

# 인공지능을 이용한 유압모터의 서보제어

## Servo Control of Hydraulic Motor using Artificial Intelligence

신 위 재\*, 허 태 욱\*\*

Wee-Jae Shin, Tae-Uk Hur

### 요 약

본 논문에서는 PID 제어기 응답을 보상하기 위해 자기구성 신경망 보상기를 추가한 제어기를 제안한다.

기존의 PID 제어기는 제어기 설계가 간단하나 계수값을 설정하는데 많은 시행착오가 필요하다. 그리고, 신경망 제어 방식은 여러 파라미터들을 설계자의 임의에 따라 결정함으로써 최적의 구조를 갖지 못하는 단점이 있다. 본 논문에서는 이러한 문제를 해결하기 위해 역전파 알고리즘을 기본으로 하여 은닉계층 노드의 활성화 함수로 가우시안 포텐셜 함수를 사용하는 자기구성 신경망을 사용해, PID 제어기의 출력을 보상하도록 하였다. 자기구성 신경망은 학습을 진행함에 따라 가우시안 함수의 위치와 모양, 갯수가 자동으로 조정 되도록 하였다.

자기구성 신경망 보상기를 추가한 PID 제어기의 성능을 확인하기 위해서 2차 플랜트에 적용하여 모의 실험하였으며 DSP 프로세서를 사용하여 제어기를 구현한 후 유압 서보시스템의 속도 제어에 적용하여 실험결과를 관찰하였다.

### Abstract

In this paper, we propose a controller with the self-organizing neural network compensator for compensating PID controller's response.

PID controller has simple design method but needs a lot of trials and errors to determine coefficients. A neural network control method does not have optimal structure as the parameters are pre-specified by designers. In this paper, to solve this problem, we use a self-organizing neural network which has Back Propagation Network algorithm using a Gaussian Potential Function as an activation function of hidden layer nodes for compensating PID controller's output. Self-Organizing Neural Network's learning is proceeded by Gaussian Function's Mean, Variance and number which are automatically adjusted.

As the results of simulation through the second order plant, we confirmed that the proposed controller get a good response compare with a PID controller. And we implemented the of controller performance hydraulic servo motor system using the DSP processor. Then we observed an experimental results.

*Key words* : self-organizing neural network

## I. 서 론

기존의 PID 제어방식은 제어기의 설계가 간단하나 목표값에 추종하도록 계수값을 설정하는데 많은 시행착오가 필요하다. 그리고, 신경망 제어방식은 비선형시스템을 모델링하고 근사화시킬 수 있으며, 비선형 활성화함수를 사용하여 전통적 제어방식으로 접근하지 못하는 난해한 비

선형 제어문제를 해결할 수 있으나, 학습시간이 많이 소요되고, 학습에 필요한 플랜트의 입출력 데이터를 구하기 힘들다.

본 논문에서는 자기구성 신경망 보상기를 사용한 PID 제어기를 제안한다. 먼저 PID 계수값을 적당히 설정한 후 플랜트 응답의 부족한 부분을 자기구성 신경망보상기를 사용하여 원하는 결과를 얻을 수 있도록 한다.

시뮬레이션은 2차 플랜트 전달함수에 적용하여 속도제어를 통해 일반적인 PID 제어기와 자기구성 신경망 보상기가 부가된 PID제어기의 제어결과를 비교 검토하고, 본 논문에서 제안한 자기구성 신경망 보상기가 부가된 PID 제어기가

\*경남대학교 전기전자 공학부 \*\*한국 소니 전자(주)  
 접수 일자 : 2003. 5. 24      수정 완료 : 2003. 7. 22  
 논문 번호 : 2003-2-14  
 ※이 논문은 2001년도 경남대학교 학술연구조성비 지원에 의해  
 석여진 것임

더욱 향상된 결과를 보임을 확인하였다. 설계된 제어를 DSP 프로세서를 사용하여 구현한 후 비선형 플랜트인 유압 서보시스템에 적용하여 그 성능을 확인하였다.

### II. 제어기 설계

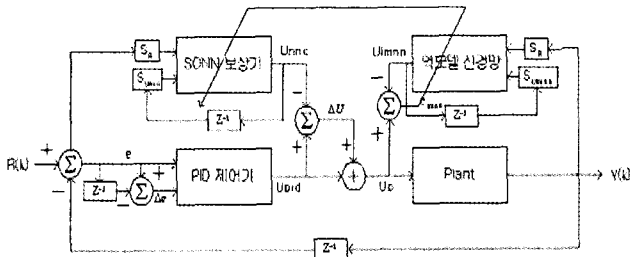


그림 1. 제어기 구조  
Fig. 1. Controller Structure

제안한 제어기는 그림 1과 같이 PID 제어기, 자기구성 신경망 보상기(SONN), 역모델 신경망(IMNN)으로 구성된다.

PID 제어기는 오차  $e(k)$ 와 오차 변화율  $\Delta e(k)$ 를 사용하여  $U_{pid}(k)$ 를 계산한다. 자기구성 신경망 보상기는 기준 입력  $R(k)$ 와 과거 보상기 출력  $U_{nnc}(k-1)$ 을 입력으로 하여 보상기의 출력  $U_{nnc}(k)$ 를 생성한다.

PID 제어기의 출력과 신경망 보상기의 출력을 이용하여 식 (1)과 (2)와 같이 플랜트의 입력을 계산한다.

$$\Delta U = U_{pid}(k) - U_{nnc}(k) \quad (1)$$

$$U_p(k) = U_{pid}(k) + \Delta U \quad (2)$$

역모델 신경망은 플랜트 출력  $Y(k)$ 와 과거의 역모델 신경망의 출력  $U_{imnn}(k-1)$ 을 입력으로 하여 역모델 신경망의 출력  $U_{imnn}(k)$ 를 생성한다. 그리고, 식(3)의 오차  $e_{imnn}(k)$ 를 이용하여 학습시킨다.

$$e_{imnn}(k) = U_p(k) - U_{imnn}(k) \quad (3)$$

그 후, 신경망 보상기의 파라미터를 역모델 신경망의 가중치로 대체한다. 이렇게 함으로써, 시간의 경과에 따라 신경망 보상기의 성능을 향상시킬 수 있다.

### III. 자기구성 신경망

#### 1. 신경망 구조

SONN 보상기와 역모델 신경망에 사용된 신경망의 구조를 그림 2와 그림 3에 보인다.

사용된 활성화함수는 가우시안 포텐셜함수로서 비선형 제어시스템에서 신경회로망의 활성화함수로 이용할 경우 양호한 학습결과를 얻을 수 있다.

$$\Psi_i = \Psi(x, p_i) = e^{-\alpha(x, p_i)^2} \quad (4)$$

$$\begin{aligned} d(x, p_i) &= d(x, m^i, K^i) \\ &= (x - m)K(x - m^i) \end{aligned} \quad (5)$$

여기서,  $x$ 는 입력 패턴이고,  $m^i$ 와  $K^i$ 는 평균벡터와 형상행렬을 표현하였고,  $i$  번째 포텐셜 함수  $d(x, m^i, K^i)$ 는 식 (6)와 같이 표현할 수 있다.

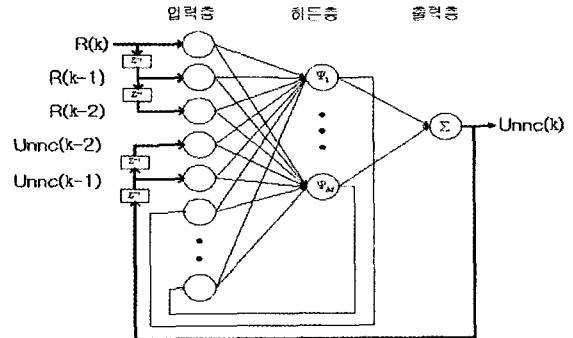


그림 2. SONN보상기 구조  
Fig. 2. The Structure of SONN Compensator

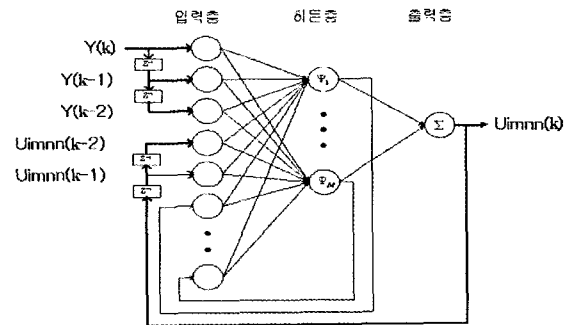


그림 3. 역모델 신경망 구조  
Fig. 3. The Structure of IMNN

$$d(x, m^i, K^i) = \sum_j \sum_k k_{jk}^i (x_j - m_j^i)(x_k - m_k^i) \quad (6)$$

여기서  $x_j$ 는  $j$  번째  $x$ 의 원소이고,  $m_k^i$ 는  $m^i$ 의  $k$  번째 원소를 표현한 것이다. 그리고,  $k_{jk}^i$ 는 다른 입력과의 상관계수로서  $K^i$ 의  $(j, k)$ 번째 원소이다. 일반적으로  $k_{jk}^i$ 는 식 (7)과 같이 표준편차  $\sigma_j^i$ 와  $\sigma_k^i$  그리고 공분산 상수  $h_{jk}^i$ 로 표현될 수 있다.

$$k_{jk}^i = \frac{h_{jk}^i}{\sigma_j^i \sigma_k^i} \quad (7)$$

$\sigma_j^i$ 는 양의 상수이고,

$$h_{jk}^i = \begin{cases} 1 & \text{if } j = k \\ |h_{jk}^i| \leq 1 & \text{otherwise} \end{cases} \quad (8)$$

따라서

$$k_{jk}^i = \begin{cases} \frac{1}{\sigma_i^2} & \text{if } j = k \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

최종적인 신경망의 출력은 식(10)과 같다.

$$y = \sum_{i=1}^n \psi_i w_i \quad (10)$$

2. 자기구성 신경망 제어기

자기구성 신경망 제어는 훈련된 표본의 예시 없이 제어 전략을 최적화하고 학습을 통해 신경망의 파라미터들을 자동으로 갱신하는 신경망 제어기 설계의 한 방법이다. 본 논문에서의 자기구성 신경망은 은닉계층의 활성화 함수로 GPF를 사용하는데 GPF의 파라미터인 평균과 분산, 그리고 신경망의 가중치, 학습률로 학습을 진행해 가면서 오차를 줄이는 방향으로 출력을 얻어내는 방법을 사용하고 있다.

GPF의 파라미터 조정은 입력공간에 정의되어진 적응경계(Accommodation boundary)와 출력공간에서 정의되어진 학습표현에 의해 이루어진다. 적응경계는 입력표본에 대하여 GPFU의 존재여부에 따라 GPFU를 새로이 생성시키거나 입력공간에 맞도록 GPFU를 변화시키는 역할을 하고, 출력공간에서 정의된 학습표현은 원하는 출력과 실제출력과의 차인 오차를 이용하여 GPFU의 변화량을 결정하는데 사용한다.

GPF를 사용한 자기구성 신경망의 기본적인 학습 방법은 다음 알고리즘과 같다.

알고리즘 1 : GPF 학습 알고리즘

① 은닉계층의 GPFU의 수를 최소화하여 입력 패턴과 적응경계의 상태에 따라 점차적으로 GPFU의 수를 늘려나가는 방법으로 학습을 진행해 나간다.

② 발생된 오차의 모양에 따라 은닉계층 GPFU의 파라미터 오차를 줄이는 방향으로 조정해 나간다.

GPF의 학습순서는 그림 4와 같다.

학습기의 성능파라미터를 P라 하면 식 (11)과 같이 표현할 수 있다.

$$P \equiv e^{-E_{rms}} \quad (11)$$

$$E_{rms} \equiv \sqrt{\frac{2}{MN} \sum_{p=1}^N E_p} \quad (12)$$

$$E_p \equiv \frac{1}{2} \sum_{j=1}^M (t_{pj} - \phi_{pj}(n_j))^2 \quad (13)$$

여기서 M은 출력노드의 수를 의미하며,  $t_{pj}$ 는 p 번째 교사패턴에 의해 정의되어진 출력패턴의 j 번째 원소를 의미한다.

GPFU의 갱신은 역전파(BPN) 방법을 기본으로 하여 앞에서 설명한 학습규칙에 의해 다음과 같이 한다. j 번째 GPFU의 출력  $n_j$ 는 아래 식과 같다.

$$n_j \equiv [w_j^i; m_j^i; \sigma_j^i; h_j^i]^t \quad (14)$$

파라미터 갱신은

$$n_j^{new} = n_j^{old} + \eta \Delta n_j \quad (15)$$

i 번째 GPFU와 j 번째 출력 사이의 가중치의 변화량은 다음과 같이 정의할 수 있다.

$$\Delta w_{ji} = -\frac{\partial E_p}{\partial w_{ji}} = (t - \phi_j) \psi_i \quad (16)$$

평균벡터의 j 번째 원소  $m^i$ 의 변화량은 다음과 같이 정의할 수 있다.

$$\begin{aligned} \Delta m_j^i &= -\frac{\partial E_p}{\partial m_j^i} \\ &= \sum_{l=1}^N k_{jl}^i (x_l - m_j^i) \psi_i \sum_{k=1}^M (t_k - \phi_k) w_{ki} \end{aligned} \quad (17)$$

표준편차  $\sigma_j^i$ 의 변화량은 다음과 같이 정의할 수 있다.

$$\begin{aligned} \Delta \sigma_j^i &= -\frac{\partial E_p}{\partial \sigma_j^i} \\ &= \sum_{l=1}^N k_{jl}^i \frac{(x_l - m_j^i)(x_l - m_j^i)}{\sigma_j^i} \psi_i \sum_{k=1}^M (t_k - \phi_k) w_{ki} \end{aligned} \quad (18)$$

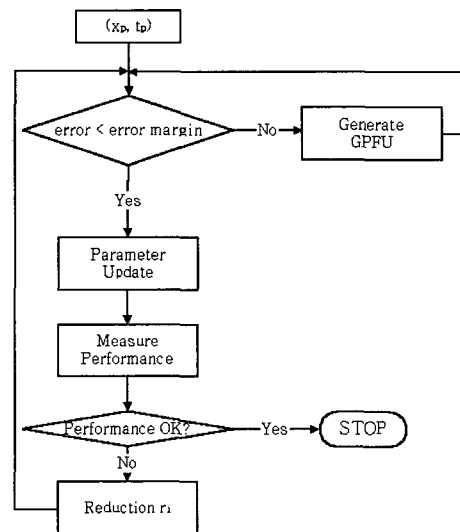


그림 4. GPF 학습방법 순서도

Fig. 4. Flowchart of GPF learning Method

알고리즘 2 : 자기구성 신경망(SONN) 학습 알고리즘

① 단계1 : 초기화

-  $i = 1, j = 1, p = 0$  (i: 학습수, j: 은닉계층 노드 수, p: 패턴수)

- 가우시안 포텐셜함수 초기화

· Mean :  $m^j = x_p$  (  $x_p$ : 입력표본)

· Deviation :  $K^j = \frac{1}{\sigma^2}$

· Weight :  $c_j = t_p$  ( $t_p$ : 설계값)

② 단계2 : 학습

1.  $p = p + 1$

$$2. \alpha(f) = \alpha(x, p_i) = e^{-\alpha(x, p_i)/2}$$

3. 만약  $|t_{pk} - G_{pk}(f)| > e_m$ 이면 새로운 은닉계층 노드 생성 ( $e_m$ :오차여유, 은닉계층 노드생성은 단계1의 방법과 동일하게 한다.)

4. 만약  $|t_{pk} - G_{pk}(f)| \leq e_m$ 이면 파라미터 갱신

5. 만약 모든 패턴에 대한 학습을 수행했으면 단계2의 1. 실행

1. 실행

③ 단계3 :  $i = i + 1$

④ 단계4 : 학습결과가 만족이면 정지, 그렇지 않으면 단계2 실행

#### IV. 모의 실험 및 결과 분석

제안한 제어기의 응답특성을 살펴보기 위해 다음의 2 차 플랜트 전달함수에 적용하여 모의 실험 하였다.

$$G(s) = \frac{1.2033 \times 10^8}{s^2 + 2.9627s + 6.5775 \times 10^6} \quad (19)$$

식(19)를 샘플링 시간 0.2ms로 이산화하여 차분방정식을 구한다.

$$y(k+1) = 1.39711y(k) - 0.538894y(k-1) + 2.59383u(k) \quad (20)$$

그림 5 와 그림 6 은 가변속일 때 PID 제어기와 자기 구성 신경망 보상기가 부가된 PID제어기 응답이고, 그림 7과 그림 8은 각각 학습 1회, 학습 4회일때 설정치 500rpm의 결과를 확대한 것이다.

표 1에서 PID 제어기와 자기구성 신경망을 부가하였을 때, 상승시간, 정정시간, 최대오버슈트의 결과를 비교하였다. PID 제어기보다 자기구성 신경망을 부가하였을 때, 응답 특성에서 더욱 향상된 결과를 보임을 알 수 있었다.

	상승시간	정정시간	최대 오버슈트
PID 제어기	0.89ms	5.4ms	9.75%
보상기가 부가된 PID 제어기(학습2회)	0.8ms	3.6ms	4.68%
보상기가 부가된 PID 제어기(학습3회)	0.6ms	2ms	0.02%
보상기가 부가된 PID 제어기(학습4회)	0.6ms	2ms	0.02%

표 1. 성능지수를 통한 응답특성 비교C(500rpm)

Tale 1. Comparison of performance through the response characteristic(500rpm)

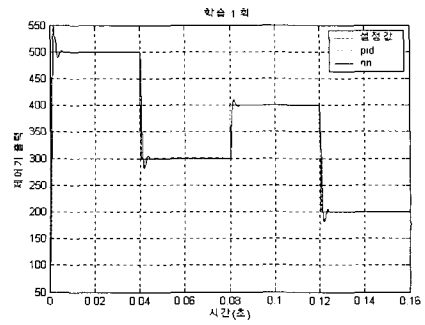


그림 5. PID제어기 응답

Fig. 5. PID Controller response

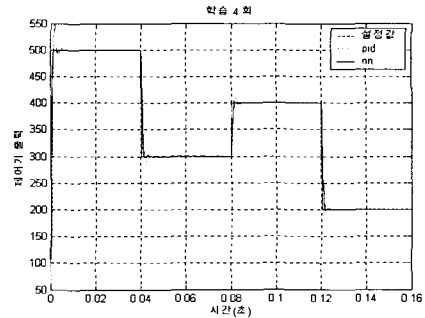


그림 6. 자기구성 신경망 보상기가 부가된 PID제어기 응답

Fig. 6. PID Controller response added The SONN Controller

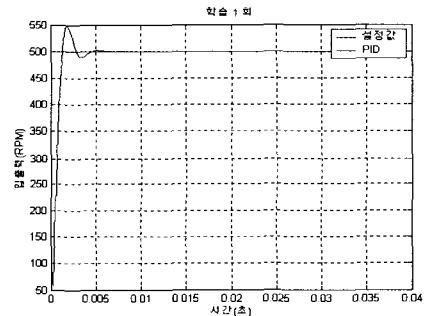


그림 7. PID제어기 응답(500rpm)

Fig. 7. PID Controller response(500rpm)

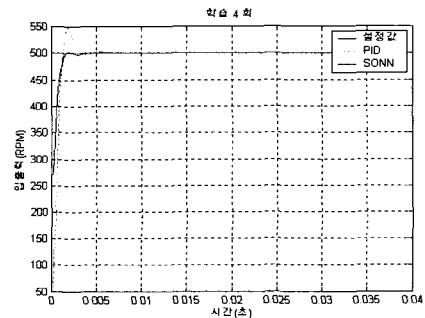


그림 8. 자기구성 신경망 보상기가 부가된 PID제어기 응답(500rpm)

Fig. 8. PID Controller response added SONN Compensator(500rpm)

### V. 실험 및 결과 고찰

본 논문에서 제안한 PID 보상 자기구성 신경망 제어를 이용한 속도제어기로서의 제어특성을 확인하기 위하여 비선형 유압 서보시스템에 적용하여 그 성능을 확인한다. 그림 9는 DSP 프로세서로 유압 서보모터의 속도제어 실험을 위한 전체 장치의 구성도이다.

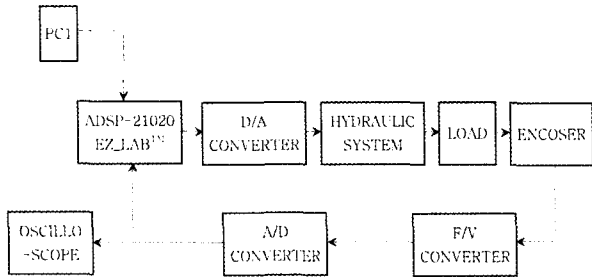


그림 9. 유압 서보 시스템 실험 구성도

Fig. 9. Experimental structure of hydraulic servo system

실험은 PID 제어기와 자기구성 신경망 보상기를 부가한 PID 제어기를 무부하 상태에서 유압 서보시스템에 적용하여 비교하였다. 유압모터의 축에 스프링 커플러를 연결하여 사용하였다. 유압 서보시스템 실험장치는 서보 밸브를 100% 열고 유압펌프의 압력을 최대로 하면 250 [rpm]으로 모터가 회전하지만, 실험은 공급 유압 70[kg/cm<sup>2</sup>]으로 유지하여 수행하였다.

그림 10과 그림 11에서 리플이 주기적으로 좀 더 크게 나타나는 것은 스프링 커플러에 의한 것이고, 큰 리플 속에 포함된 작은 리플은 제어기에서 디지털 회로가 동작할 때 나타나는 스위칭 잡음으로서 ADSP-21020 EZ-LAB™ DSP 보드가 제어 알고리즘을 수행할 때 DSP 프로세서에 의해서 유도되는 잡음이다. 필요한 노이즈 대책을 적용하여 잡음을 많이 제거하였기 때문에 측정 데이터를 확대하여 보면 A/D 변환시 발생하는 양자화 잡음으로 생각해도 무방하다.

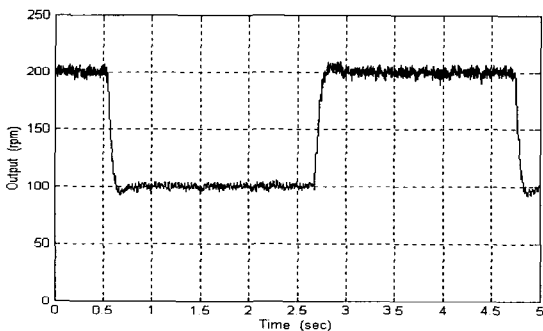


그림 10. PID 제어기의 속도 응답

Fig. 10. Speed response PID Controller

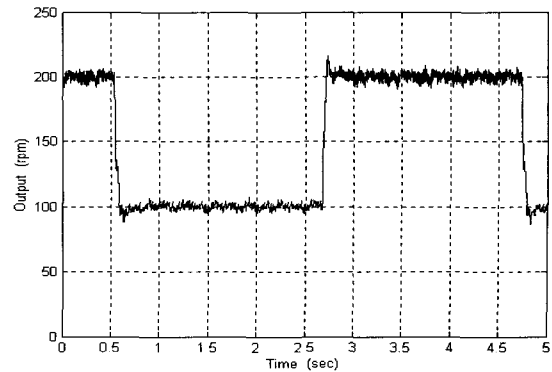


그림 11. 자기구성 신경망 보상기를 부가한 PID제어기의 속도응답

Fig. 11. Speed response of PID Controller added SONN Compensator(500rpm)

표 2의 실험 결과를 살펴보면 PID 제어기보다 자기구성 신경망 보상기가 부가된 PID 제어기가 오버슈트는 조금 더 있지만 정정시간과 상승시간에서 더 나은 결과를 얻을 수 있었다.

표 2. 제안한 제어기의 실험결과

Table 2. Experimental results of Proposed Controller

성능지수	상승시간	정정시간	최대 오버슈트
PID 제어기	66 ms	140 ms	4.7 %
자기구성 신경망 보상기가 부가된 PID 제어기	40 ms	107 ms	6.7 %

### VI. 결론

제안한 제어기는 PID 제어기에 자기구성 신경망을 부가하여 PID 제어기의 출력을 보상하도록 하였다. 학습이 부족한 신경망 제어기의 성능을 온라인으로 향상시키기 위해 플랜트의 역모델 신경망을 학습시킨 결과를 이용하여 자기구성 신경망 보상기의 가중치를 변경시킴으로써 원하는 플랜트의 동적특성을 얻도록 하였다.

신경망 제어기는 여러 종류가 있으나 비선형 시스템의 제어에 유용한 것으로 알려진 GPF를 활성화함수로 사용하는 신경망을 도입하여 신경망의 학습에서 문제가 되는 극부 최소점에 빠지는 것을 막기 위해 GPFU 파라미터를 자동적으로 조정하였으며, 아울러 학습속도 개선을 위해 은닉계층의 노드수를 초기학습 시에는 최소로 하고, 학습을 진행해 감에 따라 노드 수를 증가시키는 자기구성 신경망을 사용하였다.

비선형 유압 서보모터의 실시간 제어를 위해 DSP 프로세서를 이용하여 속도 제어기를 구현하였으며 본 논문

에서 제안한 제어기의 성능을 실험을 통하여 확인할 수 있었다.

### VI. 참고 문헌

[1] M.M. Gupta, D.H.Rao, "Neuro-Control Systems: A Tutorial, Neuro-Control Systems (Theory and Application) Edited by M.M. Gupta, D.H.Rao", New York, IEEE Press, pp.1~43, 1994

[2] J.M.Zurada, "Introduction to Artificial Neural Systems", St.Paul, West Pub.Com.,pp.1~298, 1992

[3] 이상배, "퍼지-뉴로제어 시스템", (주)교학사, pp.111~200, 1999

[4] Lefteri,H.Tsoukalas, Robert E.Uhrig, "Fuzzy and Neural Approaches in Engineering ", John Wiley & Sons, Inc., pp.13~405, 1997

[5] Simon Haykin, "Neural Networks, A comprehensive foundation ,Second Edition", Prentice Hall International, Inc., pp.1~255, pp.664~789, 1999

[6] Michael A.Arbib, "The Handbook of Brain Theory and Neural Networks", The MIT Press, pp.19~20, 1995

[7] Madan M.Gupta, Dandina H.Rao, "Dynamic Neural Processor and Its Applications to Robotics and Control: Intelligent Control Systems(Theory and Applications) Edited by Madan M.Gupta, Naresh K.Sinha", IEEE Press, pp.515~545, 1996

[8] Peter Vas, "Artificial-Intelligence-Based Electrical Machines and Drives", Oxford University Press, Inc., pp.77~234, 1999

[9] 이상윤, "지능구조를 갖는 궤환제어기 설계에 관한 연구", 박사학위논문, 경남대학교 대학원 전자공학과, 1998

[10] 김동희, "자기구성 신경회로망가 퍼지보상기를 이용한 역기구학해에 관한 연구", 박사학위논문, 경남대학교 대학원 전자공학과, 2001



허 태 욱(Tae-Uk Hur)

正會員

1995년 경남대학교 전자공학과 졸업

2002년 동대학원 전자공학과(공학석사)

2002년 한국소니전자(주)상품개발과 근무



신 위 재(Wee-jae Shin)

正會員

1975년 동아대학교 전자공학과

1979년 동대학원 전자공학과(공학석사)

1989년 동대학원 전자공학과(공학박사)

1996년-1997년 대한전자공학회 부산경남 지부장

1998년-1999년 대한전자공학회 학술위원

1998년 일본 Kyushu 대학 객원교수

2000년-현재 한국신호처리·시스템학회 부회장

1983년-현재 경남대학교 전기전자공학부 교수

관심분야 : 시스템지능제어 및 자동화, 신호처리