치 학습에 의해 각 퍼지항의 설계 파라미터를 갱신한다. 각 FLC의 학습 시간은 각 해개체가 갖는 설계 파라미터의 초기값의 차이 때문에 서로 다를 수 있다.

Step 5: 각 FLC의 목적 함수값을 구한다.

식 (4)와 (20)을 이용하여 학습에 집합 T 에 대한 각 FLC의 근사화 오차 $E(T)$ 와 평균 주행 거리 $d(T)$ 를 각각 구한다. 설계 파라미터가 더 이상 변하지 않을 때까지 3장에서 설명한 역전파 배치 학습에 의해 각 퍼지항의 설계 파라미터를 갱신한다. 각 FLC의 학습 시간은 각 해개체가 갖는 설계 파라미터의 초기값의 차이 때문에 서로 다를 수 있다. 다음, 식 (21)을 이용하여 통합 목적 함수값 $F(T)$ 를 구한다.

Step 6: FLC의 해집단을 진화시킨다.

앞 단계에서 얻어진 통합 목적 함수값을 이용하여 제안한 혼합 재생산 방법을 수행한다. 식 (22)와 (23)을 이용하여 교배 및 돌연 변이 연산을 각각 수행한다.

Step 7: 종료 조건을 테스트한다.

만약 시간 인덱스 i 가 미리 정해진 세대수 (t_{max})에 도달하면, 알고리즘 수행을 멈춘다. 이때 얻어진 가장 좋은 해개체 $C_{best}(t)$ 를 원하는 FLC로 선택한다. 그렇지 않으면 $t=t+1$ 로 놓고, 스텝 2로 되돌아간다.

그림 5. 제안한 라마키안 상호 적응 FLC 설계 알고리즘

Fig. 5. Proposed Lamarckian co-adaptive FLC design algorithm.

V. 시뮬레이션 결과 및 논의

제안한 Lamarckian 상호 적응 방법을 트럭 후진 주차 제어를 위한 최적 FLC 설계에 적용하여 다른 FLC 설계 방법에 의한 것과 근사화 능력과 제어 성능 면을 각각 근사화 오차와 평균 주행 거리를 이용하여 비교한다. 트럭 주차 문제의 목표는 가능한 빨리, 그리고 정확하게 트럭을 주차시키는 것이며, 이 제어 문제는 기존 제어 기술로는 해결될 수 없는 전형적인 비선형 제어 문제이다. 그림 6은 트럭 주차 제어 문제에서 사용된 트럭과 주차대의 위치를 보여준다. 트럭의 위치는 (x, y, ϕ) 에 의해 결정되는데 여기서, ϕ 는 트럭 진행 방향과 x 축간의 각도이며, 트럭의 후진 주행 제어는 트럭의 주행 방향과 핸들의 축 간의 각도인 θ 에 의하여 결정된다. 트럭의 후진 주차 제어 문제의 근사화 제어 운동 방정식은 아래와 같이 나타내어진다 (상세한 것은 [18] 참조).

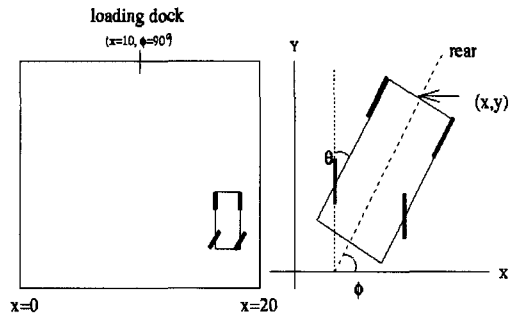


그림 6. 모형 트럭과 주차대
Fig. 6. A model truck and a loading dock.

$$\begin{aligned} x(t+1) &= x(t) + \cos[\phi(t) + \theta(t)] + \sin[\theta(t)]\sin[\phi(t)] \\ y(t+1) &= y(t) + \sin[\phi(t) + \theta(t)] - \cos[\theta(t)]\sin[\phi(t)] \\ \phi(t+1) &= \phi(t) - \sin^{-1}\left[2\sin\left(\frac{\theta(t)}{b}\right)\right] \end{aligned} \quad (24)$$

여기서 b 는 트럭의 길이이다. 본 논문에서는 $b=4$ 로 하였다. 만약 트럭과 주차대까지의 거리가 충분하다면 트럭이 $x=10, \phi=90^\circ$ 가까이 오면 트럭을 곧장 후진하기만 하면 되기 때문에 변수 y 를 퍼지 입력 변수 (x, y, ϕ) 에서 뺄 수 있다. 그래서, 트럭 주차 제어 문제에 대한 FLC의 설계 문제는 주어진 공간 ($0 \leq x \leq 20, -90^\circ \leq \phi \leq 270^\circ$)내의 임의의 초기 위치 (x_0, ϕ_0) 에서 가능하면 신속·정확하게 주차대 ($x=10, \phi=90^\circ$)쪽으로 후진하는 것으로 생각할 수 있다.

Wang과 Mendel의 논문 [19]와 공정한 비교를 하기 위하여, 입출력 변수는 5 또는 7개의 퍼지항으로

분할된다. 그림 7은 Wang과 Mendel의 논문에서 사용된 퍼지 입출력 변수의 소속 함수와 퍼지 규칙 베이스를 보여준다. 본 논문에서는 Wang의 논문에서와 똑같이 5 분할시 (S2, S1, CE, B1, B2), 7 분할시 (S3, S2, S1, CE, B1, B2, B3)을 사용하여 언어 변수의 퍼지항을 명명하였으며, 학습에 집합은 14개의 초기 지점에서 주차대까지 부드러운 주행 궤도들을 구성하는 238개의 입출력 데이터 쌍으로 구성되어 있다(상세한 것은 [18]의 표 1에서 표 14까지를 참조).

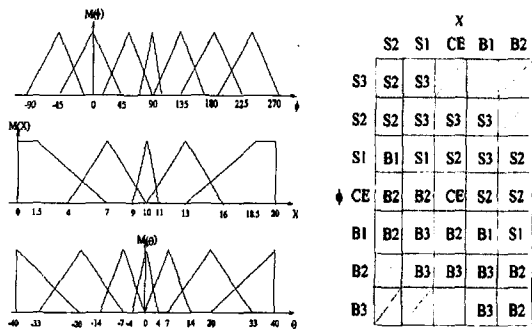


그림 7. Wang의 논문에서 사용된 소속함수(왼쪽)와 퍼지 규칙 베이스(오른쪽)
 Fig. 7. Membership functions(left) and fuzzy rule base(right) used in Wang's work.

표 1. 학습과 진화에 사용된 실행 파라메타
 Table 1. Execution parameters used in the learning and evolution.

	GA evolution-only	Lamarckian co-adaptation
No. of generations	100	100
Population size	100	100
Chromosome size	2 × (5 + 7 + 7)	2 × (5 + 7 + 7)
No. of k-tournament	3	3
pselect	0.5	0.5
Pc	0.3	0.3
Pm	0.1	0.1
No. of training examples	238	238
No. of tracing test samples	14	14
Weights $w_i(i=1, \dots, 4)$	0.25	0.25
η_{x_i}	-	0.01
η_{s_i}	-	0.10

본 논문에서는 유전 알고리즘만을 이용한 FLC 설계와 Lamarckian 상호 적응(역전파 학습 + 유전 알고리즘 진화)을 이용한 FLC 설계의 두 가지 다른 시

뮬레이션을 수행하였다. 표 1은 본 논문의 시뮬레이션에서 사용된 진화와 학습의 실행 파라메타값을 보여주는 데 이들은 여러 번의 실험을 통해 실험적으로 결정되었다. FLC 설계 방법이 어떤 실행 파라메타에 의존적이지 않으면, 대응하는 항목은 '-' 심벌로 나타내었다.

그림 8은 목적 함수의 가중치들이 모두 같을 때 ($w_i=0.25, i=1, \dots, 4$), 목적 함수 $F(T)$ 의 진화 곡선을 보여준다. 여기서, 두 진화 곡선은 각 유전 알고리즘에 의한 진화와 Lamarckian 상호 적응에 의해 얻어진 것으로, 각 세대에서 해집단 내에서 식 (21)에 의해 계산된 목적 함수중 가장 좋은 것을 나타낸 것이다. 그림 8은 진화 초기에 Lamarckian 상호 적응 방법이 국부적 탐색과 오차 역전파를 이용한 파라메타 갱신을 수행하기 때문에 유전 알고리즘만을 이용한 방법보다 더 나은 FLC를 찾는다라는 것을 보여준다.

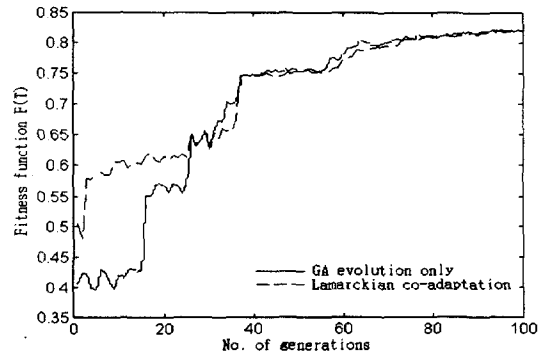


그림 8. 다른 FLC 설계 방법들의 진화 곡선
 Fig. 8. Evolution curves of different design methods.

그림 9는 목적 함수의 가중치들이 모두 같을 때 ($w_i=0.25, i=1, \dots, 4$), (a) 학습하지 않은 Wang의 FLC, (b) 유전 알고리즘만 이용한 FLC, 그리고 (c) Lamarckian 상호 적응된 FLC와 같이 세 가지 다른 FLC 설계 방법에서 얻어진 퍼지 규칙 베이스를 각각 보여준다. 이 그림으로부터 (1) 학습하지 않은 Wang의 FLC, 유전 알고리즘만을 이용한 FLC, 그리고 Lamarckian 상호 적응된 FLC의 퍼지 규칙 베이스내 유용한 규칙들의 수는 각각 27, 27 그리고 26개이며, (2) Wang의 FLC와 유전 알고리즘만을 이용한 FLC의 규칙 표현은 퍼지 규칙의 개수가 서로 같지만 퍼지 규칙 표현은 서로 다름을 알 수 있고(본 실험의 경우, 6개의 서로 다른 규칙을 발견 할 수 있음), (3)

Lamarckian 상호 적응된 퍼지 규칙 베이스는 가장 적은 수의 퍼지 규칙을 가짐을 알 수 있다.

	S3	S2	S1	CE	B1	B2	B3
S2	S3	S2	B1	B2	B2		
S1	S3	S3	S1	B2	B2	B3	
CE		S3	S2	CE	B2	B3	B3
B1			S3	S2	B1	B2	B3
B2				S2	S1	B2	B2

(a) 학습하지 않은 Wang의 FLC

	S3	S2	S1	CE	B1	B2	B3
S2	S3	S2	B1	B2	B2		
S1	S3	S3	S1	B2	B2	B3	
CE		S3	S2	CE	B2	B3	B3
B1			S3	S2	B1	B2	B3
B2				S2	S1	B2	B2

(b) GA만을 이용한 FLC

	S3	S2	S1	CE	B1	B2	B3
S2	S2	S2	B1	B2	B2		
S1	S3	S3	S1	B2	B3	B3	
CE		S3	S2	CE	B2	B2	
B1			S3	S2	B1	B3	B3
B2				S2	S2	B2	B2

(c) Lamarckian 상호 적응된 FLC

그림 9. 얻어진 퍼지 규칙 베이스
Fig. 9. Fuzzy rule bases.

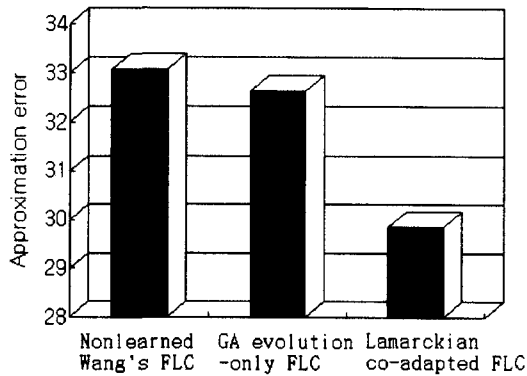


그림 10. 세 가지 다른 설계 방법사이의 근사화 오차
Fig. 10. Approximation errors among three different FLC design methods.

그림 10은 목적 함수의 가중치들이 모두 같을 때 ($w_i=0.25, i=1, \dots, 4$), 세 가지 다른 FLC 즉, 학습하지 않은 Wang의 FLC, 유전 알고리즘만을 이용한 FLC, 그리고 Lamarckian 상호 적응된 FLC들의 근사화 오차를 비교한다. 근사화 오차는 236개의 입출력 데이터로 구성된 Wang의 학습에 집합에서 $E(T) = \frac{1}{238} \sum_{i=1}^{238} (\theta(i) - \theta_c(i))^2$ 에 의해 얻어지는 데, 여기서 $\theta(i)$ 와 $\theta_c(i)$ 은 각각 주어진 입력에 대한 실제 출력과 각 FLC에 의하여 얻어진 비퍼지화값이다. 시뮬레이션 결과는 학습하지 않은 Wang의 FLC, 유전 알고리즘만을 이용한 FLC, 그리고 Lamarckian 상호 적응된 FLC의 근사화 오차가 각각 33.08, 32.63, 그리고 29.84임을 보여준다. 이로부터 Lamarckian 상호 적응

FLC 설계 방법은 효과적으로 근사화 오차를 줄인다는 것을 알 수 있다. 이러한 사실은 Lamarckian 상호 적응 FLC가 근사화 오차의 유전 알고리즘의 각 세대마다 역전과 학습에 의해 국부적 최적해를 효과적으로 탐색하기 때문에 나타나는 것으로 이해된다.

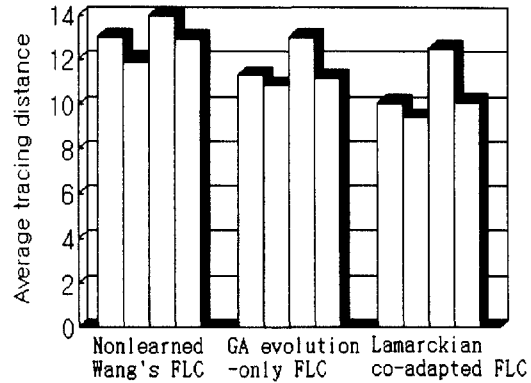
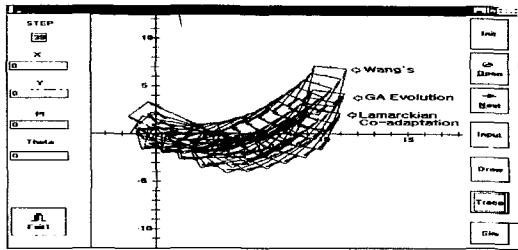


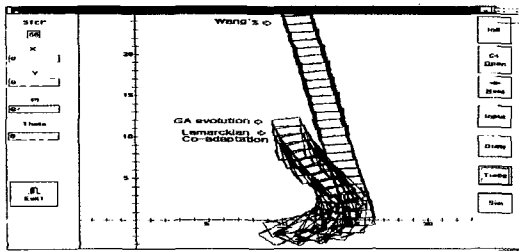
그림 11. 다른 FLC들 사이의 평균 주행 거리들의 비교
Fig. 11. Comparison of average tracing distances among different FLCs.

그림 11은 목적 함수의 가중치들이 모두 같을 때 ($w_i=0.25, i=1, \dots, 4$), 학습하지 않은 Wang의 FLC, 유전 알고리즘만을 이용한 FLC, 그리고 Lamarckian 상호 적응된 FLC의 제어 성능들을 각각 비교한다. 제어 성능은 ($0 \leq x \leq 20, 0 \leq y \leq 50, -270^\circ \leq \phi \leq 90^\circ$) 의 구간에서 임의로 선택된 100개의 시작 위치에서 얻어진 평균 주행 거리 $d(T) = \frac{1}{100} \sum_{i=1}^{100} \sum_{j=1}^n \sqrt{(x_{i,j} - x_{i,j-1})^2}$ 를 이용하여 비교하였다. 본 논문에서는 시작 위치들의 자의적 선택으로 인하여 발생할 수 있는 tweak 문제를 피하기 위하여 위의 실험을 4번 반복 수행한 결과를 비교하였다. 시뮬레이션 결과는 학습하지 않은 Wang의 FLC, 유전 알고리즘만을 이용한 FLC, 그리고 Lamarckian 상호 적응된 FLC의 평균 주행 거리들이 각각 12.96, 11.59, 그리고 10.67 스텝임을 보여준다. 이로부터 Lamarckian 상호 적응에 의한 FLC 설계 방법이 가장 좋은 제어 성능을 보임을 알 수 있는데, 이는 Lamarckian 상호 적응이 유전 알고리즘을 통한 진화의 매 세대마다 역전과 학습을 수행할 때, 이 학습이 제한한 COG 비퍼지화기로 하여금 바퀴 회전각도 θ 를 보다 정확히 조정하는 역할을 하기 때문으로 생각된다. 그림 7에서 바퀴

회전각도 θ 의 소속 함수를 보면, 바퀴 회전각도 $|\theta|$ 의 절대값이 커질수록 소속 함수의 폭이 넓어짐을 알 수 있다. 제한한 COG 비퍼지화기는 바퀴 회전각도 θ 가 커짐에 따라 얻어지는 비퍼지화값의 가중치를 크게 하여 이를 반영하며, 이러한 효과를 매 세대마다 학습을 통해 증폭시킴으로서 보다 정확한 제어 성능을 나타낸다.



(a) (2.0, 0.0, -40°)



(b) (12.0, -2.0, 0.0°)

그림 12. 서로 다른 FLC간의 주행 곡선 비교
Fig. 12. Trajectories among different FLCs.

그림 12는 목적 함수의 가중치들이 모두 같을 때 ($w_i=0.25, i=1, \dots, 4$), 세 가지 다른 퍼지 제어기, 즉 학습하지 않은 Wang의 FLC, 유전 알고리즘만을 이용한 FLC, 그리고 Lamarckian 상호 적응된 FLC에 대해 두 위치 (a) (2.0, 0.0, -40°)와 (b) (12.0, -2.0, 0.0°)에서 시작하는 모형차의 주행 곡선을 나타낸 것이다. 시뮬레이션 결과는 학습하지 않은 Wang의 FLC, 유전 알고리즘만을 이용한 FLC, 그리고 Lamarckian 상호 적응된 FLC의 각 경우에 있어 목표점에 도달하는 데 (a) 시작 위치에 대해서는 각각 14, 13, 그리고 12 스텝 걸리고, (b) 시작 위치에 대해서는 각각 34, 15, 그리고 14 스텝 걸림을 보여준다.

VI. 결 론

본 논문은 학습과 진화의 결합을 이용한 여러 개의

FLC를 Lamarckian 상호 적응시켜 최적의 뉴로 FLC를 얻는 새로운 설계 기법을 제안하였다. 제안한 Lamarckian 상호 적응은 일생동안 개체가 학습에 의하여 획득된 표현형의 특성이 개체의 후손에 유전 인자 형태로 직접 전달된다는 가설에 기반한 것이다. 따라서, 학습은 탐색 영역의 모양을 변형시키고 이 변형이 상호 적응된 개체 집합들 사이의 좋은 진화 경로를 제공하는 방식으로 학습이 직접 진화 경로를 유도할 수 있다. 최적 뉴로-FLC의 설계에서, 진화는 전체 탐색을 통하여 FLC들간의 설계 파라메타들을 조정하는 반면, 학습은 국부 탐색을 통하여 각 개체 FLC 내에서 설계 파라메타들을 조정했다. 이러한 상호 적응의 가장 중요한 특징은 진화시 수렴까지 걸리는 세대수를 크게 줄여준다는 점이다.

본 논문에서는 세 가지 다른 FLC (학습하지 않은 Wang의 FLC, 유전 알고리즘만을 이용하여 얻은 FLC, 그리고 Lamarckian 상호 적응된 FLC)의 (1) 생성된 퍼지 규칙 베이스, (2) 근사화 능력, 그리고 (3) 제어 성능을 각각 유용한 규칙들의 개수, 근사화 오차, 그리고 평균 주행 거리에 의해 서로 비교하였다. 시뮬레이션 결과는 Lamarckian 상호 적응된 FLC가 세 가지 서로 다르게 얻어진 FLC중 모든 면에서 가장 좋은 성능들을 나타내었다. 제안한 Lamarckian 상호 적응 방법의 또 다른 장점은 단순히 목적 함수의 가중치 w_i 를 변경함으로써, 여러 가지 설계 목적 (예를 최소 규칙수를 갖는 FLC, 좋은 근사화 능력을 보이는 FLC, 좋은 제어 성능을 보이는 FLC, 또는 이들 간의 균형된 성능을 나타내는 FLC등)을 만족시킬 수 있으므로, 여러 가지 주문형 (Application-specific) FLC를 쉽게 얻을 수 있는 환경을 제공한다는 점이다.

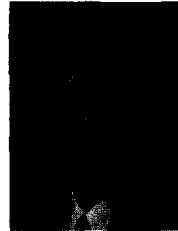
참 고 문 헌

- [1] E. H. Mandani, "Application of Fuzzy Algorithms for Control of Simple Dynamic Plant," *IEEE Proc. Control and Science*, 121(12) (1974) 1585-1588.
- [2] C. Lin and C. S. Lee, "Neural-Network-Based Fuzzy Logic Control and Decision System," *IEEE Trans. on Computers*, 40(12) (1991) 1320-1336.
- [3] J. R. Jang, "ANFIS: Adaptive-Neural-Network-Based Fuzzy Inference System,"

- IEEE Transactions on System, Man, and Cybernetics*, 23(2) (1993) 665-685.
- [4] J. L. Castro, M. Degrade and F. Headgear, "A Learning Method of Fuzzy Reasoning by Genetic Algorithms," *Proceeding of First European Congress on Fuzzy and Intelligent Technologies*, (1993) 804-809.
- [5] M. A. Lee and H. Tacking, "Integrating Design Stages of Fuzzy Systems using Genetic Algorithms," *IEEE 2na Int' l Cong. on Fuzzy Systems*, (1993) 612-617.
- [6] G. E. Hinton and S. J. Nolan, "How learning can guide evolution," in: R. K. Blew and M. Mitchell, EBS., 447-454, *Adaptive Individuals in Evolving Populations: Models and Algorithms*, (Addison Wesley, 1996).
- [7] D. Pairs and S. Noli, "The influence of learning on evolution," in: R. K. Blew and M. Mitchell, EBS., 419-430, *Adaptive Individuals in Evolving Populations: Models and Algorithms*, (Addison Wesley, 1996).
- [8] R. K. Blew, J. McInerney, and N.N. Schraudolph, "Evolving networks: Using the genetic algorithm with connectionist learning," in: C. G. Langton, J. D. Farmer, S. Rasmussen, and C. E. Taylor, eds., *Artificial Life II*, 511-548, (Addison-Wesley, 1991).
- [9] G. F. Miller, P. M. Todd, and S. U. Hedge, "Designing neural networks using genetic algorithms," *Proceedings of the Third International Conference on Genetic Algorithms*, (1989).
- [10] J. B. Lamarck, "Of the Influence of the Environment on the Activities and Habits of Animals," *Zoological Philosophy*, (Macmillan, London, 1914).
- [11] D. E. Ackley, and M. L. Littman, "A Case for Lamarckian Evolution," in: C. G. Langton, ed., *Artificial Life III*, 3-10, (Addison-Wesley, 1994).
- [12] Daijin Kim and In-Hyun Cho, "An Accurate and Cost-effective COG Defuzzifier without the Multiplier and the Divider," *Fuzzy Sets and Systems*, (In print, 1998).
- [13] J. H. Holland, *Adaptation in Neural and Artificial Systems*, (The University of Michigan Press, Ann Arbor, 1975).
- [14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, (Addison-Wesley Press, 1989).
- [15] A. H. Wright, "Genetic algorithms for ream parameter optimization," G. Rawlins ed., *Foundations of Genetic Algorithms*, (Morgan Kaufmann Publishers, 1991), 205-220.
- [16] R. L. Scheaffer and J. T. McClave, *Pobability and Statistics for Engineers*, (PWS-KENT Publishing Company, Boston, 1990).
- [17] D. E. Goldberg and K. Deb, "A Comparative Analysis of Selection Schemes Used in Generic Algorithms," Ed. Gregory J. E Rawlins, *Fundamental of Genetic Algorithms*, (Morgan Kaufmann Publishers, San Mateo, 1991) 69-93.
- [18] L. X. Wang and J. M. Mendel, "Generating Fuzzy Rules from Numerical Data, with Applications," *USC-SIPI Report*, 169 (1991).
- [19] L. X. Wang and J. M. Mendel, "Generating Fuzzy Rules by Learning From Examples," *IEEE Transactions on System, Man, and Cybernetics*, 22(6) (1992) 1414-1427.

 저 자 소 개

金 大 鎮(正會員) 第 34卷 C編 第 7號 參照
 현재 동아대학교 전기전자컴퓨터공학
 부 부교수



李 韓 別(正會員)
 1997년 2월 경성대학교 컴퓨터공학
 과(공학사), 현재 동아대학교 컴퓨터
 공학과 석사과정, 주관심 분야는 지
 능 시스템, 컴퓨터 네트워킹, 디지털
 신호처리등

姜 大 星(正會員) 第 34卷 S編 第 8號 參照
 현재 동아대학교 전기전자컴퓨터공학
 부 조교수