

다중 신경회로망을 이용한 DC Servo Motor 제어방법

<sup>0</sup>김석우, 김준식, 유종선, 이영준

LG 산전 연구소

A Control Method of DC Servo Motor Using a Multi-Layered Neural Network.

<sup>0</sup>S.W. KIM, J.S. KIM, J.S. RYOU, Y.J. LEE

LGIS R&D CENTER

### Abstract

A neural network has very simple construction (input, output and connection weight) and then it can be robusted against some disturbance.

In this paper, we proposed a neuro-controller using a Multi-Layered neural network which is combined with PD controller.

The proposed neuro-controller is learned by backpropagation learning rule with momentum and neuro-controller adjusts connection weight in neural network to make approximate dynamic model of DC Servo motor.

Computer Simulation results show that the proposed neuro-controller's performance is better than that of original PD controller.

### 1. 서 론

현대 제어이론의 발달에도 불구하고 PID 제어이론은 실재적으로 산업현장에서 가장 보편적으로 적용되는 이론이다. 이는 제어이론 자체의 간편성과 강인성에 기인한다고 볼 수 있다. 그러나, 제어대상의 정확한 모델링이 어렵고, 모타 제어와 같은 고속의 응답성을 요구하는 경우 적절한 P,I,D 파라메터를 조정하기 어렵다. 이러한 문제점은 최근 널리 이용되는 신경회로망을 도입하여 기존의 PID 제어기와 신경회로망을 병렬로 구성함으로써 해결할 수 있다.

신경회로망은 단순한 입, 출력과 중간의 연결강도를 학습시키는 구조를 가지므로 시스템 및 외부환경 변화에 유연하게 대처할 수 있다. 본 논문에서 제안한 신경회로제어기에서 신경회로망은 역전파 학습 규칙에 의해 학습되며, 제안된 학습방법에 의해 시스템 동력학에 빠르게 적용하게 된다.

본 논문에서는 PD 제어기와 신경회로 제어기를 병렬로 접속하여 DC SERVO MOTOR 를 제어 하고자 하며, 이를 COMPUTER SIMULATION 을 통해 일정한 gain 값을 갖는 PD 제어기의 경우와 신경회로 제어기를 부가시킨 경우의 제어 성능을 비교하여 제안된 제어기의 성능을 검증하고자 한다.

### 2. 제어대상 모델링

그림 1. 에 도시된 DC SERVO MOTOR 의 BLOCK DIAGRAM에서 입력과 출력사이의 관계식을 LAPLACE OPERATOR 를 사용하여 식 (1)과 같이 표현한다.

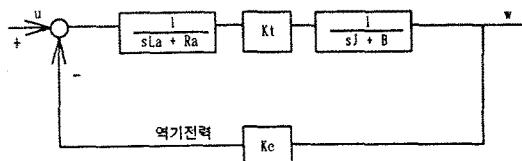


그림 1. DC SERVO MOTOR BLOCK DIAGRAM

단,  $R_a$  : 전기자 저항,  $L_a$  : 전기자 인덕턴스,  
 $J$  : 모타 관성,  $B$  : 마찰감쇠계수,  
 $K_t$  : 토크 상수,  $K_e$  : 역기전력상수

$$\frac{w(s)}{u(s)} = \frac{K_t}{R_a B (1 + L_a/R_a s) (1 + J/B s) + K_t K_e} \quad (1)$$

식(1) 을 단순화시키고 위치에 대한 식(2) 와 같이 표현한다.

$$\frac{\theta(s)}{u(s)} = \frac{K_e}{s(T_a s + 1)} \quad (2)$$

단,  $K_e = 1/K_t$ ,  $T_a = RJ/K_t^2$

식(2) 을 ZOH 를 부가하여 Discretize 시켜 식(3) 을 얻는다.

$$\frac{\theta(z)}{u(z)} = \frac{K_e (z + b)}{(z - 1)(z - c)} = \frac{K_e (z^{-1} + bz^{-2})}{1 - (1 + c)z^{-1} + cz^{-2}} \quad (3)$$

단,  $a = 1/T_a$ ,  $T = \text{sampling period}$   
 $c = \exp(-aT)$ ,  $b = (1 - c - aTc)/(aT - 1 + c)$   
 $K_e = K_t (aT - 1 + c)/a$

식(3) 으로 부터 각 sampling 구간에 따른 위치에 대한 차분방정식을 식 (4)와 같이 구한다.

$$\theta(t) = (1 + c)\theta(t - 1) - c\theta(t - 2) + K_e u(t - 1) + K_e b u(t - 2) \quad (4)$$

### 3. 신경제어기 구조 및 학습규칙

#### 3-1. 신경회로망 구조

본 논문에서 제안한 신경회로망은 Hidden Layer 를 갖는 Multi Layer 구조로 그림 2. 와 같이 Input Layer 는 4 개의 입력 cell, Hidden Layer 는 3 개의 cell, Output Layer 는 1 개의 cell 로 구성된다. 실제적으로 신경제어기의 Input Layer 에는 각각 Trajectory Planning 에 의한 위치 및 속도의 desired value 와 실제 모터의 위치, 속도가 입력되고 출력으로는 torque 가 출력된다.

#### 3-2. 신경제어기를 포함한 전체 시스템 구조

그림 3. 은 본 논문에서 제안한 전체 시스템 구조를 나타내는데 제안된 시스템의 학습방법은 최초 신경제어기의 Weight 를 조정하기 위한 Torque 값은 계산하기 위해 처음 1 scan 은 PD 제어기로 부터의 출력 torque 만으로 모터를 제어하고, 그 다음 scan 부터는 신경제어기의 출력 torque 값이 가산되어 모터를 제어하게 된다. 지정된 횟수만큼의 학습이 끝나면, 대부분 신경제어기에 의한 제어만을 행하게 된다.

#### 3-3. 신경제어기의 학습규칙

신경회로 제어기의 전체 처리 routine 은 그림 4. 와 같으며, 학습이 완료될 때 까지 반복한다. 제안된 신경제어기에는 모멘트 항을 추가하여, Input Layer 와 Hidden Layer 간의 Weight 조정시, Hidden Layer 와 Output Layer 간의 Weight 조정시 모멘트 항이 고려되어 Weight 가 Update 된다.

처음 신경제어기가 기동하면, Trajectory Planner 로 부터의 위치, 속도 값과 실제 모터로 부터의 위치, 속도값을 입력으로 받아 식(5)에 의해 Hidden Layer 의 출력을 계산한다.

$$O_h = \text{Sigmoid}(\sum W_{hi} I_i) \quad (5)$$

Hidden Layer 의 출력값을 입력으로 Output Layer 의 출력을 식(6)으로 부터 구한다.

$$O_o = \text{Sigmoid}(\sum W_{oh} O_h) \quad (6)$$

Output Layer 의 오차값은 PD 제어기의 출력 torque 값을 Target 으로 하여 식(7)로 부터 구한다.

$$\delta_o = \text{Sigmoid}'((\text{Target}-O_o)(\text{net}_o)) \quad (7)$$

Hidden Layer 의 오차값은 Output Layer 의 오차값에 의해 식(8)로 부터 구한다.

$$\delta_h = \text{Sigmoid}'(\sum \delta_o W_{oh} f'(\text{net}_h)) \quad (8)$$

Hidden Layer 와 Output Layer 간의 Weight 값을 식(9)로 부터 구한다.

$$W_{ij}(n+1) = \text{momentum} * W_{ij}(n) + \alpha \delta_o H \quad (9)$$

단, i : Output number, j : Hidden number

H : Hidden Layer 의 출력값

Input Layer 와 Hidden Layer 간의 Weight 값을 식(10)로 부터 구한다.

$$W_{jk}(n+1) = \text{momentum} * W_{jk}(n) + \alpha \delta_h I \quad (10)$$

단, j : Hidden number, k : Input number

I : Input Layer 의 출력값

식 (9) (10)에서 구해진 Weight 값은 식(5), (6)을 계산할 때 적용된다.

#### 4. Computer Simulation

본 논문에서 제어대상이 되는 DC SERVO MOTOR 의 파라미터는 식(11)과 같고, Trajectory Planner 에 의한 위치 및 속도 값은 식(12)과 같다.

$$\begin{aligned} R_s &= 1.2 \text{ Ohms}, J = 12 \times 10^{-4} \text{ Kg.m}^2 \\ K_t &= K_b = 0.12 \text{ Nm/A} \end{aligned} \quad (11)$$

$$\begin{aligned} \text{위치} &= 0.25\pi \sin(0.5\pi t) + \\ &0.25\pi \sin(0.25\pi t) \end{aligned}$$

$$\begin{aligned} \text{속도} &= 0.25 \times 0.5\pi \times \pi \cos(0.5\pi t) + \\ &0.25 \times 0.25\pi \times \pi \cos(0.25\pi t) \end{aligned} \quad (12)$$

신경회로 제어기에서  $\alpha = 0.045$ , momentum = 0.01 으로 하고, sampling time 은 10 msec 로 하였으며, PD제어기의 gain  $K_p = 3.5$ ,  $K_d = 0.4$  로 simulation 하였고, t = 3000msec 시점에서 impulse noise 를 가하여 제어기의 주증상을 확인하였다.

그림 5.은 PD 제어기만으로 제어시의 위치의 출력 그래프이고, 그림 6.은 신경회로 제어기를 부가한 경우의 위치의 출력 그래프이다.

그림 8.은 impulse noise 부가시 PD 제어기만으로 제어시의 위치의 출력 그래프이고, 그림 9.은 신경회로 제어기를 부가한 경우의 위치의 출력 그래프이다.

그림 7.은 PD 제어기와 신경회로 제어기를 포함한 경우의 잔류오차를 표시하였다.

그림 10.은 impulse noise 부가시 PD 제어기와 신경회로 제어기를 포함한 경우의 잔류오차를 표시하였다. simulation 결과에서 볼 수 있듯이 기존의 PD 제어기에 의한 경우보다 신경회로 제어기를 부가한 경우가 잔류오차가 작아짐으로 성능면에서 좋은 결과를 나타내었다.

#### 5. 결 론

본 논문에서는 기존의 PD 제어기의 제반 문제점을 PD 제어기와 신경회로 제어기를 병렬로 접속함으로써 해결하고자 하였으며, 제안된 신경회로 제어기의 성능을 DC SERVO MOTOR 를 제어하여 평가하고자 하였다.

COMPUTER SIMULATION 결과에서 알 수 있듯이 일정한 gain 값을 갖는 PD 제어기의 경우 보다 신경회로 제어기를 부가 시킨 경우가 제어성능이 우수함을 확인하였다.

앞으로 실제 Motor 제어에 확장, 적용하여 Local Minimum 에 빠지지 않는 조건하에서 신경회로 제어기의 적당한 학습계수를 쉽게 찾을 수 있는 방법에 관한 연구가 요구된다.

#### 참 고 문 헌

- 1) D.E. Rumelhart, J.C. McClelland : "Parallel Distributed Processing", MIT Press, 1988
- 2) D. Psaltis, A. Sideris, and A. Yamamura : "A Multilayered neural network Controller", IEEE Control System Magazine, 8(2), p.17, 1988
- 3) B. Widrow and M.A. Lehr : "30 years of adaptive neural network :perceptron, madaline and backpropagation", Proc IEEE, vol.78, pp 1415 ~ 1442, Sept. 1990

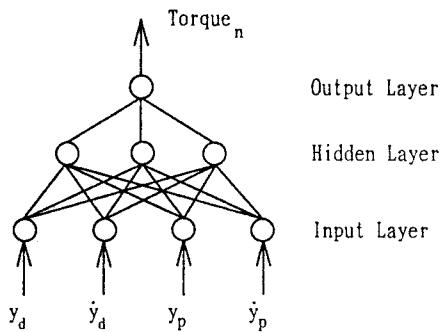


그림 2. 제안된 신경회로망 구조

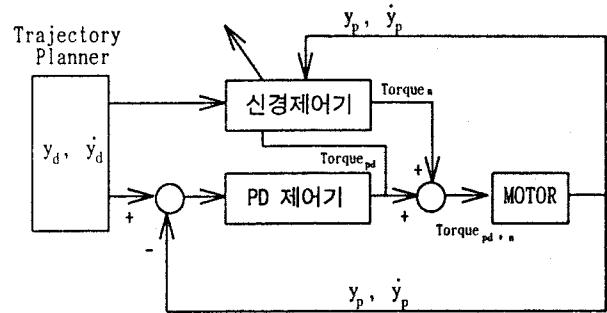


그림 3. 전체 시스템 구조

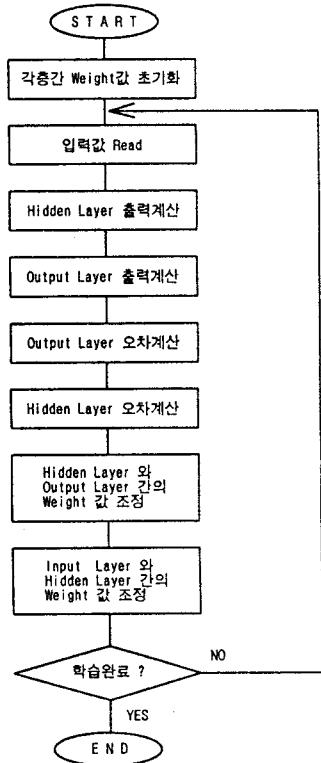


그림 4. 신경회로 제어기의 처리 ROUTINE

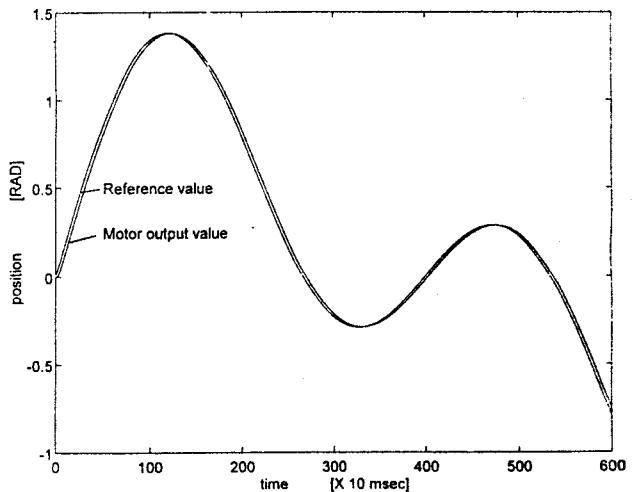


그림 5. PD 제어기의 경우

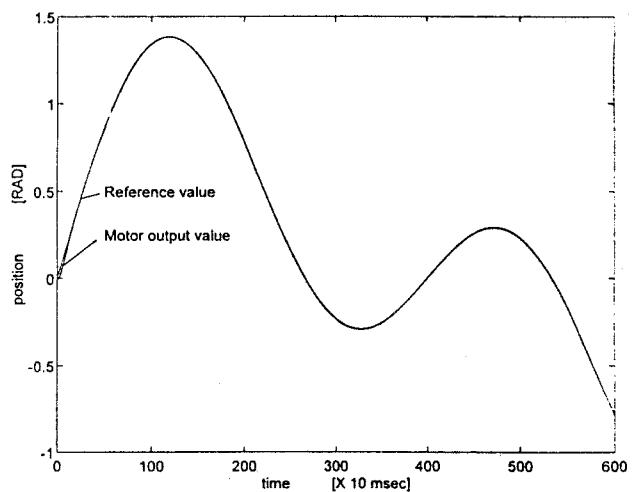


그림 6. 신경회로 제어기의 경우 : 25회 학습

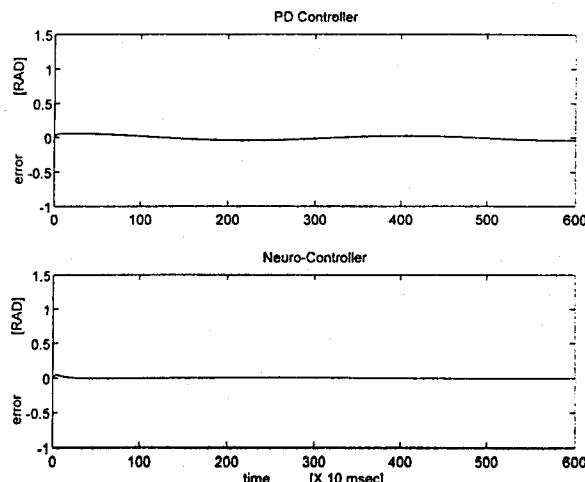


그림 7. PD 제어기와 신경회로 제어기의 잔류오차 비교

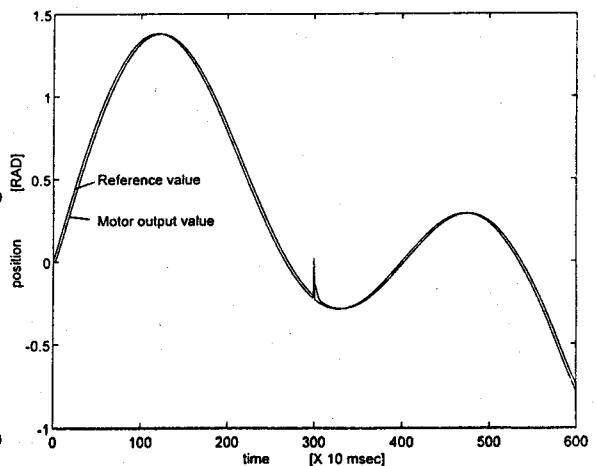


그림 8. PD 제어기의 경우 ( NOISE 인가시 )

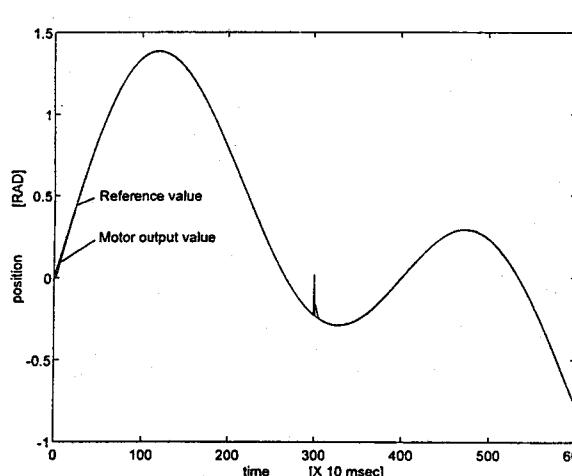


그림 9. 신경회로 제어기의 경우 : 25회 학습 ( NOISE 인가시 )

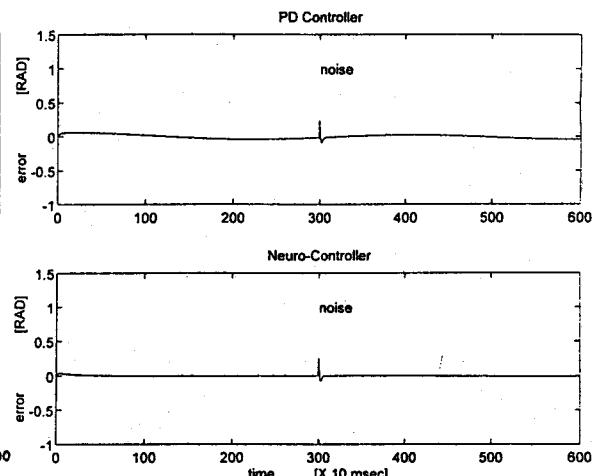


그림 10. PD 제어기와 신경회로 제어기의 잔류오차 비교  
( NOISE 인가시 )