

성능평가 계층을 가지는 신경망제어기 설계

* 이현철, 조원철, 전기준

경북대학교 공과대학 전자공학과, 제어계측신기술연구센터(서울대)

Neural Network Controller Design with a Performance Evaluation Level

Hyun Cheol Lee, Won Chul Cho, Gi Joon Jeon

Department of Electronics, Kyungpook National University, Taegu, Korea and
Engr. Research Center for Adv. Control and Instr. of SNU by KOSEF

ABSTRACT

We propose a new control architecture which consists of a PI controller and a neural network(NN) controller connected together in parallel. This architecture is well adapted to a wide range of uncertainties and variations of systems. The NN controller is learned through weights of the emulator which identify the dynamic characteristics of the systems. A performance evaluation level of two NN's decides automatically which controller of the two controllers will be used mainly. The PI controller operates mainly during learning phase of the NN controller whereas a good performance is obtained from the NN controller only, when the NN controller is learned sufficiently.

1. 서론

고전적인 제어기는 제어하고자 하는 시스템의 특성을 수학적으로 모델링하고 그 모델에 대한 수치적 계산을 통해 제어를 하기 때문에 모델링의 불확실성과 모델링후의 시스템 동특성의 변화는 제어기의 성능에 큰 영향을 미치게 된다. 기존의 수학적방법으로 모델링된 시스템의 불확실성을 극복하기 위한 방법으로 시스템의 상황을 적절하게 표현하여 이에 적합한 제어법칙을 구하는 방법이 모색되고 있다. 근본적으로 선형시스템에 근거한 제어방법으로는 시스템의 파라미터를 추정하여 제어법칙을 구하는 적응제어와 시스템 파라미터의 변화범위를 고려하여 제어기를 구성하는 강건제어 방법을 들수있다. 그러나, 시스템이 복잡해지고 시스템의 비선형성과 불확실성이 커지면 기존의 수학적 방법으로는 시스템을 정확히 모델링 할수 없게 되고 또한 시스템 성능의 다양한 요구로 인하여 기존의 제어 이론으로는 그 목적을 달성하기가 어렵거나 불가능하게 된다.

이와같이 동특성이 복잡하면서 불확실성과 비선형성이 강한 시스템을 모델링하고 제어하기 위하여 신경회로망을 이용한 지능 제어 방법이 등장하게 되었다[1].

신경회로망을 이용한 제어는 시스템의 입출력 정보를 이용하여 비선형성을 갖는 시스템의 모델링 및 제어를 하는 동시에 시스템의 동특성 변화에 대해 지속적으로 학습할수 있는 능력을 가진다. 이는 실제 시스템이 대부분 비선형 특성을 가지고 지속적인 변이가 존재하는 동시에 복잡성을 가지므로 기존의 제어 방법의 한계성에 따른 문제점의 해결 방안을 제시하고 있다. 신경회로망만을 제어기로 사용한 구조로서 Psaltis[2]는 플랜트의 입력을 광범위하게 바꿔가면서 플랜트의 역동특성을 학습후 제어기 구성에 사용하였고, Nguyen[3]은 시스템의 특성을 모사하는 emulator를 신경망으로 구성한후 이를 이용하여 제어 신경망을 학습하여 제어법칙을 구하는 방법을 제시하였으나 이들은 특정 입력력 쌍으로 플랜트 특성을 완전히 학습하는데 대한 문제점과 플랜트의 동특성 변이에 대처할수 없다는 문제점을 갖는다. 이에 반하여 Narendra등[4]은 시스템의 특성을 모사하는 식별신경망과 제어신경망을 동시에 on-line 학습 시킴으로서 플랜트의 동특성 변이에 잘 대처할수 있는 구조를 제안하였다.

기존의 제어방식과 병렬로 신경망제어기를 결합하여 사용하는 구조로서 Iiguni[5]는 기존의 최적제어기법의 한계성을 신경망으로 보상하여 제어성능을 개선시키는 구조를 제안하였고 Kawato등[6,7]은 PD 제어기와 그들이 제안한 신경망제어기로 구성된 로봇 제어기를 제안하였는데 이 제어기에서 신경회로망의 학습을 위한 오차신호는 PD 제어기의 출력으로 구성된다. 여기서 신경망제어기가 하는 역할은 플랜트에 대한 역함수 기능을 하는 것인데 이 구조의 취약점은 플랜트의 역함수를 제어기로 사용하였을때 문제가 발생하는 경우가 있는데 이런 경우는 제외되어야 한다는 것이다. 또한, PD 제어기의 출력이 신

경망제어기의 학습을 위한 오차신호로 사용되므로 PD 제어가 시스템의 특성을 얼마나 잘 파악하고 있느냐 하는 것이 제어기 학습에 큰 영향을 주게 된다.

본 논문에서는 기존의 고전적 제어방법을 최대한으로 이용하고자 기존의 고전적인 제어기와 신경망제어기를 병렬로 결합하여 보다 넓은 범위의 모델링의 불확실성과 모델링후의 변이에 잘 적응하는 제어기를 구성하고자 하였다. 기존의 고전적 제어기로는 PI 제어를 이용하고 플랜트를 모사하는 식별신경망의 정보를 이용하여 신경망제어기를 학습시킨다. 또한, 식별신경망과 제어신경망의 성능을 평가하는 계층을 두고 PI 제어기와 신경망제어기중 어느 제어기를 더 주된 제어기로 사용할 것인가를 자율적으로 결정할 수 있는 능력을 부가하여, 학습 초기에는 PI 제어가 주된 작용을 하고 신경망이 완전히 학습된 후에는 신경망제어기만으로 원하는 제어성능을 얻을 수 있도록 하였다. 이러한 제어방법은 현재까지의 축적된 지식으로 시스템을 수학적으로 근사화 할 수 있고 실제 플랜트가 주어진 모델과 오차가 있거나 변이가 있는 경우 잘 이용될 수 있다.

II. 신경망의 학습이론

신경 회로망은 인간의 신경구조를 모방하여 병렬분산처리를 통해 주어진 입력력 정보만으로 복잡한 계산을 반복적인 간단한 계산으로 대체할 수 있는 특성을 갖는다. 본 논문에서 사용하는 다층 신경회로망 모델은 입력층과 중간층, 그리고 출력층으로 구성되어 있고, 각 층에는 뉴런들이 병렬로 연결되어 있어 병렬분산처리를 가능하게 한다. 여기서, 서로 연결된 뉴런간의 가중치는 학습에 의해 조절되며 널리 쓰이는 학습법칙으로는 수학적으로 최대경사법의 일종인 오차역전파 방법을 들 수 있다. 오차역전파 방법을 이용한 신경회로망의 학습과정은 다음과 같다. 먼저, 각 층의 연결강도를 임의의 작은 값으로 초기화한다. 다음으로 입력패턴을 신경회로망에 인가하여 실제 출력값을 계산하게 된다. 이때, 각 뉴런의 출력력 특성은 출력단의 k번째 뉴런을 예로들면 다음의 수학적 모델을 갖는다.

$$f(\text{net}_k) = \frac{1 - \exp(-\text{net}_k)}{1 + \exp(-\text{net}_k)} \quad (2-1a)$$

$$\text{Out}_k = f(\text{net}_k) \quad (2-1b)$$

$$\text{net}_k = \sum_j (W_{kj} \cdot \text{Out}_j) \quad (2-1c)$$

여기서 Out_j 는 중간층에서 뉴런 j의 출력이며 W_{kj} 는 출력층의 뉴런 k와 중간층의 뉴런 j사이의 가중치를 나타내고, $f(\cdot)$ 은 미분가능한 단조증가 함수로서 신경회로망의 비선형 특성에 관

여한다. 일단 출력값이 결정되면 신경회로망의 출력층에서 원하는 출력값(D)과 실제 출력값(Out)사이의 오차함수를 다음과 같이 정의하고, 아래의 오차함수를 최소화하도록 가중치를 변 화시킨다.

$$E = \frac{1}{2} \sum_k (D_k - \text{Out}_k)^2 \quad (2-2)$$

식(2-2)의 오차E를 감소시키기 위한 출력단의 가중치들의 변화량은 gradient descent방법에 의해 다음과 같이 구할 수 있다.

$$\Delta W_{kj} = \eta \cdot \delta_k \cdot \text{Out}_j \quad (2-3a)$$

$$\delta_k = (D_k - \text{Out}_k) \cdot f'(\text{net}_k) \quad (2-3b)$$

$$f'(\text{net}_k) = \frac{\partial f(\text{net}_k)}{\partial \text{net}_k} \quad (2-3c)$$

$$= (1/2) \cdot \{1 - (f(\text{net}_k))^2\}$$

여기에서 η 는 학습률 계수로서 $0 < \eta < 1$ 이고, δ_k 는 역방향 으로부터 전달되어 오는 오차이다. 또한, 중간층에서의 가중치 변화량은 다음식에 의해 구할 수 있다.

$$\Delta W_{ji} = \eta \cdot \delta_j \cdot \text{Out}_i \quad (2-4a)$$

$$\delta_j = f'(\text{net}_j) \cdot \sum_k (\delta_k \cdot W_{kj}) \quad (2-4b)$$

식 (2-3)과 (2-4)를 이용하여 각 층에서의 새로운 가중치들은 최종적으로 다음식을 통해 조정된다.

$$W_{kj}(n+1) = W_{kj}(n) + \Delta W_{kj} + \alpha \cdot (W_{kj}(n) - W_{kj}(n-1)) \quad (2-5a)$$

$$W_{ji}(n+1) = W_{ji}(n) + \Delta W_{ji} + \alpha \cdot (W_{ji}(n) - W_{ji}(n-1)) \quad (2-5b)$$

여기서 n은 학습의 반복횟수를 나타내며, α 는 학습속도를 보다 빠르게 하기 위해 첨가된 운동량 항(momentum term)으로서 $0 < \alpha < 1$ 이다.

III. 신경망 제어구조 및 학습

3.1. 신경망 제어구조

일반적으로 시스템은 해석, 설계, 제작의 과정을 통하여 제어기를 구성함으로써 모든 작업이 완성된다. 우리가 설계를 하고 이에따라 정확히 제작을 한다면 우리는 원하는 제어기를 제작할 수 있으나 실제로 이 과정에서 제작한 시스템의 동특성이 설계한 특성과 일반적으로 일치하지 않는다는 문제점이 발생한다. 본 논문에서는 이러한 문제점을 해결하기 위해 기존의 제작된 제어기와 신경망 제어기를 병렬로 결합하여 모델링의 불확실성과 모델링후의 변이에 잘 적응하는 신경망 제어구조

를 제안하고자 한다. 제안하는 제어시스템의 구조는 그림 1과 같다.

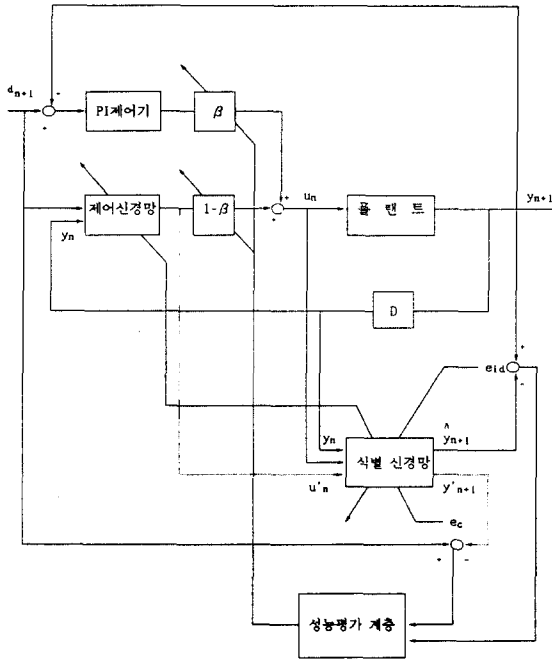


그림 1. 제안하는 신경망 제어구조

Fig.1. Proposed neural network control architecture

그림 1에서 다층신경망으로 구성된 신경망제어기와 기존의 PI 제어기가 병렬로 연결되어있다. 플랜트 제어를 위한 입력은 PI제어기의 출력과 신경망제어기의 출력의 변형된 합으로서 식 (3-1)과 같이 나타낸다.

$$U(n) = \beta \cdot U_{PI}(n) + (1 - \beta) \cdot U_{NN}(n) \quad (3-1)$$

여기서 $U_{NN}(n)$ 은 n 시간에서의 신경망제어기의 출력을 나타내고 $U_{PI}(n)$ 은 이산시간 형태의 PI제어기의 출력으로 식(3-2)와 같다.

$$\begin{aligned} U_{PI}(n) &= P(n) + I(n) \\ P(n) &= K_p \cdot (d_n - y_n) \end{aligned} \quad (3-2)$$

$$I(n+1) = I(n) + K_i \cdot T \cdot (d_n - y_n)$$

여기서 $P(n)$ 은 PI 제어기의 비례요소, $I(n)$ 은 적분요소를 나타내며, T 는 샘플링 시간을 나타낸다. 또한, β 는 식별신경망과 제어신경망의 오차의 함수로 정의되며 제어결과의 성능평가로부터 얻은 정보를 이용하여 0과 1 사이의 값으로 자동조정된다. 신경회로망의 학습초기에는 β 는 1로서 식 (3-1)에서 나타나는것 처럼 플랜트의 제어를 위한 입력으로 PI 제어기의 출력만이 사용되고 점점 신경회로망의 학습이 이루어질때 0으로

가까와져서 신경회로망의 오차가 완전히 없어졌을때 β 는 0이 되어 신경망제어기의 출력만이 플랜트 제어를 위한 입력으로 사용된다.

3.2. 식별 신경망의 학습

신경회로망을 이용하여 간접제어 방식의 제어기를 구성할때에는 학습된 식별신경망의 정보를 이용하여 제어기를 학습하므로 식별신경망이 플랜트의 특성을 정확히 모사하도록 학습이 되어야 한다. 여기서 사용되는 다층신경망 모델은 하나의 중간계층을 가지고 두개의 입력단자와 하나의 출력단자를 가지는 구조를 하고 있다. 두개의 단자중 하나는 플랜트에 들어가는 입력값 U_n 이고 다른 하나는 플랜트의 이전 출력값 y_n 이다. 식별신경망의 연결강도는 다음의 식들에 의해 보정된다. 연결강도 조절을 위한 성능 지수는 플랜트의 출력과 신경망의 출력과의 오차들의 합으로서 다음과 같이 정의한다.

$$e_{id} = \frac{1}{2} \sum_k (y_k - \hat{y}_k)^2 \quad (3-3)$$

출력계층에서의 연결강도를 보정하기 위한 k 번째 출력 단자에서의 오차는 다음과 같이 구해질수 있다.

$$\delta_{ek} = (y_k - \hat{y}_k) \cdot f'_k(\text{net}_k) \quad (3-4)$$

위에서 구한 식을 사용하여 식별신경망의 출력계층의 k 번째 단자와 중간계층의 j 번째 단자 사이의 연결강도를 보정하기 위한 식은 다음과 같이 구해진다.

$$\Delta W_{kj} = -\eta \cdot \frac{\partial e_{id}}{\partial W_{kj}} = \eta \cdot \delta_{ek} \cdot O_{oj} \quad (3-5)$$

여기서 W 는 식별신경망의 연결강도의 벡터이고 O_j 는 중간계층의 j 번째 단자에서의 출력값을 나타낸다. 위의 식들에 붙은 첨자 e 는 식별신경망(emulator)을 의미한다. 또한, 중간계층에서의 연결강도 조절을 위한 오차는 다음과 같이 구해질수 있다.

$$\delta_{oj} = \sum_k \delta_{ek} \cdot W_{kj} \cdot f'_j(\text{net}_j) \quad (3-6)$$

이를 이용하여 중간계층의 j 번째 단자와 입력계층의 i 번째 단자사이의 연결강도를 보정하기 위한 식은 다음과 같이 구해진다.

$$\Delta W_{ji} = -\eta \cdot \frac{\partial e_{id}}{\partial W_{ji}} = \eta \cdot \delta_{oj} \cdot Z_{oi} \quad (3-7)$$

여기서 Z_{oi} 는 식별신경망의 입력계층의 i 번째 단자에 들어오는 입력값을 나타낸다. 위에서 구한 식 (3-5)과 (3-7)의 연결강도의 변화분을 식 (2-5a)와 (2-5b)에 대입함으로서 오차역전달

학습 알고리즘에 의한 학습이 이루어 진다.

3.3. 제어신경망의 학습

식별신경망이 플랜트의 특성을 어느정도 학습한뒤 부터는 식별신경망의 학습과 동시에 제어신경망의 학습이 행해진다. 제어신경망의 입력은 현재의 출력값과 목표값이 되고 출력은 플랜트 제어를 위한 입력값이 된다. 제어신경망의 학습은 학습된 식별신경망의 정보를 이용하여 수행된다. 즉, 그림1에서 보여지는것처럼 플랜트의 특성을 잘 모사하는 식별신경망의 출력과 목표값 사이의 오차가 식별신경망의 연결강도를 통하여 제어기의 출력층에 역전달됨으로써 제어기의 연결강도의 보정이 이루어진다. 제어신경망의 연결강도를 조정하기 위한 성능지수는 다음과 같다.

$$e_c = \frac{1}{2} \sum_k (d_k - y_k')^2 \quad (3-8)$$

위의 식을 사용하여 구한 식별신경망의 출력 계층과 중간계층의 각 단자들의 오차는 다음과 같이 구해질수 있다.

$$\delta_{ek} = (d_k - y_k') \cdot f_k'(\text{net}_k) \quad (3-9)$$

$$\delta_{ej} = \sum_k \delta_{ek} \cdot w_{kj} \cdot f_j'(\text{net}_j) \quad (3-10)$$

식 (3-9)과 (3-10)에서 구한 식별신경망 각 단자의 오차들은 식별신경망의 연결강도를 보정하기 위한것이 아니라 식별신경망의 출력과 목표값 사이의 오차를 제어기에 역전달하기 위하여 구해지는 것이다. 그러므로, 식별신경망의 연결강도는 변화되지 않는다. 제어신경망의 학습을 위한 오차는 위에서 구한 식별신경망의 중간계층 단자들의 오차를 제어기의 출력단자에 전달하여 다음과 같이 구할수 있다.

$$\delta_{ck} = \sum_j \delta_{ej} \cdot w_{ji} \cdot f_k'(\text{net}_k) \quad (3-11)$$

위에서 아래첨자 c는 제어기를 의미한다. 이를 이용하여 제어기의 출력계층과 중간계층 사이의 연결강도를 보정하기 위한 식은 다음과 같이 구해진다.

$$\Delta V_{kj} = -\eta \cdot \frac{\partial e_c}{\partial V_{kj}} = \eta \cdot \delta_{ck} \cdot o_{cj} \quad (3-12)$$

여기서 W는 식별신경망의 연결강도의 벡터이고 V는 제어신경망의 연결강도의 벡터이다. 또한, 중간계층에서의 오차는 다음과 같다.

$$\delta_{cj} = \sum_k \delta_{ck} \cdot v_{kj} \cdot f_j'(\text{net}_j) \quad (3-13)$$

식 (3-13)를 이용하여 제어기의 중간계층과 입력계층 사이의

연결강도는 다음과 같이 보정된다.

$$\Delta V_{ji} = -\eta \cdot \frac{\partial e_c}{\partial V_{ji}} = \eta \cdot \delta_{cj} \cdot z_{ci} \quad (3-14)$$

3.4. 성능평가 계층

그림 1에서 성능평가 계층은 두 신경망의 학습 정도를 파악하여 계수 β 를 자동적으로 조정하는 기능을 수행한다. 이때, 계수 β 는 두 신경망의 학습오차의 함수로 정의되어 적절히 조정된다. 먼저 M시간 동안의 식별신경망과 제어신경망의 학습오차를 각각 다음과 같이 정의한다.

$$E_{id} = \sum_{n=mM+1}^{(m+1)M} \frac{1}{|d.c.}| |e_{id}(n)|, \quad m = 0, 1, 2, 3, \dots \quad (3-15)$$

$$E_c = \sum_{n=mM+1}^{(m+1)M} \frac{1}{|d.c.}| |e_c(n)|, \quad m = 0, 1, 2, 3, \dots \quad (3-16)$$

여기서 d.c.는 목표값(set-point)의 변화량으로서 목표값에 따른 신경망의 오차값을 표준화하기 위해 사용되며 목표값의 변화가 없는 경우에는 이전상태의 값을 계속 유지한다. 또한, M 시간 동안의 신경망의 전체오차를 식 (3-17)로 정의한다.

$$E_{total} = p \cdot E_{id} + q \cdot E_c \quad (3-17)$$

여기서 p 와 q 는 어느 신경망의 오차에 더 큰 비중을 둘것인가를 결정하는 인자로서 동일하게 0.5로 두었다. 만약 p를 보다 크게하면 E_{total} 값이 플랜트의 변화에 더 민감하게 되고, q를 크게하면 신경망제어기의 성능이 보다 좋을때 두 제어기의 전환이 이루어지게 된다. 계수 β 의 조정은 매 M 시간마다 일어나며 β 의 값은 E_{total} 의 값에 따라 다음과 같이 결정된다.

$$\beta = \frac{1}{1 + \exp[-h \cdot (E_{total} - 3)]} \quad (3-18)$$

여기서 h 는 이 함수의 기울기로서 신경망의 오차가 어떤값 이하로 떨어질때 두 제어기의 전환을 시작할것 인가를 결정한다. 기울기 h의 변화에 따른 이 함수의 변화는 그림 2 와 같다.

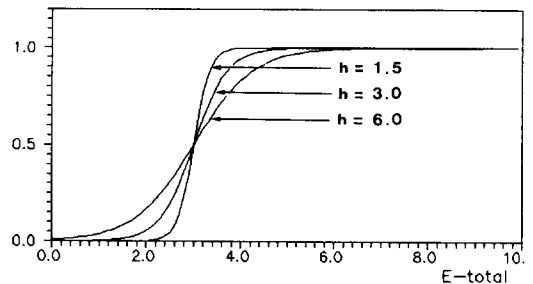


그림 2. 기울기에 따른 계수 β 의 변화

Fig. 2. Variation of coefficient β with slope

그림 2에서 계수 β 의 값은 E_{total} 값이 클때 1이 되고 E_{total} 값이 점점 감소함에 따라 같이 감소하여 E_{total} 값이 거의 0이 되었을때 0의 값을 가지게 된다. 플랜트 제어를 위한 전체 입력 U 는 식 (3-1)에 나타나는 것처럼 다음과 같다.

$$U = \beta \cdot U_{PI} + (1-\beta) \cdot U_{NN}$$

여기서 β 의 값이 1일때는 $U = U_{PI}$ 가 되고 β 가 0으로 가까와 질수록 U 에 U_{NN} 의 영향이 커지게 되어 β 가 0이 되면 $U = U_{NN}$ 이 된다. 또한, 신경망이 어느정도 학습된 다음에 플랜트의 갑작스런 변화에 의해 축적되는 E_{total} 의 값이 한계치 이상을 넘어간다면 그 순간에 β 의 값을 조정한다. 이와같이 계수 β 를 조정함으로써 신경망이 학습이 얹된 초기에는 PI 제어기가 주로 동작하고 신경망이 점점 학습이 되어갈수록 신경망 제어가 플랜트 제어를 담당하게 된다. 그리고, 플랜트의 작은 변화와 순간적인 작은 외란에 대해서는 신경망 제어가 좋은 견실성을 가지며 플랜트의 갑작스런 큰 변화나 지속적인 큰 외란에 대해서는 신경망이 다시 학습될때까지 PI제어기가 플랜트 제어를 담당하게 된다.

IV. 모의 실험

모의실험에서는 입력층, 중간층, 그리고 출력층으로 구성되는 2층 신경회로망 모델을 사용하였다. 두 신경망은 입력층에 2개의 뉴런, 중간층에 30개의 뉴런, 그리고 출력층에 하나의 뉴런을 가지는 구조를 하고 있다. 각 신경망의 가중치는 -0.5에서 +0.5사이의 임의의 값으로 초기화하였으며 학습률과 모멘텀은 각각 두신경망에 0.5, 0.3으로 동일하게 두었다. 모의 실험에 사용된 플랜트는 다음과 같다.

$$y[n+1] = 0.7 \cdot \sin(2 \cdot u[n]) + y[n]$$

여기서 플랜트의 초기치($y[0]$)와 입력의 초기치($u[-1]$)는 모두 0으로 두고 샘플링 시간 T 는 0.1로 두었다. 플랜트 출력의 목표값은 0.5와 0으로 두고 PI 제어기의 계수 K_p 와 K_i 는 각각 0.45, 0.5로 두었다.

그림 3은 신경망 제어가 15회 학습하고난 후의 플랜트 출력의 추종결과로서 점선은 PI제어기만으로 제어했을때의 결과를 나타내고 실선은 병렬결합된 제어구조로서 제어한 결과를 나타낸다. 여기서 1회는 0.5와 0의 목표값이 한번 나타나는 시간으로 10초 즉, 100 step이다. 그리고, 계속 학습되어 30회 학습하고난 후의 결과가 그림 4에 나타난다. 그림 5.6은 신경망이 거의 완전히 학습이된 후에 플랜트의 변이가 있었을

때의 추종결과를 나타낸다. 그림 5는 플랜트의 파라미터가 갑자기 10% 정도 변했을때의 추종결과로서 이 정도의 작은 변화에는 신경망이 그 변화에 쉽게 적응함을 보여준다. 그림 6은 플랜트의 파라미터가 갑자기 40% 정도 변했을때 병렬결합된 제어구조로서 제어한 결과와 신경망제어기만으로 제어했을때의 결과를 나타낸다. 여기서, 점선은 신경망 제어기만으로 제어했을 때의 결과를 나타내고 실선은 PI 제어기와 신경망제어기를 병렬결합하여 제어한 결과이다. 플랜트의 큰 변화로 신경망이 다시 학습을 해야 된다면 학습될때까지 PI 제어기가 다시 동작하게 된다. 또한, 플랜트 변화 후 다시 여러번 학습후에 그림 3을 거쳐 그림 4와 같은 결과를 얻을 수 있다. 즉, 병렬결합된 제어구조는 플랜트의 변화에도 계속 적용할수 있는 능력을 가진다.

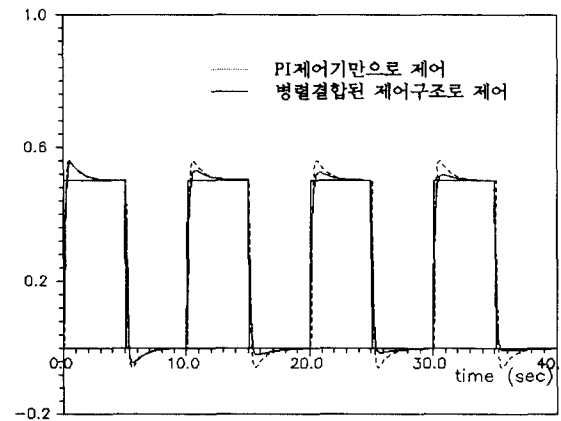


그림 3. 15회 학습후의 출력응답

Fig. 3. Output response after 15 iterations

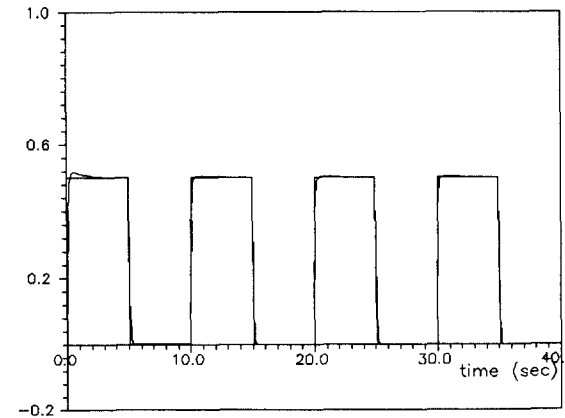


그림 4. 30회 학습후의 출력응답

Fig. 4. Output response after 30 iterations

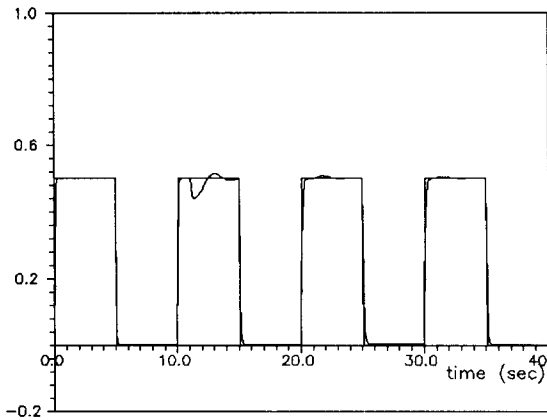


그림 5. 학습후 파라미터가 10% 변했을 경우의 출력응답

Fig.5. Output response in the case of parameter 10 % change after learning

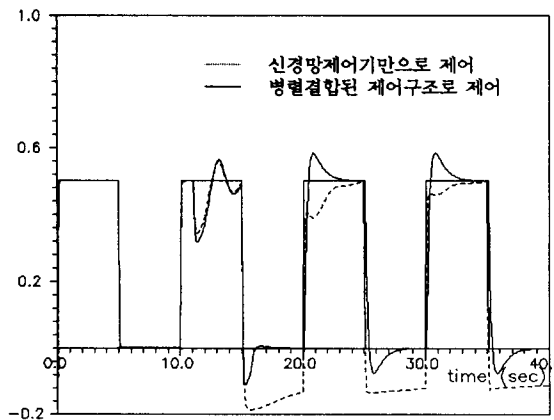


그림 6. 학습후 파라미터가 40% 변했을 경우의 출력응답

Fig.6. Output response in the case of parameter 40 % change after learning

V. 결론

본 연구에서는 기존의 PI 제어기와 신경망제어기를 병렬로 결합하여 신경망의 학습정도에 따라 두 제어기중 어느 제어기를 더 주된 제어기로 사용할 것인가를 자동적으로 결정할수 있는 제어구조를 제안하였다. 여기서 신경망 제어기는 플랜트의

특성을 입출력 관계로 표시한 식별신경망의 정보를 이용하여 학습하므로 플랜트의 정확한 모델없이도 제어가 가능하게 된다. 또한, 학습후 플랜트의 변화가 있을 경우에 그 변화에 학습되지 않은 초기에는 PI 제어기가 다시 주 제어기로 동작하고 신경망이 학습이 되어감에 따라 신경망 제어기가 주 제어기로 동작하게 되어 지속적으로 성능을 향상시킬수 있는 기능을 가진다.

참고 문헌

- [1] B. Bavarian, " Introduction to Neural Networks for Intelligent Control," *IEEE Contr. Syst. Mag.*, 8, 3-7, Apr. 1988.
- [2] D. Psaltis, A. Sideris, and A.A. Yamamura, " A Multilayered Neural Network Controller," *IEEE Contr. Syst. Mag.*, 8, 17-21, Apr. 1988.
- [3] D.H. Nguyen and B. Widrow, " Neural Network for Self-Learning Control System," *IEEE Contr. Syst. Mag.*, 10, 18-23, Apr. 1990.
- [4] K.S. Narendra and K. Parthasarathy, " Identification and Control of Dynamical Systems Using Neural Network , " *IEEE Trans. on Neural Networks*, 1, 4-27, 1990.
- [5] Y. Iiguni, H. Sakai and H. Tokumaru, " A Nonlinear Regulator Design in the Presence of System Uncertainties Using Multilayered Neural Networks," *IEEE Trans. on Neural Networks*, 2, 410-417, 1991.
- [6] M. Kawato, Y. Uno, R. Suzuki, " Hierarchical Neural Network Model for Voluntary Movement with Application to Robotics," *IEEE Contr. Syst. Mag.*, 8, 8-16, Apr. 1988.
- [7] 오세영, 류연식, " 다층 신경회로 및 역전달 학습방법에 의한 로봇팔의 다이내믹 제어," *전기학회 논문지*, 39, 1306-1316, 1990.