

라마키안 상호 적응에 의한 뉴로-퍼지 제어기의 최적 설계

An Optimal Design of Neuro-Fuzzy Logic Controller Using Lamarckian Co-adaptation

이한별, 김대진

Hanpyul Lee, Daijin Kim
동아대학교 컴퓨터공학과

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 성능등 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.

1. 서 론

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].

FLC를 설계하는 한 방법은 신경망 구조와 학습 능력을 이용한 뉴로-퍼지 방법[2,3]이다. 이 방법은

1) 산술적 데이터만 이용할 수 있고, 2) 국부체에 빠지기 쉽기 때문에 항상 최적 시스템 성능을 기대할 수 없으며, 3) 가중치의 내부적 표현 성질 때문에 생성된 퍼지 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.

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[8]이다. D. Ackley와 M. Littman[9]은 Lamarckian 가설이 컴퓨터 상에 쉽게 구현될 수 있고, 동시에 그것이 갖는 빠른 수렴 특성으로 인해 최적화 문제에 훨씬 더 효율적이라는 것을 보였다. 본 논문에서는 최적 FLC 설계를 위하여 학습을 진화에 결합시킨 Lamarckian 상호 적용을 사용할 것을 제안한다.

2. 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 노드로서 퍼지 규칙 베이스를 형성한다. 3층에 있는 링크들은 퍼지 규칙의 전단부를 정의하고 4층에 있는 링크들은 퍼

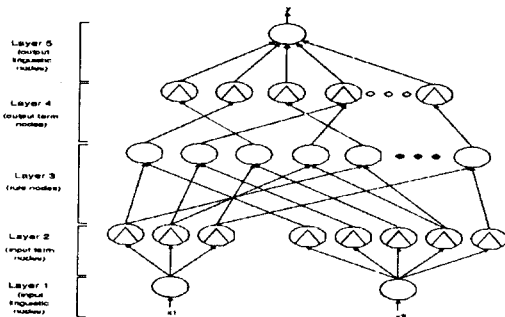


그림 1. 뉴로-퍼지 제어기의 구조

지 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[10]에 잘 기술되어 있다. 이러한 잘못을 바로잡기 위하여 본 논문은 비퍼지화값을 다음과 같이 구하는 새로운 COG 비퍼지화 방법을 제안한다.

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

여기서 $A_r(y)$ 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 $\mu_r^{\frac{1}{r}}(y)$ 의 경우에 면적 $A_r(y_i)$ 의 정확한 식의 상세한 유도는 [10]에서 나타나 있다. 여기서 r 은 소속 함수를 수축($r > 1$)시키거나 팽창($r < 1$)시키는 모양 상수이다. 유도 결과는 $\mu_r^{\frac{1}{r}}(y)$ 의 절단된 소속 함수 아래의 면적 $A_r(y_i)$ 이 다음과 같이 계산됨을 보여준다.

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

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

$$\begin{aligned} y_c &= \frac{\sum_{i=0}^n A_r(y_i) \cdot y_i}{\sum_{i=0}^n A_r(y_i)} \\ &= \frac{\sum_{i=0}^n \mu_r^e(y_i) \cdot s_i \cdot \left(1 - \frac{r}{r+1} \mu_r^{\frac{1}{r}}(y_i)\right) \cdot y_i}{\sum_{i=0}^n \mu_r^e(y_i) \cdot s_i \cdot \left(1 - \frac{r}{r+1} \mu_r^{\frac{1}{r}}(y_i)\right)} \end{aligned} \quad (3)$$

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 특성지어진다. 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 역전과 학습에 기초한다. N 개의 샘플로 구성된 학습에 집합 $E = \{e^1, e^2, \dots, e^M\}$ 가 미리 알려져 있다고 가정한다. 한 학습에 $e^t (t=1, 2, \dots, M)$ 는 $(x_1^t, x_2^t; y^t)$ 로 구성된다. 여

기서 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 학습에 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.

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

$$\begin{aligned} E &= \frac{1}{2} \sum_{t=1}^n (y^t - y_c^t)^2 \\ &= \frac{1}{2} \sum_{t=1}^n \left(y^t - \frac{\sum_{i=1}^n \mu_V^e(y_i^t) \cdot s_i \cdot y_i}{\sum_{i=1}^n \mu_V^e(y_i^t) \cdot s_i} \right)^2 \\ &= \frac{1}{2} \sum_{t=1}^n \left(y^t - \sum_{i=1}^n \alpha_i^t \cdot y_i \right)^2 \end{aligned} \quad (4)$$

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

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

2.1 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 따라 음의 기울기 방향으로 이동하면서 얻어진다.

$$\begin{aligned} y_c(T+1) &= y_c(T) - \eta_y^c \frac{\partial E}{\partial y_c}, \\ s_c(T+1) &= s_c(T) - \eta_s^c \frac{\partial E}{\partial s_c}. \end{aligned} \quad (5)$$

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

$$\begin{aligned} \frac{\partial E}{\partial y_i} &= - \sum_{t=1}^n (y^t - y_c^t) \alpha_i^t, \\ \frac{\partial E}{\partial s_i} &= - \sum_{t=1}^n (y^t - y_c^t) \frac{\alpha_i^t}{s_i} (y_i - y_c^t), \end{aligned} \quad (6)$$

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.

$$\begin{aligned} y_c(T+1) &= y_c(T) + \frac{\sum_{t=1}^n \alpha_i^t (y^t - y_c^t)}{\sum_{t=1}^n \alpha_i^t}, \\ s_c(T+1) &= s_c(T) + \frac{\sum_{t=1}^n \beta_i^t (y^t - y_c^t) (y_i - y_c^t)}{\sum_{t=1}^n \beta_i^t}. \end{aligned} \quad (7)$$

여기서 $\beta_i^t = \frac{\alpha_i^t}{s_i}$ 이고, 학습율 η_y^c 과 η_s^c 는 각각

$$\eta_y^c = \frac{1}{\sum_{t=1}^n \alpha_i^t}, \quad \eta_s^c = \frac{1}{\sum_{t=1}^n \beta_i^t}$$

로 정하였다.

앞 층으로 전파되는 오차는 다음과 같이 주어진다.

$$\delta^5 = - \frac{\partial E}{\partial net^5} = \sum_{t=1}^n (y^t - y_c^t) \quad (8)$$

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

2.2 4층

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

$$\begin{aligned} \delta_i^4 &= - \frac{\partial E}{\partial net_i^4} = - \frac{\partial E}{\partial net^5} \frac{\partial net^5}{\partial \mu_V^e(y_i)} \\ &= - \sum_{t=1}^n (y^t - y_c^t) \left(s_i \cdot \frac{(y_i \cdot \sum_{i=1}^n \mu_V^e(y_i^t) \cdot s_i - \sum_{i=1}^n \mu_V^e(y_i^t) \cdot s_i \cdot y_i)}{(\sum_{i=1}^n \mu_V^e(y_i^t) \cdot s_i)^2} \right). \end{aligned} \quad (9)$$

여기서, net_i^4 은 $\mu_V^e(y_i)$ 와 같은 값을 갖는다.

2.3 3층

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

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

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.

2.4 2층

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

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

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 파라미터를 갱신하는 경우와 유사한 계산을 통해 입력 파라미터에 대한 오차의 기울기 값을 구하면 다음과 같다.

$$\begin{aligned} \frac{\partial E}{\partial x_{ij}} &= \frac{\partial E}{\partial \mu_{X_i}(x_i)} \frac{\partial \mu_{X_i}(x_i)}{\partial x_{ij}} \\ &= - \sum_{j=1}^N (y^j - y_c^j) \frac{s_i \cdot \left(\sum_{k=1}^N (x_{ij} - x_{ik}) \cdot \mu_{X_i}(x_i) \cdot s_{ik} \right)}{\left(\sum_{k=1}^N \mu_{X_i}(x_i) \cdot s_{ik} \right)^2} \\ &\quad \cdot \frac{\text{sgn}(x_{ij}) \cdot \frac{1}{s_{ij}}}{s_{ij}}, \\ \frac{\partial E}{\partial s_{ij}} &= \frac{\partial E}{\partial \mu_{X_i}(x_i)} \frac{\partial \mu_{X_i}(x_i)}{\partial s_{ij}} \\ &= - \sum_{j=1}^N (y^j - y_c^j) \frac{s_i \cdot \left(\sum_{k=1}^N (x_{ij} - x_{ik}) \cdot \mu_{X_i}(x_i) \cdot s_{ik} \right)}{\left(\sum_{k=1}^N \mu_{X_i}(x_i) \cdot s_{ik} \right)^2} \\ &\quad \cdot \frac{|x_{ij} - s_{ij}|}{s_{ij}^2} \end{aligned} \quad (13)$$

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

$$\begin{aligned} x_{ij}(T+1) &= x_{ij}(T) - \text{sgn}(x - x_{ij}) \eta_{x_{ij}} \frac{\sum_{j=1}^N (y^j - y_c^j) \left(\sum_{k=1}^N (x_{ij} - x_{ik}) \cdot \mu_{X_i}(x_i) \cdot s_{ik} \right)}{\left(\sum_{k=1}^N \mu_{X_i}(x_i) \cdot s_{ik} \right)^2}, \\ s_{ij}(T+1) &= s_{ij}(T) - \eta_{s_{ij}} \frac{|x_{ij} - s_{ij}|}{s_{ij}} \frac{\sum_{j=1}^N (y^j - y_c^j) \left(\sum_{k=1}^N (x_{ij} - x_{ik}) \cdot \mu_{X_i}(x_i) \cdot s_{ik} \right)}{\left(\sum_{k=1}^N \mu_{X_i}(x_i) \cdot s_{ik} \right)^2}. \end{aligned} \quad (14)$$

3. Lamarckian 상호 적용에 의한 FLC 설계

본 논문에서는 생물학적 가설에서 얻어진 진화와 학습의 Lamarckian 상호 적용에 기반한 FLC의 새로운 설계 방법을 소개한다. 기본적으로, Lamarckian 상호 적용 설계 기법은 학습을 진화에 포함시킨다. 이 기법에서는 각 개체는 역전과 규칙에 학습되며, 개체들끼리는 유전 알고리즘을 통한 진화를 수행한다. 제안한 Lamarckian 상호 적용 방법의 설계 절차 다음과 같다.

Step 1: 초기화

학습에 $T = \{e_1, e_2, \dots, e_N\}$ 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 출력 x_i 와 y_j 는 각각 N_x 개와 N_y 개의 퍼지항으로 구성되므로,

각 해개체는 $2 \times \left(\sum_{i=1}^m N_x + \sum_{j=1}^n N_y \right)$ 개의 실수의 연속적인 나열로 이루어진다.

Step 2: 각 FLC의 포함 정도 테스트를 수행한다. 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. 원래의 퍼지 규칙 베이스로부터 유용한 퍼지 규칙만을 택한다.

Step 4: (optional) 각 FLC를 학습시킨다.

설계 파라미터가 더 이상 변하지 않을 때까지 3장에서 설명한 역전과 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.

학습에 집합 T 에 대한 각 FLC의 근사화 오차 $E(T)$ 와 평균 주행 거리 $d(T)$ 을 각각 구한다. 설계 파라미터가 더 이상 변하지 않을 때까지 3장에서 설명한 역전과 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 6: FLC의 해집단을 진화시킨다.

앞 단계에서 얻어진 통합 목적 함수값을 이용하여 제안한 혼합 재생산 방법을 수행한다.

Step 7: 종료 조건을 테스트한다.

만약 시간 인덱스 t 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. 식

그림 2. 제안한 Lamarckian 상호 적용 FLC 설계 알고리즘

4. 시뮬레이션 결과 및 논의

제안한 Lamarckian 상호 적용 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. 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 축간의 각도인 θ 에 의하여 결정된다. 트럭의 후진 주차 제어 문제의 근사화 제어 운동 방정식은 아래와 같이 나타내어진다 (상세한 것은 [11]참조).

$$\begin{aligned} x(t+1) &= x(t) + \cos[\phi(t) + \theta(t)] + \sin[\theta(t)]\sin[\phi(t)] \\ y(t+1) &= y(t) + \cos[\phi(t) + \theta(t)] - \sin[\theta(t)]\sin[\phi(t)] \\ \phi(t+1) &= \phi(t) - \sin^{-1} \left[2\sin \left(\frac{\theta(t)}{b} \right) \right] \end{aligned} \quad (15)$$

여기서 b 는 트럭의 길이이다. 본 논문에서는

$b=4$ 로 하였다. 트럭 주차 제어 문제에 대한 FLC의 설계 문제는 주어진 공간 ($0 \leq x \leq 20$,

$w_i=0.25, i=1, \dots, 4$), 목적 함수 $F(T)$ 의 진화 곡선을 보여준다. 여기서, 두 진화 곡선은 각 유전

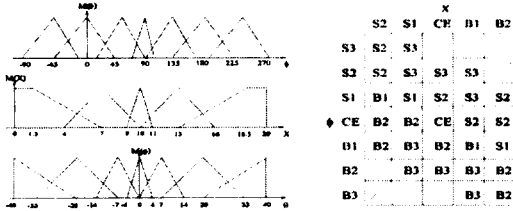


그림 3. Wang의 논문에서 사용된 소속함수(왼쪽)와 퍼지 규칙 베이스(오른쪽)

$-90^\circ \leq \phi \leq 270^\circ$ 내의 임의의 초기 위치 (x_0, ϕ_0)에서 가능하면 신속·정확하게 주차대 ($x=10, \phi=90^\circ$) 쪽으로 후진하는 것이다.

Wang과 Mendel의 논문[12]와 공정한 비교를 하기 위하여, 입출력 변수는 5 또는 7개의 퍼지항으로 분할된다. 그림 3은 Wang과 Mendel의 논문

표 1 제안한 CBFCL의 구조 및 학습 파라미터

	GA evolution-only	Lamarckian co-adaptation
No. of generations	100	100
Population size	100	100
Chromosome size	$2 \times (5+7+7)$	$2 \times (5+7+7)$
No. of k-tournament	3	3
p_{select}	0.5	0.5
P_c	0.3	0.3
P_m	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
$\eta_{x_{ij}}$	-	0.01
$\eta_{\phi_{ij}}$	-	0.01

서 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 [11]의 표 1에서 표 14까지를 참조).

본 논문에서는 유전 알고리즘만을 이용한 FLC 설계와 Lamarckian 상호 적응(역전파 학습 + 유전 알고리즘 진화)을 이용한 FLC 설계의 두 가지 다른 시뮬레이션을 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 4는 목적 함수의 가중치들이 모두 같을 때

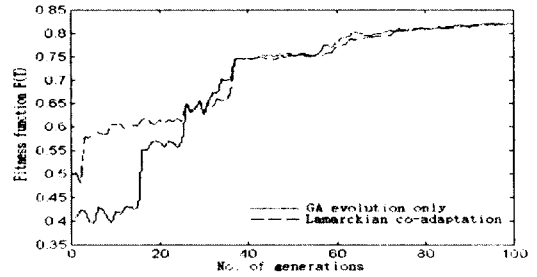


그림 4. 다른 FLC 설계 방법들의 진화 곡선

알고리즘에 의한 진화와 Lamarckian 상호 적응에 의해 얻어진 것으로, 각 세대에서 해집단 내에서 계산된 목적 함수중 가장 좋은 것을 나타낸 것이다. 그림 4는 진화 초기에 Lamarckian 상호 적응 방법이 국부적 탐색과 오차 역전파를 이용한 파라메타 갱신을 수행하기 때문에 유전 알고리즘만을 이용한 방법보다 더 나은 FLC를 찾는다는 것을 보여준다

그림 5는 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 성능들 ($0 \leq x \leq 20, 0 \leq y \leq 50$)을 각각 비교한다. 제어 성능은 $-270^\circ \leq \phi \leq 90^\circ$ 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 있다.

5. 결론

본 논문은 학습과 진화의 결합을 이용한 여러 개의 FLC를 Lamarckian 상호 적응시켜 최적의 뉴로-FLC를 얻는 새로운 설계 기법을 제안하였다. 제안한 Lamarckian 상호 적응은 일생 동안 개체가 학습에 의하여 획득된 표현형의 특성이 개체의 후손에 유전 인자 형태로 직접 전달된다는 가설에 기반한 것이다. 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 규칙들

의 개수, 근사화 오차, 그리고 평균 주행 거리에 의해 서로 비교하였다. 시뮬레이션 결과는 Lamarckian 상호 적용된 FLC가 세 가지 서로 다르게 얻어진 FLC중 모든 면에서 가장 좋은 성능들을 나타내었다. 제안한 Lamarckian 상호 적용 방법의 또 다른 장점은 단순히 목적 함수의 가중치 w 를 변경함으로써, 여러 가지 설계 목적 (예를 최소 규칙수를 갖는 FLC, 좋은 근사화 능력을 보이는 FLC, 좋은 제어 성능을 보이는 FLC, 또는 이들간의 균형된 성능을 나타내는 FLC등)을 만족시킬 수 있으므로, 여러 가지 주문형 (Application-specific) FLC를 쉽게 얻을 수 있는 환경을 제공한다는 점이다.

참고문헌

- [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-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] J. B. Lamarck, "Of the Influence of the Environment on the Activities and Habits of Animals," *Zoological Philosophy*, (Macmillan, London, 1914).
- [9] 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).
- [10] 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).
- [11] L. X. Wang and J. M. Mendel, "Generating Fuzzy Rules from Numerical Data, with Applications," *USC-SIPI Report*, 169 (1991).
- [12] 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.

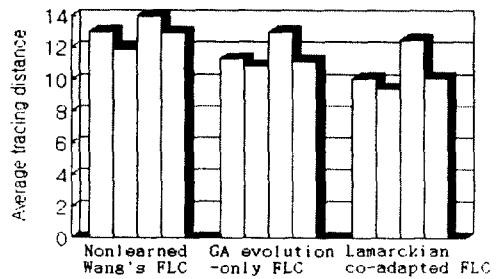


그림 5. 다른 FLC들 사이의 평균 주행 거리들의 비교