

# 신경회로망 구조를 가진 적응퍼지제어기의 구축

홍윤광

조성원

Yoon-Kwang Hong

Seongwon Cho

홍익 대학교 공과대학 전자·전기공학부

School of Electronic and Electrical Engineering, Hong Ik University

## Construction of Adaptive Fuzzy Controller with Neural Network Architecture

### Abstract

Fuzzy logic has been successfully used for nonlinear control systems. However, when the plant is complex or expert knowledge is not available, it is difficult to construct the rule bases of fuzzy systems. In this paper, we propose a new method of how to construct automatically the rule bases using fuzzy neural network. Whereas the conventional methods need the training data representing input-output relationship, the proposed algorithm utilizes the gradient of the object function for the construction of fuzzy rules and the tuning of membership functions. Experimental results with the inverted pendulum show the superiority of the proposed method in comparison to the conventional fuzzy controller.

### I. 서론

전통적인 동적 시스템 제어에는 제어 대상의 정확한 수학적 모델이 필요하다. 그러나 실제 시스템이 비선형적이고 미지인 경우에는 그 시스템을 정확히 모델링하는 것이 거의 불가능하며 또한 부하특성을 완전히 파악하기도 매우 어렵다. 각종 산업기기 및 전통적인 동적 시스템제어에서 일정이득 PID제어 방식을 널리 사용하고 있으나 이러한 제어 시스템에서는 여러 가지 형태의 기준명령과 외란에 대하여 정밀한 성능을 보장하는 이득 선정이 어려워져서 고성능 제어 특성을 기대하기 어렵다. 일반적으로 퍼지제어기는 시스템의 특성이 복잡하여 기존의 정량적인 방법으로 는 해석할 수가 없거나 얻어지는 정보가 정성적이고 부정확하고 불확실한 경우에는 전문가의 지식, 경험에 의한 제어규칙을 만들으로써 기존 제어기보다 우수한 제어결과를 나타내고 선형 시스템은 물론 비선형 특성을 지닌 시스템에 대단히 유용하다. 그러나 퍼지시스템은 전문가의 지식, 경험을 획득하기 어려운 제어대상에는 룰의 구축과 조정애 많은 시간을 소요하게 된다는 단점을 가지고 있다.

이러한 문제를 해결하기 위해 퍼지이론과 신경망 회로망을 융합하려는 연구가 활발히 진행되고 있다. 이러한 융합 연구는 각 이론의 장점을 결합하여 단점에 대해 상호보완적인 시스템을 만들려는 방향과 연산의 유사성에 기반 하여 새로운 계산 모델을 구축하려는 방향으로 진행되고 있다. 이러한 연구는 네 가지로 크게 구별된다. 첫째는 퍼지시스템의 성능 향상을 위해 신경망의 구조나 학습방법을 사용하려는 접근 방법으로 퍼지 시스템을 신경회로망으로 구현하거나 소속함수 등의 파라미터(Parameter)를 조정하기 위해 신경망회로에서 사용되는 학습방법을 변형하여 적용하는 것 등이 연구되

고 있다.

둘째는 신경회로망의 구성이나 학습을 위해 퍼지 논리를 사용하려는 접근 방법으로 신경회로망 구성을 위해 퍼지논리의 유사성을 이용하려는 방법, 신경회로망의 학습속도 향상을 위해 학습 알고리즘에 퍼지기법을 도입하려는 방법 등이 연구되고 있다.

셋째는 문제 해결을 위해 신경회로망과 퍼지기법을 각각 적합한 부분에 사용함으로써 문제를 상호 협조하에 해결하도록 하려는 접근 방법으로 전처리에서는 신경회로망을 사용하고 후처리에서는 퍼지시스템을 사용하는 방법, 학습에 의해 지식을 표현하고 있는 신경회로망 모듈로하여 이를 이용하는 퍼지시스템 등이 연구되고 있다.

마지막 접근 방법으로 신경회로망 자체가 퍼지정보를 처리할 수 있도록 하기 위해 뉴런 및 시냅스에서의 연산을 확장하려는 연구이다.

이러한 퍼지와 신경회로망의 융합을 위한 연구중에서 본 논문은 퍼지시스템의 구현을 위해서 신경회로망의 구조와 학습 능력을 도입하는 새로운 네트워크에 관한 연구이다.

본 논문에서 제안하는 학습 능력을 가진 퍼지 네트워크는 제어 대상의 입력 정보로부터 퍼지 제어의 규칙 베이스(rule base)를 자동으로 구축하는 방법을 제안한다. 그리고 제안한 시스템의 유용성을 보이기 위해서 도입된 자체제어시스템에 대해 실험하였다.

### II. 퍼지 네트워크

본 논문에서는 학습데이터없이 퍼지시스템을 자동으로 구축할 수 있는 다층구조의 퍼지 네트워크를 제안한다. 제안된 다층구조의 퍼지 네트워크는 제어 목표를

함축하고 있는 목적 함수를 근거로 학습이 진행되며, 각 상태에 따른 퍼지에 가까운 출력값을 추적해 나갈 수 있도록 설계되어 있다. 학습을 통해서 입출력 관계를 IF-THEN구조로 구성하기 때문에 제어대상에 대한 정보를 체계화할 수 있다. 제어대상이 변화하거나 제어목표가 재구성 되었을 때도 시스템 설계자에게 별다른 노력이 요구되지 않으며 학습에 의해 자동으로 새로운 상태에 적응할 수 있다는 장점이 있다.

제안된 다중구조의 퍼지 네트워크를 학습시키기 위해서는 제어목표를 잘 해석할 수 있는 목적함수이다. 하나의 제어대상에 대한 목적함수는 유일하지 않다. 목적함수를 어떻게 선정하느냐에 따라 제안된 퍼지 네트워크의 제어능력이 크게 바뀌므로 목적함수의 선정에 세심한 고려를 할 필요가 있다. 그러나 일반적인 제어대상이라면 오차자승을 최소화하는 목적함수로 충분할 것이라고 생각된다. 일반적인 목적함수는 다음 식(1)과 같다.

$$J = \frac{1}{2} (\alpha_1(e_1)^2 + \alpha_2(e_2)^2 + \dots + \alpha_n(e_n)^2) \quad (1)$$

여기서  $\alpha_i$ 는  $i$ 번째 오차항에 대한 가중치이고  $e_i$ 는  $i$ 번째 오차항을 나타낸다.

제안된 퍼지 네트워크는 크게 3가지의 동작모드를 가지고 있다. 제어규칙을 구축하는 과정과 입력 퍼지집합을 조정하는 과정, 그리고 실질적인 동작을 하기위한 퍼지 추론과정이 바로 그것이다. 이러한 퍼지 네트워크의 동작에 대해서 설명하기 위해 퍼지 네트워크의 구조에 대해 먼저 설명하겠다.

## 1. 구조

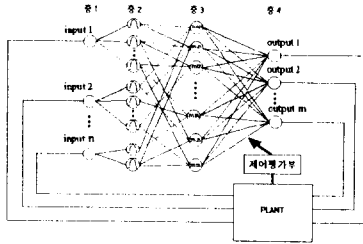


그림 1. 제안된 시스템의 전체 구조

그림 1은 본 논문에서 제안된 신경회로망의 학습 알고리즘에 기반한 퍼지시스템을 보여 주고 있다. 이 시스템은 전체적으로 4개의 층과 제어평가부를 갖고 있다. 입력층의 노드는 언어 변수를 나타내는 입력 노드들이다. 퍼지 함수 층의 노드는 각 언어 변수의 퍼지 함수들을 나타낸다. 퍼지 함수는 중심값과 분산에 의해서 나타낼 수 있는 가우시안(Gaussian) 형태의 함수를 사용하였다. 규칙층의 모든 노드들은 퍼지 규칙베이스(Fuzzy Rule Base)를 형성한다. 출력층의 노드는 출력부의 언어 변수를 나타낸다.

## 2. 퍼지추론 과정

퍼지 네트워크의 지식기반이 완성되어 있는 상태에서 퍼지 네트워크가 퍼지추론 시스템으로 동작하는 과정에 대해서 설명한다. 퍼지 네트워크에 입력값이 전달되면 첫 번째 층은 그 값을 두 번째 층으로 전달한다. 두 번째 층에서는 전달받은 값이 각 퍼지집합에 대해 어느 정도 소속도를 갖는가를 계산한다. 이렇게 계산된 값은 2번째층과 3번째층 사이의 연결에 의해서 전달되고

3번째층에서는 전달받은 값을 T-norm연산을 수행한 후 저장한다. 3번째층에 저장된 값은 제어규칙의 적합도를 나타내게 되며 3번째층과 4번째층 사이를 연결하는 연결강도와 함께 4번째층에서 수행되는 비퍼지화에 이용된다. 마지막으로 4번째층에서 제어대상으로 비퍼지화된 값을 출력한다. 퍼지추론 과정에서는 제어평가부는 동작하지 않는다.

## 3. 제어규칙의 구축 알고리즘

제안된 시스템의 제어규칙의 구축 알고리즘을 설명한다. 제어규칙의 구축이란 입력 퍼지집합의 수와 모양을 이용하여 후건부에 해당하는 퍼지싱글톤을 생성하는 것을 말한다. 먼저 퍼지 이 값은 앞에서 설명한 전체 구조에서 세 번째 층에서 네 번째 층으로 연결된 결합의 가중치에 해당된다. 이 값을 조정하기 위해서 경쟁학습 신경회로망과 유사한 학습방식을 이용한다.

현재 퍼지 네트워크의 입력값이 첫 번째층에 전달 되면 이 값에 따라 두 번째 층에서의 소속도가 결정된다. 결정된 소속도는 세 번째 층에서 T-norm연산이 수행된 후에 각 노드에 저장된다. 이때 저장된 값이 가장 큰 노드가 승자로 선택된다. 현재 입력에 대해 가장 강세를 보이는 규칙에 대해서만 학습이 진행된다는 의미이다. 여기에서 승자로 선택된 뉴런의 연결 가중치, 즉 퍼지 싱글톤에 대해서 학습을 진행시킨다.

제어규칙의 구축을 위한 과정에서 네 번째 층은 비퍼지화를 수행하지 않고 승자로 선정된 뉴런의 연결 가중치를 그대로 전달하는 역할만을 수행함으로 승자 뉴런의 연결가중치의 변화와 퍼지 네트워크의 출력의 변화는 같다.

따라서 퍼지 네트워크의 출력값에 대한 목적함수의 Gradient  $G$ 는 다음 식(2)와 같이 결정된다.

$$G(t) = \frac{\partial J}{\partial U(t)} = \frac{\partial J}{\partial u(t)_r} \quad (2)$$

이때  $J$ 는 목적함수,  $U(t)$ 는 현재 퍼지 네트워크의 출력값,  $u(t)_r$ 는 세 번째 층과 네 번째 층사이의  $i$ 번째 연결가중치이고,  $i^*$ 는 승자뉴런임을 나타낸다. 이때 현실적으로 정확한 Gradient를 얻을 수 없기 때문에 Gradient  $G$ 는 다음 식(3)와 같이 다시 정의한다.

$$G(t) \approx \frac{\Delta J}{\Delta U(t)} = \frac{\Delta J}{\Delta u(t)_r} \quad (3)$$

이렇게 얻어진 Gradient를 이용해서 선택된 노드의 연결 가중치를 학습시킨다. 이때 사용되는 학습 방정식은 다음 식(4)에서 나타냈다.

$$u(t+1)_r = u(t)_r - \eta(t)M(i^*)G(t) \quad (4)$$

여기서  $u_i$ 는 규칙층의  $i$ 번째 노드의 연결강도를 나타내고,  $i^*$ 는 여기서 선택된 승자뉴런임을 나타낸다.  $\eta(t)$ 는 학습계수로 시간에 따라 감소하는 함수를 이용한다.  $M(i)$ 은 규칙층의  $i$ 번째 노드에 저장된 값으로 규칙의 적합도를 나타낸다.

## 4. 입력 퍼지집합의 조정 알고리즘

3절의 과정을 통해서 제어규칙을 생성하고 나면 제어 성능을 향상시키기 위해서 입력 퍼지집합을 조정하는 단계가 필요하다.

입력 퍼지집합을 조정하는 이유는 시스템 설계자에 의해 초기에 주관적으로 설정된 중심값과 분산값을 어느 정도 자동으로 조정하여 목적함수에 대해서 시스

템을 조금이라도 더 최적화하기 위한 것이다. 입력 퍼지 집합을 조정하는 알고리즘은 일반적인 최급하강법을 사용한다. 조정해야 할 대상은 퍼지집합의 중심값과 분산이다. 먼저 최급하강법에 대해 설명한다. 조정해야 할 대상이  $u$ 이고 목적함수가  $J$ 라면 식(6)와 같다.

$$G(t) = \frac{\partial J}{\partial w(t)} \quad (5)$$

$$w(t+1) = w(t) - \eta(t) \cdot G(t) \quad (6)$$

입력 퍼지집합을 조정하기 위해서 퍼지 네트워크의 각 층은 퍼지 추론 과정에서의 같은 동작을 한다. 이 상태에서  $i$ 번째 언어변수의  $j$ 번째 퍼지집합의 중심값  $m_{ij}$ 를 학습시키는 학습규칙은 다음 식(8)과 같이 나타낼 수 있다.

$$G(t) = \frac{\partial J}{\partial m_{ij}(t)} = \frac{\Delta J}{\Delta m_{ij}(t)} \quad (7)$$

$$m_{ij}(t+1) = m_{ij}(t) - \eta(t) \cdot M(i, j) \cdot G(t) \quad (8)$$

여기서  $\eta$ 는 학습률이고  $M(i, j)$ 는  $i$ 번째 언어변수의  $j$ 번째 퍼지집합의 소속도를 나타내는 함수이다. 마찬가지로 분산  $\sigma_{ij}$ 는 식(10)을 이용하여 학습시킨다.

$$G(t) = \frac{\partial J}{\partial \sigma_{ij}} = \frac{\Delta J}{\Delta \sigma_{ij}(t)} \quad (9)$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - \eta(t) \cdot M(i, j) \cdot G(t) \quad (10)$$

### 5. 입력 퍼지집합의 개수 결정 알고리즘

본 논문에서는 퍼지집합의 수를 결정하기 위해서 먼저 충분히 많은 수의 퍼지 집합으로 위에서 제안한 규칙 생성 알고리즘을 이용해서 퍼지규칙을 생성한다. 생성된 퍼지규칙표에서 바로 이웃행거리 혹은 이웃열거리 유사한 퍼지 출력값을 가지고 있다면 그 행 혹은 그 열은 하나로 통합 가능한 퍼지집합이라고 할 수 있다. 이때 유사정도를 다음식(11)와 같이 판별한다. 이 값이 미리 설정한 제한값 이내로 들어오면 통합하고 줄어드는 퍼지집합의 수를 이용하여 최종적인 퍼지규칙을 다시 생성한다.

$$\beta = \frac{\sum_{i=1}^n abs(\gamma_i - \nu_i)}{n} \quad (11)$$

여기서  $n$ 은 다른 퍼지 입력 변수에 대하여 정의된 현재의 퍼지 분할의 수이고,  $abs(\cdot)$ 는 절대값을 구하는 함수,  $\gamma_i$ 와  $\nu_i$ 는 서로 이웃한 퍼지집합의  $i$ 번째 출력 퍼지집합이다.

### III. 도립진자 제어실험

도립진자는 그림 6과 같은 형태로 구성한다. 이때 제어의 목표는 도립진자에 연결된 수레에 적절한 힘을 주어서 도립진자를 수직상태로 유지하는 것이다. 이러한 도립진자는 다음 식(12)의 미분방정식으로 모델링되어진다.

시뮬레이션 환경은 IBM PC (Pentium 75MHZ)에서 Visual C++를 이용하였다. 도립진자의 무게, 도립진자의 길이, 수레의 무게는 각각 0.1kg, 0.5m, 1kg으로 설정하였다. 그리고 입력값과 출력값은 다음과 같이 설정하였고 퍼지추론을 수행하기 위해서 모든 입력력값은 -1에서 1사이 값으로 정규화 하였다.

$$\begin{aligned} \theta &: -30^\circ \sim +30^\circ \\ d\theta &: -240 \sim +240 \text{ (rpm)} \end{aligned}$$

$$F \quad : -24 \sim +24$$

$$d\theta = \frac{g \sin \theta - \cos \theta \left( \frac{F + m_c \theta^2 \sin \theta}{m_c + m_p} \right)}{\frac{4}{3} l - \frac{m_p l \cos^2 \theta}{m_c + m_p}} \quad (12)$$

- 여기서  
 F : 힘(출력)  
 g : 중력가속도  
 $m_c$  : 수레의 무게  
 $m_p$  : 도립진자의 무게  
 l : 도립진자의 길이  
 $\theta$  : 막대기의 움직인 거리

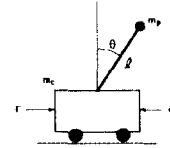


그림 2. 도립진자의 구조

실험은 전형적인 퍼지시스템과 제안된 자동구축 퍼지 네트워크에 대해서 수행하였다. 전형적인 퍼지시스템은 시스템설계자가 시행착오를 통해서 최적하게 튜닝한 퍼지시스템이다. 전형적인 퍼지시스템에서 입력 퍼지 집합과 출력퍼지집합은 그림 3과 그림 4와 같이 분할하였다. 그리고 그리고 비퍼지화 방법으로는 무게 중심법을 이용하였고, 사용된 퍼지규칙베이스는 표 1과 같다.

제안된 퍼지시스템에서는 입력공간을 먼저 충분히 많은 수인 9\*9로 분할하여 학습을 시작했고, 제안한 입력 퍼지집합의 수 결정 알고리즘으로 퍼지집합의 수를 7\*7로 결정하였다. 그리고 지식기반의 생성을 위해 사용한 목적함수는 다음 식(13)와 같다.

$$J = \frac{1}{2} [6 * \frac{1}{2^2} * (X)^2 + 1 * \frac{1}{2^2} * (\dot{X})^2] \quad (13)$$

제어규칙구축과정에서 입력값  $\theta$ 와  $d\theta$ 는 -1에서 1까지 간격 0.1로 총 400가지의 입력패턴을 만들었다. 그리고 학습률은 식(14)을 사용하였다.

$$\frac{9}{t+10} + 0.1 \quad (14)$$

그리고 총 학습횟수는 100 epoch를 수행하였다. 아래의 표 2는 퍼지 네트워크에 의해 생성된 제어 규칙을 보여주고 있다.

입력 퍼지 집합 조정과정도 마찬가지로 방법으로 입력 패턴 400개를 얻었고 학습률은 식(15)을 사용하였다.

$$\frac{9}{t+40} + 0.1 \quad (15)$$

학습횟수도 제어규칙구축과정에서와 마찬가지로 100 epoch를 수행하였다. 입력 퍼지집합과 제어 규칙 생성 결과를 그림 5와 표 2에서 보이고 있다.

본 실험에서 제안한 알고리즘이 전형적인 퍼지시스템보다 더욱 빠르게 제어목적을 수행하는 것을 확인할 수 있었다. 또 전형적인 퍼지시스템에 오버슈트가 상당히 감소하는 것도 확인할 수 있었다. 출력 결과를 그림 6과 그림 7에서 나타내고 있다.

#### IV. 결론

본 논문에서 퍼지 추론을 수행할 수 있고, 퍼지 규칙이 결핍부의 퍼지 싱글톤 출력값을 학습을 통해서 생성할 수 있는 퍼지 네트워크를 제안하였다. 제안한 시스템은 적당히 퍼지 분할 된 입력 퍼지집합과 제어 대상의 목적함수를 이용하여서 퍼지제어기를 자동으로 구현하게 된다. 제안한 시스템은 신경회로망중 하나인 단순 경쟁학습의 형태와 최급하강법이 결합된 형태로 학습을 진행하여 필요한 규칙 베이스를 구축한다. 마지막으로 제안된 시스템을 도입진자에 적용하여 보임으로써 그 유용성을 보였다.

#### 참고문헌

- [1] L. A. Zadeh, "Fuzzy Sets", Information Control, vol.8, pp. 338-353, 1965
- [2] E.H. Mamdani, "Application of Fuzzy Algorithm for Control of Simple Dynamic Plant", Inst. Electri. Eng., Vol. 121, pp. 1569-1588, 1974
- [3] Y. F. Li and C.C. Lau, "Development of fuzzy algorithms for servo systems", IEEE Control Syst. Mag., pp. 65-72, Apr. 1989.
- [4] B. Kosko, Neural Networks and Fuzzy System: A Dynamical Approach to Machine Intelligence, Eglewood Cliffs, NJ:PrenticeHall, 1992.
- [5] C. T. Lin and C.S.G. Lee, "Reinforcement Structure/Parameter Learning for Neural-Network-Based Fuzzy Logic Control System", IEEE Transaction on Fuzzy Systems, Vol. 2, No. 1, pp. 46-63, February, 1994.
- [6] K. W. Lim, H. C. Cho, H. Kang, and H. T. Jeon, "Design of the fuzzy-neural hybrid control system", Proc. of Korea-Japan Joint Conf. on Fuzzy Syst. and Eng., pp 328-334, 1992.
- [7] J. Hertz, A. Krogh and R. G. Palmer, Introduction to the Theory of Neural Computation, Addison-Wwsey, Redwood City, CA, 1991.
- [8] P. D. Wasserman, Neural Computing: Theory and Practice, Van Nostrand Reinhold, New York, 1989.
- [9] M. Figueiredo, F. Gomide, A. Rocha and R. Yager, "Comparison of Yager's Level Set Method for Fuzzy Logic Control with Mamdani's and Rasen's Method". IEEE Trans. Fuzzy System, Vol. 1, No. 2, May 1993.

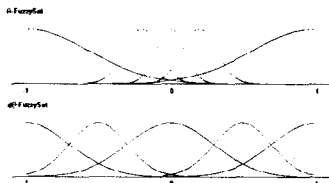


그림 3. 전형적인 퍼지 시스템의 입력 퍼지 집합

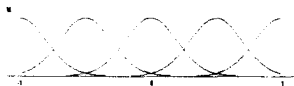


그림 4. 전형적인 퍼지 시스템의 출력 퍼지 집합

$\theta$ \ d $\theta$	NM	NS	Z	PS	PM
NM	PM	PM	PM	PS	Z
NS	PM	PM	PS	Z	NS
Z	PM	PS	Z	NS	NM
PS	PS	Z	NS	NM	NM
PM	Z	NS	NM	NM	NM

표 1. 전형적인 퍼지 시스템의 제어 규칙

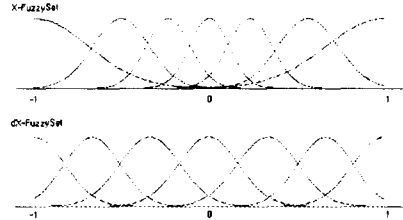


그림 5. 제안된 퍼지 퍼지네트워크에 의해서 생성된 입력 퍼지 집합

$\theta$ \ d $\theta$	1	2	3	4	5	6	7
1	1	1	1	1	1	0.706	0.505
2	1	1	1	1	0.664	0.117	-0.468
3	1	1	1	0.690	0.066	-0.477	-0.501
4	1	1	0.627	0.000	-0.628	-1	-1
5	0.558	-0.479	-0.062	0.696	-1	-1	-1
6	0.303	-0.121	-0.655	-1	-1	-1	-1
7	-0.503	-0.741	-1	-1	-1	-1	-1

표 2. 퍼지 네트워크에 의해 생성된 제어 규칙

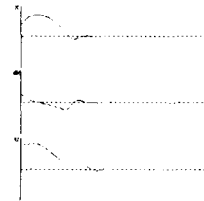


그림 6. 전형적인 퍼지 시스템에 의한 출력의 예

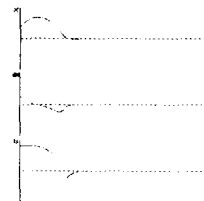


그림 7. 제안된 퍼지 네트워크에 의한 출력의 예