

퍼지 신경망을 이용한 자기학습 퍼지논리 제어기

이 우 영
한국전기연구소

A Self-Learning Fuzzy Logic Controller using Fuzzy Neural Network

Lee Woo Young
Korea Electrotechnology Research Institute

Abstract- In this paper, we proposed a design method of self-learning fuzzy logic controller using fuzzy neural network. The parameters of membership function in premise are modified by descent method and also consequent parameters by learning mechanism of animal conditioning theory. The proposed method is applied to pole balancing system in order to confirm the feasibility.

1. 서론

Zadeh에 의해 1965년 퍼지이론이 정립되고 Mamdani의 스텝 엔진에 대한 퍼지 제어기를 시작으로 많은 퍼지제어기가 제안되었으며 이는 주로 수학적 모델링이 어려운 복잡한 시스템으로 기존의 제어 방식으로는 제어하기 힘든 대상에 대해 단순히 전문가가 알고 있는 제어에 필요한 경험적 지식에 의해 양호한 결과의 제어목적을 달성할 수 있다는 것이 퍼지 제어기가 많은 관심을 받게된 이유이다.

이러한 퍼지 제어기의 특징은 제어에 대한 질적이고 부정확한 언어적 기술사항들을 이용하여 컴퓨터가 처리할 수 있는 수치적 형태로 표현할 수 있는 방법을 제시한 것이다.

그러나 퍼지 제어기를 설계하는데는 지금까지 주로 제어 대상에 대한 전문가의 지식이나 기존의 제어기 동작을 고려하여 주관적으로 소속함수나 규칙을 설정하고 이를 시행착오적인 방법으로 수정해 나가는 것에 의존하고있어 많은 시간과 노력이 필요하다는 것이 퍼지 제어기의 적용확대에 중요한 걸림돌이 되고있다. 그래서 최근에는 이러한 문제점을 극복하기위해 퍼지 제어시스템 설계의 체계화에 많은 연구가 진행 중이며 여기에는 신경회로망, 자기동조, 유전 알고리즘들이 많이 이용되고 있는 실정이다.

본 연구에서는 자기학습을 통한 퍼지 논리제어기의 설계 방법을 제안하며 사용된 퍼지 논리제어기로는 퍼지논리와 신경회로망의 개념을 결합한 퍼지-신경망으로 오차 역전파 알

고리즘에 의해 학습하고 제어기 규칙의 후진부와 전진부를 모두 학습할 수 있게 하였다. 자기학습에 있어서는 동물적 조건이론에 의한 적응 신경모델을 사용하여 퍼지논리 제어기가 학습되어지는 학습 기준망을 설정해 주었다. 그리고 역진자 제어시스템의 시뮬레이션 결과로부터 제안된 설계방법의 타당성을 확인하여 학습데이터 혹은 전문가의 지식이 없는 경우에도 실제 경험을 통한 퍼지논리제어기의 체계적 설계가능함을 보여주었다.

2. 퍼지 시스템

시스템의 입력변수가 x_1, x_2, \dots, x_m 그리고 출력변수가 y 로 표현된 단순화된 퍼지모델의 추론규칙 형태는 다음과 같이 표시될 수 있다.

$$R_i : \text{if } x_1 \text{ is } A_i \text{ and } \dots \text{ and } x_m \text{ is } B_i \text{ then } y \text{ is } w_i, \\ i=1, 2, \dots, n$$

여기서 n 은 규칙의 갯수이며 A_i, B_i 는 입력 언어변수에 해당되는 i 번째 규칙에서의 전진부 소속함수를 나타내고 w_i 는 후진부 상수 값을 나타낸다.

전진부 소속함수의 형태는 식(1)과 같은 Gaussian으로 평균값(x_a)과 편차(σ_j)에 의해 소속함수가 결정되며 이러한 규칙들의 조합에 의해 구성된 퍼지시스템을 신경망 회로 형태로 나타내면 그림 1과 같고 퍼지 시스템의 출력은 식(2)와 같다.

$$\mu_A(x_i) = \exp\left(-\frac{1}{2} \frac{(x_i - x_a)^2}{\sigma_j^2}\right) \quad \text{-----(1)}$$

$$y_a(X) = \frac{\sum_{j=1}^m w_j (\prod^n \mu_A(x_i))}{\sum_{j=1}^m (\prod^n \mu_A(x_i))} \quad \text{-----(2)}$$

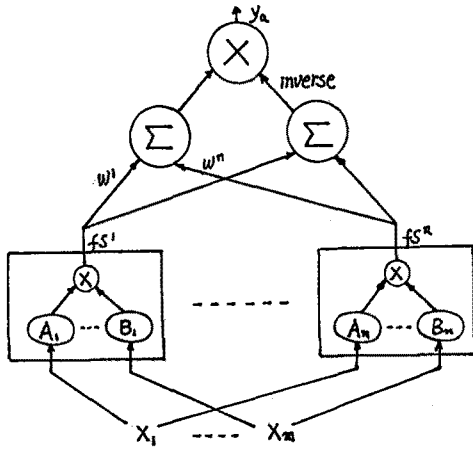


그림 1 신경망 형태로 표현된 퍼지 시스템

퍼지 시스템을 설계하기 위해서는 이를 구성하고 있는 변수들 즉 소속함수의 변수와 후진부 상수항을 결정해야 하는데 교사학습에 의해 설계를 하는 경우 실 출력값 y_a 와 원하는 출력값 y_d 의 차로 표현되는 출력오차 $e (=1/2(y_a - y_d)^2)$ 의 각 설계변수에 대한 기울기를 사용하여 식(3)-(5)와 같이 각 변수들을 샘플 데이터를 사용한 학습을 통해 수정해 나감으로 최소의 출력오차 특성을 지닌 퍼지 시스템으로 구성하게 된다[3].

$$w^j(k+1) = w^j(k) - \alpha \frac{\partial e}{\partial w^j} \Big|_{w^j=w^j(k)} \\ = w^j(k) - \alpha \frac{y_a - y_d}{b} fs^j \quad \text{---(3)}$$

$$x_a^j(k+1) = x_a^j(k) - \alpha \frac{\partial e}{\partial x_a^j} \Big|_{x_a^j=x_a^j(k)} \\ = x_a^j(k) - \alpha \frac{y_a - y_d}{b} (w^j - y_d) fs^j \frac{xp_i - x_a^j(k)}{\sigma_j^2(k)} \quad \text{---(4)}$$

$$\sigma_j(k+1) = \sigma_j(k) - \alpha \frac{\partial e}{\partial \sigma_j} \Big|_{\sigma_j=\sigma_j(k)} \\ = \sigma_j(k) - \alpha \frac{y_a - y_d}{b} (w^j - y_d) fs^j \frac{(xp_i - x_a^j(k))^2}{\sigma_j^3(k)} \quad \text{---(5)}$$

그러나 이는 출력오차의 크기를 알기 위해 샘플 데이터가 마련되어 있어야 하는 교사 학습 알고리즘으로 학습 데이터가 없는 경우에는 적용할 수 없다.

이를 위해 다음에는 동물조건이론을 적용하여 이러한 경우에 출력오차에 해당되는 설계변수 학습 수정량을 설정해주는 방법에 대해 제안한다.

3. 설계변수 수정에 필요한 수정 기준량 설정

본 제어기의 제어대상으로는 학습데이터가 없고 제어과정에서 제어가 실패했음을 알려주는 신호가 제어기 성능 검증에서 유일하게 존재하는 시스템이기 때문에 임의의 초기상태에서 실패하기까지 취해졌던 일련의 제어법칙들에 대해 실패에 대한 책임을 부여(credit assignment)하고 이를 토대로 제어 법칙들을 수정해 나갈 수 있도록 하기 위한 평가함수가 설정되어야 한다. 이는 Pavlovian condition 이론을 위한 실험에서 개가 종소리에 대해 침이 분비되는 관계를 결정해 가는 함수로 어떤 두 입력상태들에 대한 평가함수 값의 상대적 평가는 어느쪽이 실패하기 쉬운가를 구별할 수 있게한다.

평가함수로는 연상비평뉴런(associative critic neuron)에 의한 식(6)-(9)를 사용하여 내부강화신호(internal reinforcement) \hat{r} 를 설정하고 이로부터 규칙 후진부 제어량 w_i 를 수정하게 한다.

$$v_i(k+1) = v_i(k) + \beta \hat{r}(k) \bar{x}_i(k) \quad \text{---(6)}$$

$$\hat{r}(k) = r(k) + \tau p(k) - p(k-1) \quad \text{---(7)}$$

$$p(k) = G(\sum v_i(k) x_i(k)) \quad \text{---(8)}$$

$$\bar{x}_i(k+1) = \delta \bar{x}_i(k) + (1 - \delta) |v_i(k) x_i(k)| \quad \text{---(9)}$$

$$e_i(k+1) = \lambda e_i(k) + (1 - \lambda) F(k) x_i(k) \quad \text{---(10)}$$

$$w_i(k+1) = w_i(k) + \alpha_i \hat{r}(k) e_i(k) \quad \text{---(11)}$$

식(3)과 (11)에서는 w_i 의 학습에 대해 나타내고 있는 식이며 식(3)는 학습 데이터에 의한 오차량으로 식(11)은 내부강화신호량에 의해 w_i 의 수정량이 표현되고 있는데

$$-\alpha \frac{\partial e}{\partial w^j} \Big|_{w^j=w^j(k)} \implies \alpha \hat{r}(k) e_i(k) \quad \text{---(12)}$$

로 대체하면 식(4), (5)도 마찬가지로 내부강화신호량에 의해 표현 되어질 수 있어 학습 데이터 없이 규칙의 전, 후진부 설계변수를 수정할 수 있다.

4. 시뮬레이션

시뮬레이션에서 제어대상으로 사용한 것은 제어 알고리즘의 성능평가에 많이 사용되고있는 역진자 시스템으로 이의 동적상태를 나타내는 식으로는 동차와 마대의 마찰계수를 무시한 식(13), (14)를 사용하였다.

$$\ddot{\theta} = \frac{g \sin \theta + \cos \theta \left[\frac{-F_t - m l \dot{\theta}^2 \sin \theta}{M_c + m} \right]}{1 \left[\frac{4}{3} - \frac{m \cos^2 \theta}{M_c + m} \right]} \quad \text{---(13)}$$

$$\ddot{x} = \frac{F_t + ml [\theta^2 \sin \theta - \ddot{\theta} \cos \theta]}{m_c + m} \quad \text{---(14)}$$

여기서 θ 는 막대의 각도(rad), $\dot{\theta}$ 막대의 각속도(rad/sec), g 는 중력가속도, m 은 막대중량, m_c 는 동차의 중량, F_t 는 동차에 인가되는 힘, x 는 동차의 위치, \dot{x} 는 동차의 속도를 나타내며 학습시 사용된 각 학습률들은 $\alpha=0.9$, $\alpha_1=0.9$, $\beta=0.5$, $\tau=0.95$, $\delta=0.9$, $\lambda=0.8$ 으로하고 수직선율 기준으로 각도 θ 가 12 이상 기울어지면 실패한 것으로 간주하여 이 범위내의 임의의 상태에서 다시 시작한다.

퍼지 제어시스템에 사용되는 규칙의 초기 형태는 θ 와 $\dot{\theta}$ 를 각 다섯개의 소속함수로 구성하게하고 소속함수의 평균값이 각 변수의 전체영역을 중간적으로 나누도록 설정하며 후건부 상수 값은 "0"으로 학습전에는 대상 시스템의 제어를 위한 아무런 지식도 필요하지 않도록 한다.

그림 2는 역진자 제어를 위한 시뮬레이션 흐름도를 나타낸 것으로 계산된 제어량은 제한된 크기 내의 임의의 방향으로 이산적 시간간격동안 일정크기의 힘을 동차에 인가하도록하며 샘플링 간격은 10ms로 한다.

그림 3은 학습후 초기값 $\theta=0.1(\text{rad})$ 에서의 제어결과를 나타내었으며 이때 소속함수의 각 변수들은 표 1과 같다

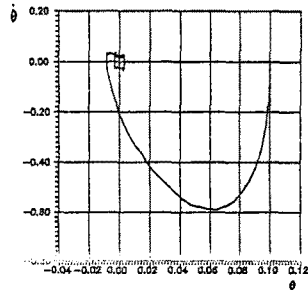


그림 3 설계된 퍼지 제어기의 제어결과

표 1 소속함수의 각 변수값

	소속 함수	초기치		학습후	
		평균값	편차	평균값	편차
각도	1	0	8.0	0.0	0.0
	2	18	8.0	18.0	18.0
	3	36	8.0	36.4	35.8
	4	54	8.0	54.0	54.1
	5	72	8.0	71.0	71.1
각속도	1	0	8.0	7.9	8.0
	2	18	8.0	7.9	8.0
	3	36	8.0	7.1	7.8
	4	54	8.0	7.9	7.9
	5	72	8.0	8.0	7.7

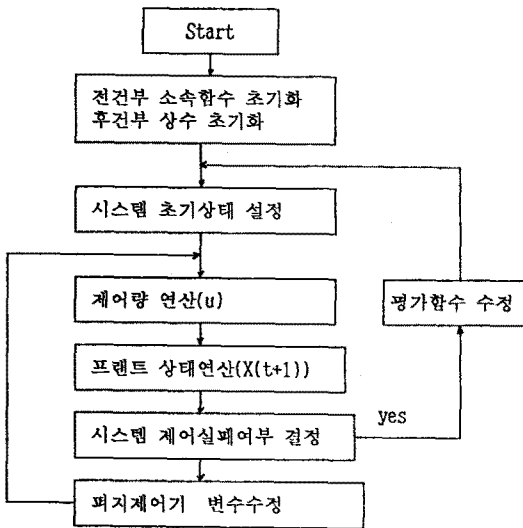


그림 2 시뮬레이션 흐름도

5. 결 론

제어대상에 대한 사전지식과 학습데이터 없이 반복적으로 수행되는 경험을 통해 퍼지 제어기의 전, 후건부 변수를 동시에 설계하는 방법을 제안하고 역진자 시스템에 대해 제안한 방식으로 퍼지제어기를 구성하였다.

초기에 등간격으로 설정한 소속함수들이 설계후 많은 변화를 보이지 않는 것은 역진자 시스템의 제어함수가 단순한 단조함수로 구성되어 소속함수에는 큰 영향을 받지 않기 때문이다.

참 고 문 헌

[1] C.C.Lee, "A self-learning rule based controller employing approximate reasoning and neural net concepts", International Journal of Intelligent System, Vol.6, p71-93, 1991

[2] A.G.Barto, R.S.Sutton and C.W.Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems", IEEE Trans. on System, Man, and Cybernetics, Vol.SMC-13, No.5, p834-846, 1983

[3] Li-Xin Wang and J.M.Mendel, "Back-propagation fuzzy system as nonlinear dynamic system identifier", proc. IEEE International Conference on fuzzy system and neural network, 1992