

뉴럴-퍼지 융합을 이용한 퍼지 제어 규칙의 자동생성에 관한 연구

(Auto Generation of Fuzzy Control Rule using Neural-Fuzzy Fusion)

林 光 宇*, 金 容 虎*, 姜 勳**, 全 洪 兌*

(Kwang Woo Lim, Yong Ho Kim, Hoon Kang, and Hong Tae Jeon)

要 約

본 논문에서는 퍼지 논리와 신경회로망의 장점들을 결합한 퍼지-신경망(fuzzy-neural network; FNN)을 제안한다. FNN의 기본적인 구조는 퍼지 제어기의 규칙베이스와 추론의 과정을 신경회로망을 이용하여 구현한 것이며 퍼지 제어기의 중요한 매개변수들은 신경회로망의 가중치들로 표현된다. 그리고, FNN과 일반적인 선형제어기인 P-제어기를 병렬로 연결한 새로운 형태의 퍼지-뉴럴 하이브리드 제어기를 설계한다. 제안한 구조에서 FNN의 규칙베이스는 오차 역전파 학습 알고리즘에 의하여 자동생성된다. 또한, FNN과 병렬연결한 P-제어기는 FNN의 규칙베이스가 완전히 생성되기 전인 초기상태에서 제어기의 안정성을 보장 할 것이다. 본 논문에서 제안한 방법의 효율성은 2자유도 평면 로봇 매니퓰레이터를 이용한 컴퓨터 모의 실험을 통해 입증된다.

Abstract

In this paper we propose a fuzzy-neural network(FNN) which includes both advantages of the fuzzy logic and the neural network. The basic idea of the FNN is to realize the fuzzy rule-base and the process of reasoning by neural network and to make the corresponding parameters be expressed by the connection weights of neural network. After constructing the FNN, a novel controller consisting of a conventional P-controller and a FNN is explained. In this control scheme, the rule-base of a FNN are automatically generated by error back-propagation algorithm. Also the parallel connection of the P-controller and the FNN can guarantee the stability of a plant at initial stage before the rules are completely created. Finally the effectiveness of the proposed strategy will be verified by computer simulations using a 2 degree of freedom robot manipulator.

*正會員, 中央大學校 電子工學科

(Dept. of Elec. Eng., Chungang Univ.)

**正會員, 中央大學校 制御計測工學科

(Dept. of Control and Instrumentation Eng.,
Chungang Univ.)

接受日字: 1991년 12월 13일

(※ 이 논문은 1992년도 교육부 지원 한국학술진흥재단의 자유공모 과제 학술 연구조성비에 의하여 연구되었음.)

I. 서 론

퍼지 논리 제어기(fuzzy logic controller)는 전문가의 경험적 지식에 기초한 언어적 제어전략(control strategy)을 기계적인 시스템에 구현할 수 있는 수단을 제공한다. 퍼지 제어는 규칙베이스(rule-base) 제어의 특수한 형태라고 볼 수 있으며, 일반적으로 퍼지 제어기는 퍼지화기(fuzzifier), 규칙베이스(rule-base), 퍼

지추론기(fuzzy inference engine) 그리고 비퍼지화기(defuzzifier)의 네가지 중요한 요소로 구성되어 있다.

퍼지화기는 입력 데이터를 적당한 퍼지집합으로 변환하는 기능을 갖고 있으며 규칙베이스는 전문가의 경험등을 "IF~THEN" 형의 문장으로 구성된 지식베이스이다. 그리고 퍼지추론기는 규칙베이스의 규칙을 이용, 퍼지입력에 대해 퍼지 출력을 추론하는 기능을 가지며, 비퍼지화기는 추론기에서 추론한 결과를 명확한 값(crisp value)으로 변환 시키는 기능을 갖는다. 퍼지 제어를 설계하는데 있어서 제어대상의 정확한 수학적 모델링은 필요하지않고, 단지 제어대상에 대한 경험적 지식만으로 퍼지 제어 규칙을 만들 수 있으며 그럼에도 불구하고 기존의 제어기에 비해 훌륭한 성능을 얻을 수 있다. 퍼지 논리 제어기의 실제적인 응용은 이미 많은 분야에서 보고되어 있다.^{[1]-[3]}

그러나, 위에서 기술한 단순한 퍼지 제어 알고리즘은 다음과 같은 관점에서 몇가지 문제점을 갖는다. 첫째, 이와같은 구조에서 퍼지 제어기의 규칙베이스를 전문가의 경험에 의지하여 임의로 구성하는 것은 매우 어려운 일이다. 둘째, 한번 결정된 규칙베이스는 고정되어 있으므로 제어환경의 변화에 대하여 적절히 대응하지 못한다. 더욱이, 로봇트와같이 비선형의 동적특성을 갖고 있는 제어대상의 경우 그 어려움은 더욱 가중된다.

따라서, 이러한 문제점들을 해결하기 위해서는 규칙베이스의 자동생성과 더불어 퍼지 제어기에 적응성을 부여해야 할 필요성이 제기된다. 위와 같은 문제점들을 극복할 수 있는 방법중 하나는 퍼지 논리와 신경회로망을 결합하는 것이다. 신경회로망은 인간의 뇌를 모델로한 인공뉴런의 소자들을 상호 연결한 시스템으로써 대규모의 병렬분산처리 특성과 학습능력을 가지고 있다. 최근 널리 응용되는 신경회로망 모델은 다층신경망(multilayer neural network), 홉필드신경망(Hopfield neural network), 볼츠만머신(Boltzmann machine)등을 들수있으며 제어기 구성에는 다층신경망 모델을 널리 사용하고 있다.^{[4]-[6]}

본 논문에서는 퍼지 논리와 다층신경망의 장점들을 결합한 새로운 퍼지-신경망(fuzzy-neural network; FNN)을 제안한다. FNN의 기본적인 구성은 퍼지제어기의 규칙베이스와 추론의 과정을 신경회로망을 이용하여 구현한 것이다. 이때, 퍼지 제어기의 중요한 매개변수(parameter)들은 신경회로망의 가중치(weight)들로 표현된다. 아울러, 본 논문에서는 일반적인 P-제어기와 제안한 FNN으로 구성되어 있는 새로운 형태의 퍼지-뉴럴 하이브리드 제어기를 제안한다. 제안한 구조에서 FNN의 규칙베이스는 오차 역전파(error back propagation) 학습 알고리즘에 의하여 자동생성된다. 그리고,

FNN과 병렬연결한 P-제어기는 FNN의 규칙베이스가 완전히 생성되기 전인 초기상태에서 제어기의 안정성을 보장 할 것이다.

한편, 본 논문의 구성은 다음과 같다. 본 논문에서 제안한 FNN의 구조가 II장에서 설명될 것이다. III장에서는 퍼지-뉴럴 하이브리드 제어기가 제안된다. IV장에서는 제안한 제어구조의 효율성을 입증하기 위하여 컴퓨터 모의 실험을 행한다. 마지막으로 V장에서는 결론을 논한다.

II. 퍼지-신경망(Fuzzy-Neural Network; FNN)

1. 퍼지-신경망의 구조

일반적인 퍼지 논리 제어기의 구조는 그림 1과 같으며 각 구성요소의 기능들은 표 1에 정리되어 있다.

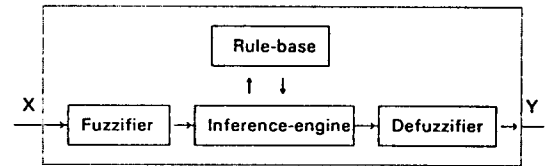


그림 1. 퍼지 논리 제어기의 구조

Fig. 1. The structure of a fuzzy logic controller.

표 1. 각 구성요소의 기능들

Table 1. The functions of basic component.

Componnt	Functions	Method
Fuzzifier	<ul style="list-style-type: none"> measure the value of input variable scale mapping convert input data into corresponding fuzzy sets 	<ul style="list-style-type: none"> fuzzy singleton fuzzy number isosceles triangle
Rule-Base	<ul style="list-style-type: none"> provide control goals and strategy of the expert by means of a set of lingistic control rules 	
Inference engine	<ul style="list-style-type: none"> Simulating human decision making based on fuzzy inference logic 	<ul style="list-style-type: none"> direct indirect hybrid
Defuzzifier	<ul style="list-style-type: none"> convert an inferred fuzzy sets into a crisp control action scaling mapping 	<ul style="list-style-type: none"> max criterion Mean of Max. Method(MOM) Center of Area Method(COA)

퍼지 논리 제어기의 설계는 인간의 경험과 지식에 기초한 추론과정을 모델링(modeling)하는 것으로 간주할 수 있다. 이와 같은 관점에서 최적의 규칙베이스 설계와 추론방법의 결정은 대단히 중요하다. 퍼지 논리 제어기의 규칙베이스는 일반적으로 다음과 같은 형태를 갖는다.

IF (a set of conditions are satisfied)
 THEN (a set of consequences can be inferred)

일반적으로 퍼지규칙의 후건부는 하나의 퍼지집합이다. 반면에 본 연구에서 사용한 퍼지규칙의 후건부는 입력변수들의 선형함수로 이루어져있다. 이 구조는 Takaki와 Sugeno가 제안한 바 있다.^[7] 여기서 i번째 규칙은 다음과 같이 표현된다.

$$R_i : \text{If } X_1 \text{ is } A_1^i \text{ and } \dots X_n \text{ is } A_n^i, \\ \text{Then } Y^i = c_0^i + c_1^i X_1 + \dots + c_n^i X_n \quad (1)$$

여기서, R^i 는 i번째 규칙, A_j^i 는 언어 변수, Y^i 는 R^i 로 표현되는 i번째 규칙의 출력이며 c_0^i, \dots, c_n^i 는 상수이다. 식 (1)과 같은 퍼지규칙과 다음에 설명될 추론과정을 신경망으로 구현한 것이 그림 2의 FNN이다. 이에대한 구체적인 설명은 다음과 같다.

1) 전건부(Premise part)

그림 2에서 (A)~(C)층(layer)은 FNN의 전건부(premise part)이다. 그림 2의 경우 FNN의 전건부는 2

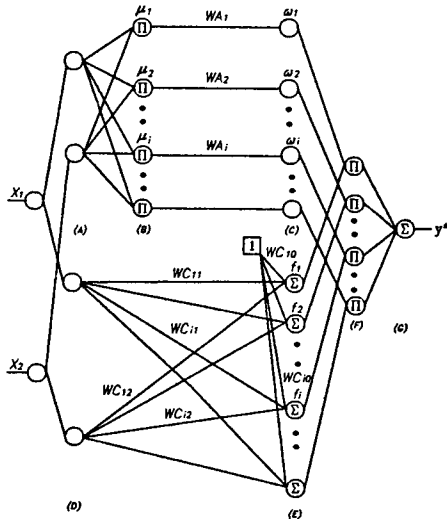
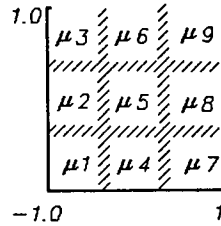
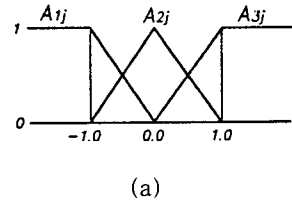


그림 2. 퍼지-신경망의 구조
 Fig. 2. The structure of a fuzzy-neural network (FNN).



(a)

(b)

(c)

(b)

(c)

그림 3. FNN의 전건부

- (a) 전건부의 소속함수
- (b) 전건부의 퍼지부공간
- (c) 퍼지변수의 소속정도

Fig. 3. Premise part of FNN.

- (a) Membership functions of premise,
- (b) Fuzzy subspaces of premise,
- (c) Degree of membership of a fuzzy variable.

개의 입력 X_1, X_2 그리고 3개의 소속함수(그림 3(a)참조)로 이루어져 있다. 그리고, 소속함수의 전체 집합(universe of discourse)은 정규화 되어있다(그림 3(a)참조). 또한, 그림 3(a), (b)에서 보는 바와 같이 2개의 입력과 3개의 소속함수는 입력공간을 $9(=3 \times 3)$ 개의 퍼지부공간(fuzzy subspace)으로 분할 한다. 그림 2에서 (B)층과 (C)층을 연결하는 가중치(connection weight) WA_i 는 전건부의 규칙또는 퍼지 부공간의 중요도를 의미한다. 한편, 그림 3(c)에서 보는 바와같이 $A_{ij}(x_j)$ 는 입력이 x_j 인 경우 퍼지변수 A_{ij} 의 소속정도를 의미한다. 여기서, A_{ij} 는 언어변수이며 예컨대, "Positive", "Negative" 또는 "Zero" 등으로 표시된다. 퍼지규칙의 진리값(truth value)은 (B), (C)층의 출력으로 부터 결정된다. 전건부 퍼지규칙의 각부공간의 진리값은 식 (2), (3)과 같이 주어진다.

$$\begin{aligned} \mu_1 &= A_{11}(x_1) \cdot A_{12}(x_2) & \mu_6 &= A_{21}(x_1) \cdot A_{32}(x_2) \\ \mu_2 &= A_{11}(x_1) \cdot A_{22}(x_2) & \mu_7 &= A_{31}(x_1) \cdot A_{12}(x_2) \\ \mu_3 &= A_{11}(x_1) \cdot A_{32}(x_2) & \mu_8 &= A_{31}(x_1) \cdot A_{22}(x_2) \\ \mu_4 &= A_{21}(x_1) \cdot A_{12}(x_2) & \mu_9 &= A_{31}(x_1) \cdot A_{32}(x_2) \\ \mu_5 &= A_{21}(x_1) \cdot A_{22}(x_2) & & \end{aligned} \quad (2)$$

$$\omega_i = WA_i \mu_i \quad (3)$$

여기서, (i=1, ..., 9)

2) 후건부(Consequence part)

FNN의 후건부는 식 (4)와 같이 입력변수들의 선형함수로 표현되며 그림 2에서 (D)~(G)-층에 해당한다. 그림 2의 FNN은 2개의 입력변수와 1개의 출력을 가지고 있으므로 규칙베이스는 식 (5)와 같이 표현된다. 또한, 이 경우 추론 및 비퍼지화의 과정은 식 (6)과 같이 표현된다. 여기서 y^* 는 FNN의 출력인 명확한 값(crisp value)이다.

$$f_i(x_1, x_2) = WC_{i0} + WC_{i1}x_1 + WC_{i2}x_2 \quad (4)$$

WC_{ij} (j=0, 1, 2) : 상수(constant)

$$R^i : \text{If } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ Then } y=f_i(x_1, x_2) \quad (5)$$

여기서, (i=1, ..., n)

$$y^* = \sum_{i=1}^n \mu_i WA_i f_i(x_1, x_2) = \sum_{i=1}^n \omega_i f_i(x_1, x_2) \quad (6)$$

2. 학습 방법(Learning method)

본 절에서는 제어목적에 맞는 규칙베이스를 찾기 위하여 FNN의 가중치들을 학습시키는 방법을 설명한다. FNN을 학습시키는 방법은 역전파 학습 알고리즘에 기초한 것이다.^[8] 역전파 학습 알고리즘을 적용하기 위해서는 학습을 위한 입·출력 데이터가 필요하며 출력층에서의 적절한 에너지함수를 정의 해야한다. 입·출력 데이터는 FNN의 입력이 $x_1(t)$, $x_2(t)$ 일 경우 실제 출력 $y^*(t)$, 이때의 원하는 출력 $d(t)$ 로 구성한다. 또한, FNN의 출력층(G)에서의 에너지 함수를 식 (7)과 같이 정의 한다.

$$E = 1/2 (d - y^*)^2 \quad (7)$$

최급강하법(gradient descent rule)을 이용하여 식 (7)의 에너지 함수를 최소화 할 경우 가중치의 증분은 다음과 같이 결정된다.

$$\begin{aligned} \Delta WA_i &= -\eta \frac{\partial E}{\partial WA_i} = -\eta \frac{\partial E}{\partial \omega_i} \cdot \frac{\partial \omega_i}{\partial WA_i} \\ &= -\eta \frac{\partial E}{\partial \omega_i} \cdot \mu_i \end{aligned} \quad (8)$$

여기서,

$$\begin{aligned} \delta_i &\equiv -\frac{\partial E}{\partial \omega_i} = -\frac{\partial E}{\partial y^*} \cdot \frac{\partial y^*}{\partial \omega_i} \\ &= (d - y^*)(WC_{i0} + WC_{i1}x_1 + WC_{i2}x_2) \end{aligned} \quad (9)$$

따라서,

$$\Delta WA_i = \eta \delta_i \mu_i \quad (10)$$

$$\begin{aligned} \Delta WC_{i0} &= -\eta \frac{\partial E}{\partial WC_{i0}} = -\eta \frac{\partial E}{\partial f_i} \cdot \frac{\partial f_i}{\partial WC_{i0}} \\ &= -\eta \frac{\partial E}{\partial f_i} = -\eta \frac{\partial E}{\partial y^*} \cdot \frac{\partial y^*}{\partial f_i} \\ &= \eta (d - y^*) \omega_i \end{aligned} \quad (11)$$

$$\begin{aligned} \Delta WC_{i1} &= -\eta \frac{\partial E}{\partial WC_{i1}} = -\eta \frac{\partial E}{\partial y^*} \cdot \frac{\partial y^*}{\partial WC_{i1}} \\ &= \eta (d - y^*) \cdot \frac{\partial}{\partial WC_{i1}} \left[\sum_{i=1}^n \omega_i f_i(x_1, x_2) \right] \\ &= \eta (d - y^*) \omega_i x_1 \end{aligned} \quad (12)$$

$$\begin{aligned} \Delta WC_{i2} &= -\eta \frac{\partial E}{\partial WC_{i2}} = -\eta \frac{\partial E}{\partial y^*} \cdot \frac{\partial y^*}{\partial WC_{i2}} \\ &= \eta (d - y^*) \cdot \frac{\partial}{\partial WC_{i2}} \left[\sum_{i=1}^n \omega_i f_i(x_1, x_2) \right] \\ &= \eta (d - y^*) \omega_i x_2 \end{aligned} \quad (13)$$

여기서, η : 학습율(learning rate)

따라서, FNN의 가중치는 식 (14), (15)와 같이 갱신(update)된다.

$$WA_{i, \text{new}} = WA_{i, \text{old}} + \Delta WA_i \quad (14)$$

$$WC_{j, \text{new}} = WC_{j, \text{old}} + \Delta WC_j \quad (15)$$

여기서, j=0, 1, 2

III. 퍼지-뉴럴 하이브리드 제어시스템

본 장에서는 FNN을 이용한 퍼지-뉴럴 하이브리드 제어시스템을 제안한다.(그림 4참조)

그림 4에서 보는 바와 같이 본 구조에서는 로봇트를 제어하기 위하여 FNN과 일반적인 P-제어기를 병렬로 연결하였다. 그러나, 여기에는 식 (7)의 에너지함수를 정의

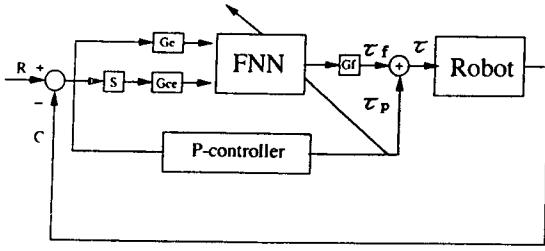


그림 4. 퍼지-뉴럴 하이브리드 제어시스템
Fig. 4. Fuzzy-neural hybrid control system.

해야하는 문제가 남아 있다. 이 문제는 식 (16)과 같이 토오크 오차를 에너지함수로 정의함으로써 해결한다.

$$E = 1/2 (\tau_p)^2 \tag{16}$$

토오크 오차가 영(zero)에 수렴하게되면 에너지 함수 역시 영(zero)에 수렴하게되므로 식 (16)은 적절한 에너지 함수라고 볼 수 있다. 학습기간동안 FNN의 토오크와 P-제어기의 토오크는 식 (17)과 같이 더하여져서 로봇의 입력으로 공급된다.

$$\tau = \tau_f + \tau_p \tag{17}$$

학습이 완료되면 FNN의 규칙베이스는 τ_p 의 값을 최소화 할 수 있도록 완전히 생성된다. 그리고, 로봇의 입력은 오직 FNN에 의해서만 주어진다. 한편, FNN의 규칙베이스가 완전히 생성되기전의 과도상태에서는 P-제어기가 전체 시스템을 안정화 할 것이다.

IV. 컴퓨터 모의실험

컴퓨터 모의실험에 사용된 2자유도 평면 로봇 매니플레이터는 그림 5와 같다.

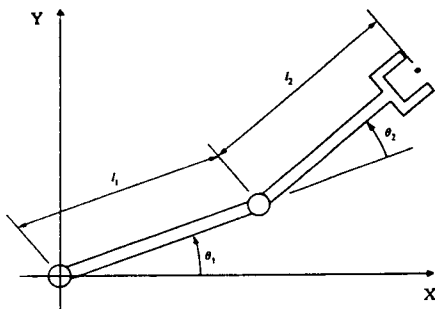


그림 5. 2자유도 평면 로봇 매니플레이터
Fig. 5. 2DOF planar robot manipulator.

표 2는 모의 실험에 사용된 FNN의 환산계수와 학습율을 나타내고 있다. P-제어기의 이득은 100.0이며 샘플링 시간은 10.0(msec)이다.

표 2. FNN의 매개변수 값
Table 2. The parameters used by FNN.

Parameter	Value	Content
Ge	5.0	scaling factor of premise
Gce	0.5	scaling factor of premise
Gf	100.0	scaling factor of cosequence
η	0.001	learning rate

1. 모의실험(1)

본 모의 실험에서는 로봇 매니플레이터가 식 (18)에 주어진 궤적을 추종하도록 하였다. 표 3은 모의 실험에 사용된 로봇링크의 길이와 질량을 보여주고 있다. 표 4는 학습하기전의 FNN의 초기가중치이다. 학습이 이루어지기전 상태에서의 전건부의 규칙 또는 퍼지 부공간의 중요도는 모두 동등하므로 W_{A_i} 는 모두 1.0으로 두었고, 후건부의 규칙은 전혀 없는 상태이므로 $W_{C_{i0}}$, $W_{C_{i1}}$, $W_{C_{i2}}$ 는 -0.01~+0.01사이의 값중에서 임의로 선택하여 초기화 하였다.

$$X = 0.3 + 0.25\cos[\theta(t)] \tag{18a}$$

$$Y = 0.3 + 0.15\sin[\theta(t)] \tag{18b}$$

모의실험 결과가 그림 6~ 그림 13에 주어져 있다. 학습하는 동안의 로봇의 실제경로와 원하는 경로가 그림 6, 7에 주어져 있다. 그림 6에서 보는 바와 같이 학습시 처음에는 약간의 오차가 존재하나 학습을 거듭함에 따라 오차는 서서히 줄어들어다 그리고 그림 7과 같이 충분한 학습을 거친후에 로봇은 거의 완벽하게 주어진 궤적을 추종함을 볼 수 있다.

그림 8~그림 11은 학습하는 동안 로봇의 각각의 조

표 3. 자유도 로봇의 링크 매개변수
Table 3. Link parameters of 2 DOF robot manipulator.

	Link 1	Link 2
Length	0.53m	0.47m
Mass	2.5kg	2.0kg

표 4A. 조인트 1에 대한 FNN의 초기 가중치

Table 4A. Initial weight of FNN for joint 1.

i	WA _i	WC ₁₀	WC ₁₁	WC ₁₂
1	1.000	0.006	-0.006	0.009
2	1.000	-0.003	0.007	-0.004
3	1.000	-0.006	0.002	-0.000
4	1.000	0.008	-0.001	-0.005
5	1.000	-0.009	-0.004	0.002
6	1.000	0.007	-0.010	0.005
7	1.000	-0.006	-0.007	-0.006
8	1.000	-0.002	0.003	-0.005
9	1.000	0.003	0.005	0.005

표 4B. 조인트 2에 대한 FNN의 초기 가중치

Table 4B. Initial weight of FNN for joint 2.

i	WA _i	WC ₁₀	WC ₁₁	WC ₁₂
1	1.000	-0.004	-0.010	-0.003
2	1.000	-0.003	-0.003	-0.002
3	1.000	0.008	-0.005	0.002
4	1.000	-0.001	0.001	0.004
5	1.000	0.004	0.005	-0.006
6	1.000	-0.008	-0.003	-0.007
7	1.000	-0.007	0.003	0.010
8	1.000	0.001	0.009	0.003
9	1.000	0.002	0.009	-0.008

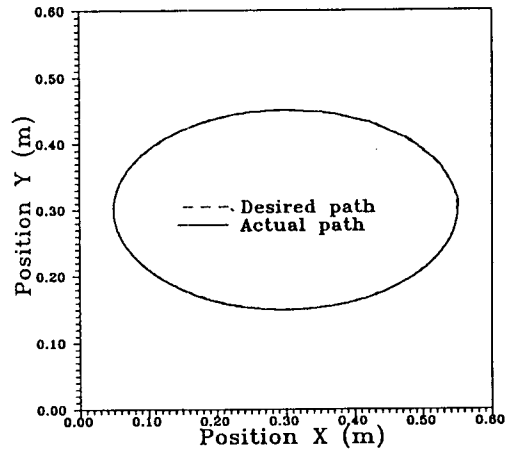


그림 7. 20번째 학습시 로봇의 경로

Fig. 7. Desired and actual path of a robot at the 20th learning trial.

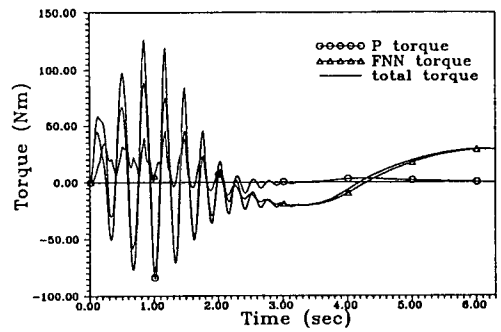


그림 8. 첫번째 학습시 조인트 1에 가해지는 토크의 변화

Fig. 8. Torque applied to joint 1 during the 1st learning trial.

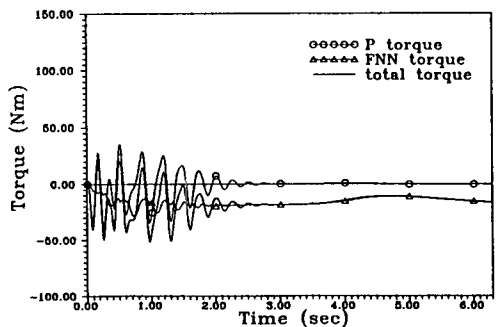


그림 9. 첫번째 학습시 조인트 2에 가해지는 토크의 변화

Fig. 9. Torque applied to joint 2 during the 1st learning trial.

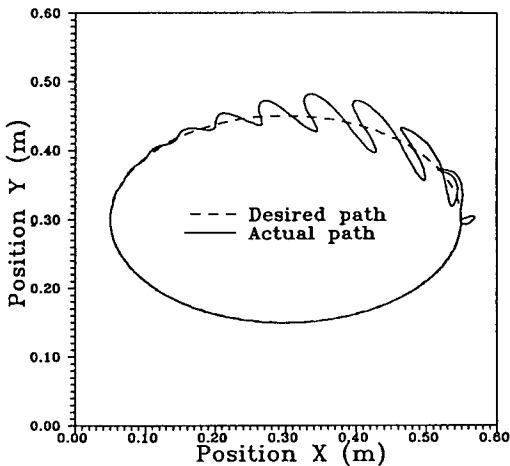


그림 6. 첫번째 학습시 로봇의 경로

Fig. 6. Desired and actual path of a robot at the 1st learning trial.

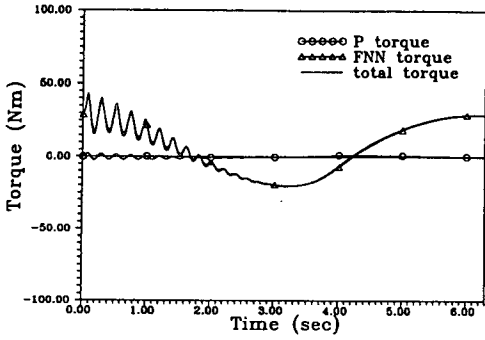


그림 10. 20번째 학습시 조인트 1에 가해지는 토크의 변화
 Fig. 10. Torque applied to joint 1 during the 20th learning trial.

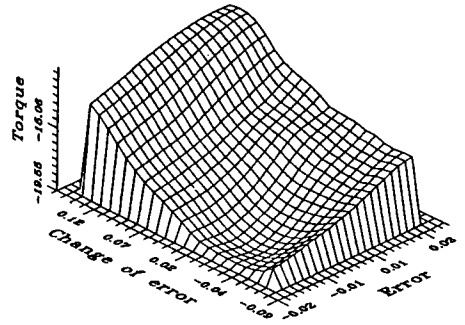


그림 13. 20번째 학습시 조인트 2의 제어평면
 Fig. 13. Control surface of joint 2 during the 20th learning trial.

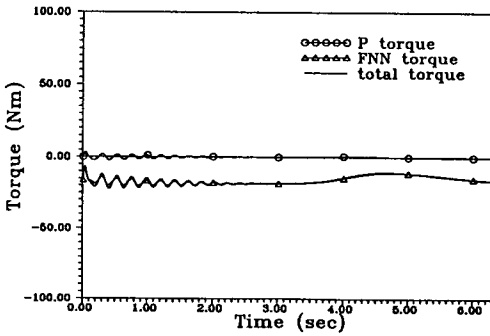


그림 11. 20번째 학습시 조인트 2에 가해지는 토크의 변화
 Fig. 11. Torque applied to joint 2 during the 20th learning trial.

표 5A. 20회 학습후 조인트 1에 대한 FNN의 가중치

Table 5A. Identified weight of FNN for joint 1 after the 20th learning trial.

i	WA _i	WC _{i0}	WC _{i1}	WC _{i2}
1	1.193	-0.525	0.244	0.315
2	1.017	-0.156	0.050	0.091
3	1.839	-1.406	0.654	0.063
4	1.152	-0.553	0.161	0.014
5	1.097	-0.361	0.167	-0.230
6	1.007	-0.105	0.030	-0.043
7	1.118	-0.247	0.455	0.124
8	1.062	-0.292	0.149	0.176
9	1.650	0.157	1.331	0.110

표 5B. 20회 학습후 조인트 2에 대한 FNN의 가중치

Table 5B. Identified weight of FNN for joint 2 after the 20th learning trial.

i	WA _i	WC _{i0}	WC _{i1}	WC _{i2}
1	1.203	-0.134	0.679	0.126
2	1.080	0.052	0.432	0.093
3	1.008	-0.042	0.128	-0.041
4	1.101	0.363	0.187	-0.237
5	1.001	0.039	0.012	-0.020
6	1.780	1.322	0.674	0.015
7	1.056	0.335	0.062	0.025
8	1.228	0.542	0.289	0.389
9	1.003	0.064	0.016	0.035

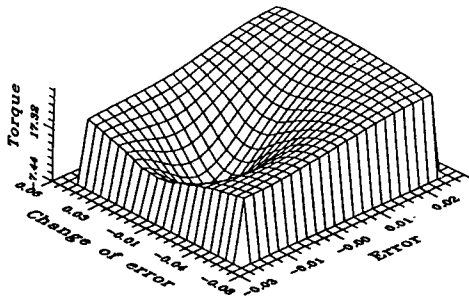


그림 12. 20번째 학습시 조인트 1의 제어평면
 Fig. 12. Control surface of joint 1 during the 20th learning trial.

인트에 가해지는 토크의 변화를 나타낸다. 학습을 거듭함에 따라 P-제어기의 토크는 영(zero)으로 수렴하며 FNN의 토크만이 로봇에 입력으로 주어짐을 볼 수 있다. 표 5는 20회 학습후 FNN의 가중치이다. 제어목적에 맞는 적절한 규칙이 생성되었음을 볼 수 있다. 그림 12와 그림 13은 20번째 학습시 조인트 1과 조인트 2의 제어평면을 나타낸다. 이와같은 결과로부터 제안한 알고리즘이 로봇제어에 효과적임을 알 수 있다.

2. 모의실험(2)

본 모의실험에서는 로봇이 20회의 학습후에 식 (19)로 주어지는 학습하지 않는 다른 궤적을 추종하도록 하였다.

$$X = 0.5 + 0.17(1 - \cos[\theta(t)])\cos[\theta(t)] \quad (19a)$$

$$Y = 0.3 + 0.17(1 - \cos[\theta(t)])\sin[\theta(t)] \quad (19b)$$

그림 14의 결과로부터 로봇은 더이상의 학습 없이도 새로 주어진 궤적을 훌륭하게 추종함을 알 수 있다.

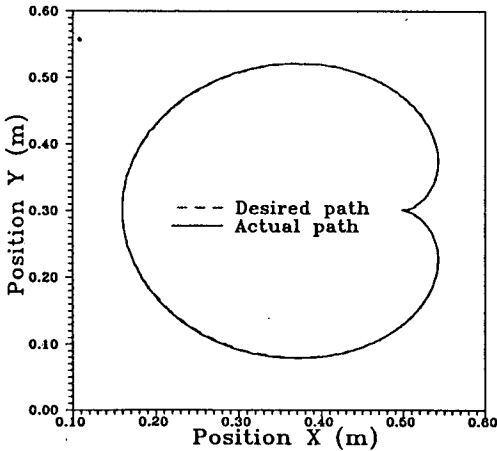


그림 14. 학습하지 않은 궤적에 대한 응답
Fig. 14. Response to unlearned trajectory.

3. 모의실험 (3)

본 모의실험에서는 20회의 학습수에 로봇의 각 링크의 무게를 30% 증가 시켰다. 그림 15는 링크의 무게를 증가시킨 후 첫번째 학습시의 응답을 나타낸 것이고 그림 16은 10회 학습시의 응답이다. 이 결과는 FNN이 로봇 링크 무게의 변화에도 훌륭한 적응성을 보임을 보여준다.

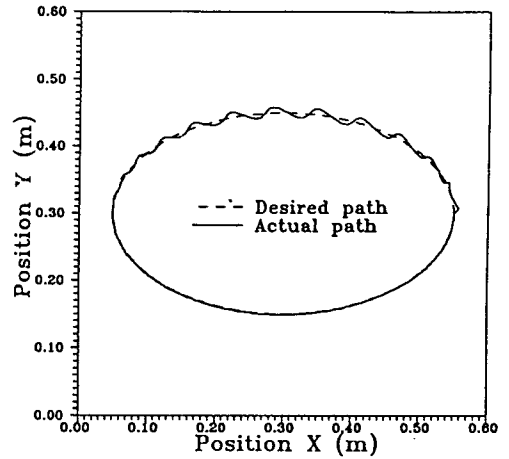


그림 15. 첫번째 학습시 로봇의 경로 (링크의 무게를 30% 증가시킨 후)
Fig. 15. Desired and actual path of a robot at the 1st learning trial(after increase the weight of link by 30%).

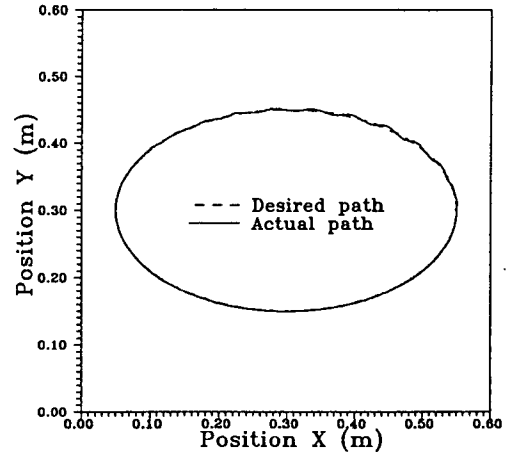


그림 16. 10번째 학습시 로봇의 경로 (링크의 무게를 30% 증가시킨 후)
Fig. 16. Desired and actual path of a robot at the 10st learning trial(after increase the weight of link by 30%).

V. 결 론

본 논문에서는 기존의 퍼지 논리 제어기에서 문제점으로 지적되었던 규칙베이스의 자동생성과 제어환경 변화에 대한 적응성의 문제를 해결하기 위하여 퍼지논리와 신경회로망의 장점들을 결합한 퍼지-신경망(FNN)을 제안

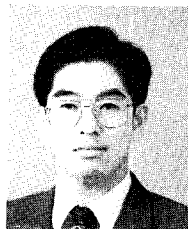
하였다. 제안된 FNN은 퍼지 제어기의 규칙베이스와 추론과정을 신경망으로 구현한 것이며 역전파 학습 알고리즘을 이용 매개변수를 학습하도록 되어 있다.

그리고, 상기 FNN과 일반적인 선형제어기인 P-제어기를 병렬로 연결하여 퍼지 제어 규칙을 오차역전파 학습 알고리즘에 의하여 자동생성하는 새로운 형태의 퍼지-뉴럴 하이브리드 제어기를 설계하였다. 또한, 2자유도 로봇트 매니퓰레이터를 이용한 컴퓨터 모의 실험을 통하여 제안된 방식이 비선형의 동특성을 갖는 시스템의 제어에 유효하며 다음과 같은 장점을 가지고 있음을 보였다. 첫째, 제안된 방식은 학습을 통하여 제어목적에 맞는 규칙을 자동생성한다. 둘째, 본 방식은 제어환경의 변화에도 유연하게 대응한다.

參 考 文 獻

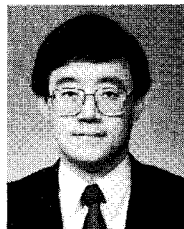
- [1] M. Sugeno and M. Nishida, "Fuzzy control of model car," *Fuzzy Sets and Syst.*, vol. 16. pp. 103-113, 1985.
- [2] Y. F. Li and C. C. Lau, "Development of fuzzy algorithms for servo systems," *IEEE Control Syst. Mag.*, pp. 65-72, Apr. 1989.
- [3] J. A. Bernard, "Use of a rule-based system for process control," *IEEE Control Syst. Mag.*, pp. 3-13, Oct. 1988.
- [4] M. Kawato, Y. Uno, M. Isobe, and R. Suzuki, "Hierarchical neural network model for voluntary movement with application to robotics," *IEEE Control Syst. Mag.*, pp. 8-16 Apr. 1988.
- [5] S. Nagata, M. Sekiguchi, and K. Asakawa, "Mobile robot control by a structured hierarchical neural network," *IEEE Control Syst. Mag.*, pp. 69-76 Apr. 1990.
- [6] C. W. Anderson, "Learning to control an inverted pendulum using neural networks," *IEEE control Syst. Mag.*, pp. 31-37 Apr. 1989.
- [7] T. Tagaki and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-15, pp. 116-132, Jan. /Feb. 1985.
- [8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *Parallel Distributed Processing*, The MIT Press, pp. 318-362, 1986.
- [9] C. C. Lee, "Fuzzy logic in control systems : Fuzzy logic controller-part I, II," *IEEE Trans. Syst. Man Cybern.*, vol. 20, no. 2, pp. 404-435, Mar. / Apr. 1990.
- [10] C. C. Lee, "Intelligent Controller based on Fuzzy Logic and NeuralNet theory," *Proc. of Int. Conf. on Fuzzy logic and NN*, vol. 2, pp. 759-764, 1990.
- [11] S. Horikawa, T. Furuhasi, S. Okuma, Y. Uchikawa, "Composition Methods of Fuzzy Neural Networks," *Proc. of IECON'90*, vol. 2, pp. 1253-1258, 1990.
- [12] H. Takagi, "Fusion technology of Fuzzy theory and Neural networks-Survey and future directions," *Proc. of Int. Conf. on Fuzzy Logic and NN*. vol. 1, pp. 13-26, 1990.
- [13] 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.
- [14] 임광우, 김성현, 심귀보, 전홍태, 퍼지-뉴럴 융합을 이용한 로봇트 Gripper의 힘 제어기, '91 한국자동제어학술회의 논문집, vol. 1, pp. 861-865, 1991.
- [15] 임광우, 조현찬, 강훈, 전홍태, 퍼지-뉴럴 합성을 이용한 제어기의 설계, '92 한국자동제어학술회의 논문집, vol. 1, pp. 117-122, 1992.
- [16] 전홍태, 조현찬, 퍼지 뉴럴의 융합 기법, 전자공학회지, vol. 18, no. 10, pp. 64-71, 1991.

著 者 紹 介

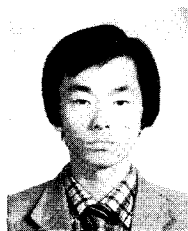


林 光 宇 (正會員)
 1967年 10月 31日生. 1990年 중앙대학교 전자공학과(공학사). 1990年 ~ 현재 중앙대학교 대학원 전자공학과 석사과정 재학중. 주관심분야는 Robotics, Neural network, Fuzzy system, Fuzzy-neural fusion

등임.



姜 勳 (正會員)
 1959年 11月 19日生. 1982年 서울대학교 전자공학과(공학사). 1984年 서울대학교 대학원 전자공학과(공학석사). 1989年 Georgia Institute of Technology(공학박사). 1990年 Georgia Institute of Technology Post Doctor 연구원. 1991年 Georgia Institute of Technology 연구교수. 1992年 ~ 현재 중앙대학교 제어계측공학과 조교수. 주관심분야는 지능제어, 퍼지시스템및 신경망, Robotics, 적응제어 등임.



金 容 虎 (正會員)
 1962年 3月 4日生. 1986年 광운대학교 전자공학과(공학사). 1988年 중앙대학교 대학원 전자공학과(공학석사). 1992年 ~ 현재 중앙대학교 대학원 전자공학과 박사과정 재학중. 주관심분야는 Robotics, Neural network, Fuzzy system, Genetic Algorithms 등임.

全 洪 兌 (正會員)
 1955年 11月 27日生. 1976年 서울대학교 전자공학과(공학사). 1983年 뉴욕 주립대 전기및 전자공학과(공학석사). 1986年 뉴욕 주립대 전기 및 전자공학과(공학박사). 1986年 ~ 현재 중앙대학교 전자공학과 전자공학과 부교수. 주관심분야는 Robotics, Neural network, Fuzzy logic control, Fuzzy-neural fusion 등임.