

# 퍼지 보상기를 사용한 리커런트 시간지연 신경망 제어기 설계 및 구현

## Design and Implementation of Recurrent Time Delayed Neural Network Controller Using Fuzzy Compensator

이상윤\*, 신위제\*\*

\*경남대학교 기계자동화공학부, \*\*경남대학교 전기전자공학부

### 요 약

본 논문에서는 신경망제어기의 출력을 보상하는 퍼지보상기를 갖는 리커런트 시간 지연 신경망(RTDNN) 제어기를 제안하였다. 학습이 완료된 신경망제어기를 사용하더라도 예상치 못한 외란으로 인해 플랜트의 출력이 좋지 못한 경우가 있는데, 이것을 적절하게 조절해 주기 위해 퍼지보상기를 사용하여 원하는 결과를 얻을 수 있도록 하였다. 그리고 플랜트의 역모델 신경망을 학습시킨 결과를 이용하여 주 신경망의 가중치를 변경시킴으로써 원하는 플랜트의 동적 특성을 얻게 된다. 2차 플랜트를 통한 모의실험 결과가 시간 지연 신경망(TDNN)제어기보다 더 좋은 응답 특성을 가짐을 확인할 수 있다. 제안한 제어기의 성능을 확인하기 위해 유압 서보시스템을 대상으로 DSP 프로세서를 사용하여 구현한 후 실험결과를 통하여 제안된 방법의 유용성을 보였다.

### Abstract

In this paper, we proposed a recurrent time delayed neural network(RTDNN) controller which compensate a output of neural network controller. Even if learn by neural network controller, it can occur an bad results from disturbance or load variations. So in order to adjust above case, we used the fuzzy compensator to get an expected results. And the weight of main neural network can be changed with the result of learning a inverse model neural network of plant, so a expected dynamic characteristics of plant can be got. As the results of simulation through the second order plant, we confirmed that the proposed recurrent time delayed neural network controller get a good response compare with a time delayed neural network(TDNN) controller. We implemented the controller using the DSP processor and applied in a hydraulic servo system. And then we observed an experimental results.

**Key Words** : neural network controller, fuzzy compensator, FLC, TDNN, RTDNN

## 1. 서 론

초기 신경망의 응용분야는 패턴 인식, 음성 인식, 화상 판단과 같은 신호처리 분야에 집중되었으나, 비선형시스템을 모델링하고 근사화시킬 수 있는 능력으로 인해 현재 많은 제어시스템에 적용되고 있다.[1][2]

신경망은 비선형 시그모이드 활성화함수를 사용하여 전통적 제어법으로 접근하지 못하는 난해한 비선형 제어문제를 해결하기 위한 비선형 사상능력을 제공해 준다.[2] 하지만, 학습시간이 많이 소요되고, 학습에 필요한 플랜트의 입·출력 데이터를 필요로 한다. 또한, 학습이 완료되어진 신경망을 사용하더라도 예상치 못한 외란으로 인해 플랜트의 출력이 좋지 못한 경우도 있다.

퍼지제어기는 제어대상에 대한 작업자의 경험이나 전문가의 언어적인 표현을 쉽게 제어기와 결합시킬 수 있는 장점을 지니고 있다. 또한 플랜트 파라미터 변동이나 외부 외란

에 크게 영향을 받지 않는 강인한 제어기의 특성도 지니고 있다.[3]

본 논문에서는 신경망과 퍼지이론의 장점을 이용하여 학습된 신경망제어기의 플랜트 출력이 외란으로 인해 좋지 못한 결과를 가져올 때, 이를 퍼지제어기를 응용한 퍼지보상기를 사용하여 원하는 결과를 얻을 수 있도록 하는 제어기[4]에서 적용한 시간지연 신경망을 리커런트 시간 지연 신경망으로 대체하여 제어성능이 개선됨을 보였다. 그리고 설계된 제어기를 DSP 프로세서를 사용하여 구현한 후 비선형 플랜트인 유압 서보시스템에 적용하여 그 성능을 확인하였다.

## 2. 신경망 제어기설계

본 논문에서 신경망 제어기는 그림 1과 같이 주 제어기인 신경망제어기와 신경망제어기의 출력을 보상시켜주는 퍼지보상기, 플랜트의 역모델을 학습하는 역모델 신경망(Inverse Model Neural Network: IMNN)으로 구성된다.

신경망제어기는 기준 입력  $r(n)$ 와 과거의 플랜트의 입력  $u_p(n-1)$ 을 이용하여 제어입력  $u_{nnc}(n)$ 을 생성한다.

접수일자 : 2002년 9월 26일

완료일자 : 2003년 5월 9일

퍼지보상기는 기준 입력  $r(n)$ 과 과거의 플랜트의 출력  $y(n-1)$ 사이의 오차  $e_{flc}(n)$ 와 오차 변화율  $\Delta e_{flc}(n)$ 을 사용하여 또 다른 제어입력  $u_{flc}(n)$ 을 계산한다.

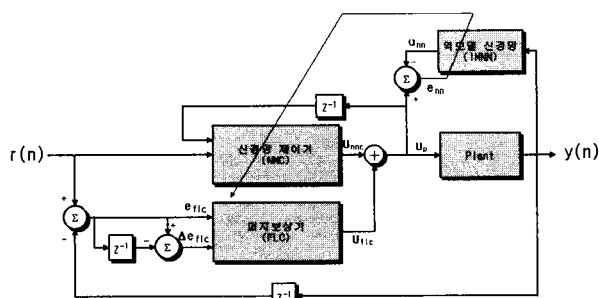


그림 1. 퍼지 보상기를 가지는 신경망 제어기 구조.  
Fig. 1. The structure of neural network controller within fuzzy compensator.

$$e_{flc}(n) = r(n) - y(n-1) \quad (1)$$

$$\Delta e_{flc}(n) = e_{flc}(n) - e_{flc}(n-1) \quad (2)$$

신경망제어기의 출력  $u_{nnc}(n)$ 와 퍼지보상기의 출력  $u_{flc}(n)$ 의 합이 플랜트의 입력  $u_p(n)$ 가 된다.

$$u_p(n) = u_{nnc}(n) + u_{flc}(n) \quad (3)$$

플랜트의 출력  $y(n)$ 을 입력으로 한 역모델 신경망의 출력  $u_{nm}(n)$ 와 플랜트 입력  $u_p(n)$ 사이의 오차  $e_{nm}(n)$ 을 이용하여 역모델 신경망을 학습시킨 후, 역모델 신경망의 가중치를 신경망제어기의 가중치로 대체시킨다. 이렇게 함으로써, 학습이 진행됨에 따라 신경망제어기의 성능을 향상시킬 수 있다.

### 2.1 신경망 구조와 학습법

신경망 제어기에는 시간지연 신경망(Time Delayed

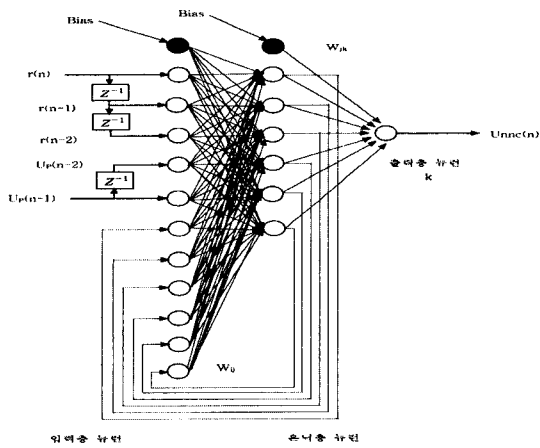


그림 2. 신경망제어기  
Fig. 2. The neural network controller.

Neural Network: TDNN)[5][6]을 개선한 리커런트 시간지연 신경망(Recurrent Time Delayed Neural Network: RTDNN)을 사용하였으며, 역모델 신경망은 RTDNN을 이용

하여 구성하였다. 사용하는 신경망은 시간지연과 리커런트 기능을 갖기 때문에, 신경망 구조를 차분방정식 또는 순환방정식으로 기술할 수 있다. 따라서, 시간에 따라 변하는 비선형 동적 시스템을 모델링하기에 유리하다.[7][8] 그리고 플랜트 역모델을 온-라인으로 학습시킬 수 있기 때문에, 학습이 부족한 신경망제어기의 성능을 온-라인으로 보완시킬 수 있다. 그림 2는 신경망제어기로 사용된 리커런트 시간지연신경망으로 기준 입력  $r(n)$ , 기준입력의 1차 지연  $r(n-1)$ , 기준입력의 2차 지연  $r(n-2)$  그리고 과거의 플랜트의 입력의 1차 지연  $u_p(n-1)$ , 과거의 플랜트의 입력의 2차 지연  $u_p(n-2)$ 을 이용하여 제어입력  $u_{nnc}(n)$ 을 생성한다.

학습 알고리즘은 일반적으로 신경망을 학습시키는데 사용하는 오류 역전파 학습 알고리즘을 사용하였다. 그리고 지역 극소점에 빠지는 것을 방지하고 빠른 학습을 위하여 모멘텀 방법을 사용하였으며 가중치 조절식은 다음과 같다.

$$\Delta W_{ij}(n+1) = 2\eta e_{nm}(n)y(n)(1-y(n))h(n)(1-h(n))W_{jk}(n)x(n) + \alpha \Delta W_{ij}(n) \quad (4)$$

$$\Delta W_{jk}(n+1) = 2\eta e_{nm}(n)y(n)(1-y(n))h(n) + \alpha \Delta W_{jk}(n) \quad (5)$$

$$W_{ij}(n+1) = W_{ij}(n) + \Delta W_{ij}(n+1) \quad (6)$$

$$W_{jk}(n+1) = W_{jk}(n) + \Delta W_{jk}(n+1) \quad (7)$$

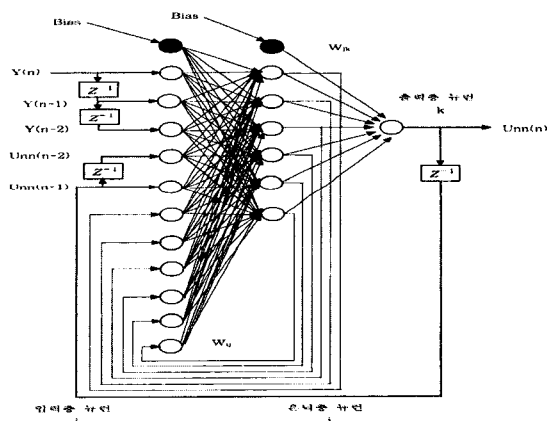


그림 3. 역모델 신경망.  
Fig. 3. The Inverse model of neural network.

본 논문에서는 학습률  $\eta$ , 모멘텀 상수  $\alpha$ 를 각각 0.4와 0.7로 설정하였다.  $x(n)$ ,  $h(n)$ ,  $y(n)$ 는 각각 입력층 뉴런값, 은닉층 뉴런값, 출력층 뉴런값을 나타내며  $W_{ij}(n)$ 는 입력층과 은닉층사이의 가중치,  $W_{jk}(n)$ 는 은닉층과 출력층사이의 가중치를 나타낸다.

역모델 신경망은 그림 3의 리커런트 TDNN 신경망을 사용하였으며, 구조상으로 신경망제어기와 차이점은 없다. 역모델 신경망은 그림 2의 신경망 제어기에서 기준 입력  $r(n)$ , 기준입력의 1차 지연  $r(n-1)$ , 기준입력의 2차 지연  $r(n-2)$  그리고 과거의 플랜트의 입력의 1차 지연  $u_p(n-1)$ , 과거의 플랜트의 입력의 2차 지연  $u_p(n-2)$  대신에 현재의 플랜트 출력  $y(n)$ , 플랜트 출력의 1차 지연  $y(n-1)$ , 플랜트 출력의 2차 지연  $y(n-2)$  그리고 신경망제어기의 제어출력의 1차 지

연  $u_{nnc}(n-1)$ , 신경망 제어기의 제어출력의 2차 지연  $u_{nnc}(n-2)$ 을 이용하여 신경망 제어기의 제어출력  $u_{nnc}(n)$ 을 학습한다.

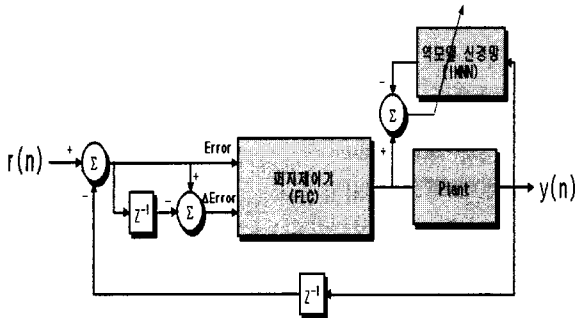


그림 4. 퍼지제어기를 사용한 플랜트 역모델 학습 구조.  
Fig. 4. The learning structure of inverse model by the fuzzy controller.

신경망 제어기가 플랜트를 제어하기 위해서는 먼저 플랜트의 역모델 신경망을 학습 시켜야 한다. 퍼지제어기를 사용하여 역모델 신경망을 학습하기 위한 구조는 그림 4와 같다. 초기 역모델 신경망을 학습시키기 위해 주 제어기는 퍼지 제어기가 담당한다. 그리고 퍼지제어기의 제어 알고리즘을 역모델 신경망에 학습시킨다. 학습이 완료된 후 역모델 신경망은 학습한 가중치를 신경망제어기에 전달한다. 가중치를 전달받은 신경망 제어기는 그림 1과 같은 퍼지보상기를 갖는 신경망제어기로 동작하게 된다. 퍼지보상기를 갖는 신경망제어기로 동작하면서 보다더 성능을 개선하기 위해(오차를 줄이기 위해) 학습을 병행하면서 제어기로 동작한다. 학습한 결과인 역모델의 가중치는 신경망제어기로 매 순간 학습이 끝날 때마다 전달된다.

2.2 퍼지보상기

그림 5는 퍼지보상기의 구조를 나타낸다. 퍼지보상기는 퍼지화기(Fuzzifier), 퍼지추론(Fuzzy reasoning), 비퍼지화기로 구성되며 퍼지추론 메카니즘을 기반으로 한다. 추론 메커니즘을 '추론엔진' 혹은 '퍼지추론'이라고 부른다.[9][10] 플랜트를 최상으로 제어하기 위한 방법에 대한 지식을 해석하고 적용하여 마치 전문가에 의해 내리는 결정처럼 흉내를 내는 역할을 하게 된다. 추론법에는 무한치 논리를 기반으로 하는 추론법과 퍼지논리를 기반으로 하는 추론법이 있으며, 시스템 제어에는 계산이 비교적 간단한 무한치 논리를 기반으로 하는 추론법이 많이 사용된다. 무한치 논리를 기반으로 하는 추론법에는 합성에 의한 방법, Tsukamoto법, TSK법이 있다. 합성에 의한 추론법은 Zadeh의 방법, Mamdani의 방법, Larson의 방법 등이 있다.[11][12]

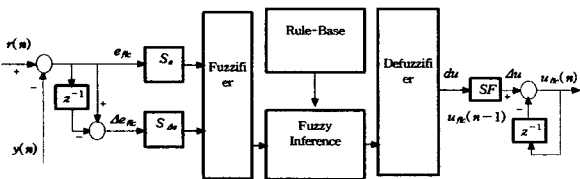


그림 5. 퍼지보상기 구조.  
Fig. 5. The structure of fuzzy compensator.

본 논문에서 사용된 퍼지보상기에는 다음과 같은 퍼지추론과 비퍼지화 과정이 사용되었다.

MISO(Multi Input Single Output) LMs(Linguistic Models)은 다음과 같다.

$$\text{IF } e_{flc} \text{ is Bi1 AND } \Delta e_{flc} \text{ is Bi2 THEN V is Di, } i=(1, m) \quad (8)$$

간략화된 퍼지 추론법에 따른 퍼지모델의 crisp출력은 면적 중심법(The center of area method : COA method)을 사용한다.[13]

$$du = \frac{\sum_{i=1}^m \tau_i du^*_i}{\sum_{i=1}^m \tau_i} \quad (9)$$

여기서,  $du^*_i$ 는 결론부 퍼지집합 Di의 비퍼지화된 값이다.

$\tau_i$ 는 주어진 crisp 입력값  $e_{flc}$ ,  $\Delta e_{flc}$ 에 대한 규칙의 적합도이다.

$$\tau_i = B_{i1}(e_{flc}) \wedge B_{i2}(\Delta e_{flc}) \quad (10)$$

간략화된 퍼지 추론방법은 개별적인 규칙들의 적합도 함수  $\tau_i$ 로 퍼지모델의 crisp 출력은 해석적 수식으로 된다. 그러나 이것은 crisp 입력값  $e_{flc}$ 과  $\Delta e_{flc}$ 으로 이루어진 함수는 아니다. 식(10)에서 해석적 수식을 얻기 위해 min operator를 product로 교체하고 전제 퍼지집합은 파라미터가  $x_i^*$ 와  $\sigma_i^*$ 를 갖는 가우스 소속함수로 가정한다면  $\tau_i$ 는 다음과 같이 재정리하여 적을 수 있다.

$$\begin{aligned} \tau_i &= B_{i1}(e_{flc}) \cdot B_{i2}(\Delta e_{flc}) \\ &= \left[ \exp\left(-\frac{1}{2} \left(\frac{e_{flc} - x_i^*}{\sigma_i}\right)^2\right) \cdot \exp\left(-\frac{1}{2} \left(\frac{\Delta e_{flc} - x_i^*}{\sigma_i}\right)^2\right) \right] \end{aligned} \quad (11)$$

여기서  $x_i^*$ 와  $\sigma_i^*$ 는 각각 가우스 함수의 중심과 폭이다. 식(11)을 식(9)에 대입하여 전제부와 결론부 퍼지집합의 파라미터  $du^*_i$ ,  $x_i^*$ 와  $\sigma_i^*$ ,  $i=(1, m)$ 에 의해 결정되는 LM의 crisp출력은 다음과 같다.

$$\begin{aligned} du &= \frac{\sum_{i=1}^m \tau_i du^*_i}{\sum_{i=1}^m \tau_i} \\ &= \frac{\sum_{i=1}^m du^*_i \left[ \exp\left(-\frac{1}{2} \left(\frac{e_{flc} - x_i^*}{\sigma_i}\right)^2\right) \cdot \exp\left(-\frac{1}{2} \left(\frac{\Delta e_{flc} - x_i^*}{\sigma_i}\right)^2\right) \right]}{\sum_{i=1}^m \left[ \exp\left(-\frac{1}{2} \left(\frac{e_{flc} - x_i^*}{\sigma_i}\right)^2\right) \cdot \exp\left(-\frac{1}{2} \left(\frac{\Delta e_{flc} - x_i^*}{\sigma_i}\right)^2\right) \right]} \\ &= \sum_{i=1}^m v_i du^*_i \end{aligned} \quad (12)$$

여기서  $v_i$ 는 다음과 같다.

$$v_i = \frac{\tau_i}{\sum_{i=1}^m \tau_i} = \frac{\left[ \exp\left(-\frac{1}{2} \left(\frac{e_{flc} - x_i^*}{\sigma_i}\right)^2\right) \cdot \exp\left(-\frac{1}{2} \left(\frac{\Delta e_{flc} - x_i^*}{\sigma_i}\right)^2\right) \right]}{\sum_{i=1}^m \left[ \exp\left(-\frac{1}{2} \left(\frac{e_{flc} - x_i^*}{\sigma_i}\right)^2\right) \cdot \exp\left(-\frac{1}{2} \left(\frac{\Delta e_{flc} - x_i^*}{\sigma_i}\right)^2\right) \right]}$$

위에서 설명한 퍼지 추론과 비퍼지화 방법을 블록다이어그램으로 정리하면 그림 6과 같이 나타낼수 있다. 그림 5에서 스케일링 계수 SF를 고려한 퍼지 보상기의 보상 증분량  $\Delta u$ 는 식(13)와 같이 표현된다.

$$\Delta u = SF \times du \quad (13)$$

퍼지 보상기의 보상 증분량은 -1과 +1사이의 값을 가지게 된다. 이 보상 증분량은 플랜트의 구동에 필요한 제어량을 만족시키지 못한다. 그러므로 퍼지 보상기의 보상 증분량과 플랜트의 구동에 필요한 구동량 사이의 비율을 맞추기 위해 적당한 상수 값을 곱하여 주는데 이 값이 스케일링 계수 SF이다.

그림 5에서와 같이 퍼지 보상기 출력  $u_{flc}(n)$ 는 식(14)와 같이 1차 지연된 퍼지 보상기 출력과 식(13)의 보상 증분량의 합으로 구성된다.

$$u_{flc}(n) = u_{flc}(n-1) + \Delta u \quad (14)$$

그림 5의 퍼지 보상기 입력부에서 오차  $e_{flc}$ 와 오차의 변화율  $\Delta e_{flc}$  다음에 나오는  $S_e, S_{\Delta e}$ 는 오차와 오차의 변화율을 -1과 +1사이의 값으로 정규화 시키는 부분으로 식(15)로 표현 할 수 있다.

$$S_e = S_{\Delta e} = \frac{1}{R - R_{old}} \quad (15)$$

- 여기서  $R$  : 설정치,  $R_{old}$  : 이전 설정치
- $S_e$  :  $e_{flc}$ 을 [-1, 1]로 정규화하기 위한 스케일링 계수
- $S_{\Delta e}$  :  $\Delta e_{flc}$ 을 [-1, 1]로 정규화하기 위한 스케일링 계수

설정치  $R$ 과 이전 설정치  $R_{old}$ 는 그림 1에서 제어기의 기준 입력  $r(n)$ 의 값이 변화를 일으킬 때 그 전과 후의  $r(n)$ 의 값을 나타낸다. 기준입력  $r(n)$ 의 값이 변화를 일으키면 설정치  $R_{old}$ 은 변화를 일으키기 전의 기준 입력  $r(n)$  값이고, 설정치  $R$ 은 변화를 일으킨 후의 기준 입력  $r(n)$ 의 값이다.

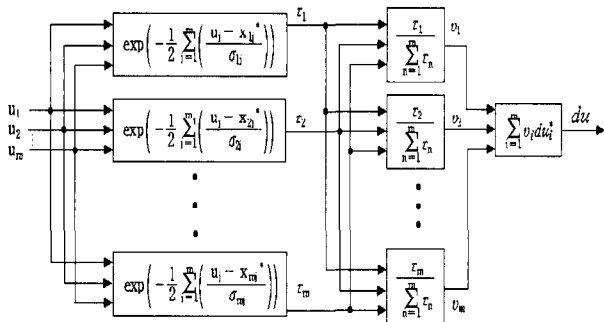


그림 6. 퍼지추론과 비퍼지화.

Fig. 6. The fuzzy inference and defuzzifier.

퍼지 보상기에서 사용한 제어 규칙과 소속함수는 기준 입력이 변경된 후, 플랜트의 출력에서 첫 번째 최대 오버슈트를 전후로 하여 제어규칙과 소속함수가 변경되도록 하였다.

첫 번째 오버슈트가 발생하기 전에는 제어규칙1(표 1 참조)을 사용하여 오버슈트를 감소시킨다. 이때, 사용한 오차  $e_{flc}$ 와 오차변화율  $\Delta e_{flc}$ , 플랜트 입력  $u_{flc}$ 의 소속함수는 그림 7과 같은 것을 사용하였다.

첫 번째 오버슈트 발생 이후에는 제어규칙2(표 2 참조)를 사용하게 되고, 이때 오차  $e_{flc}$ 와 오차변화율  $\Delta e_{flc}$ 의 소속함수는 그림 8과 같으며 플랜트 입력  $u_{flc}$ 의 소속함수는 그림 7과 같다. 제어규칙2는 설정치에 더 빨리 도달할 수 있도록

해준다. 그리고 제어규칙 전환시 불연속은 신경망제어기에 의해서 제어된다. 퍼지추론에 사용한 소속함수는 가우스 함수를 사용하였다.

표 1. 제어규칙 1.

Table 1. The control rule 1.

E ΔE	NB	NMB	NM	NMS	NS	NVS	ZE	PVS	PS	PMS	PM	PMB	PB
NB	NB	NB	NB	NB	NB	NB	NB	NMB	NM	NMS	NS	NVS	ZE
NMB	NB	NB	NB	NB	NB	NB	NMB	NM	NMS	NS	NVS	ZE	PVS
NM	NB	NB	NB	NB	NB	NMB	NM	NMS	NS	NVS	ZE	PS	PS
NMS	NB	NB	NB	NB	NMB	NM	NMS	NS	NVS	ZE	PVS	PS	PS
NS	NB	NB	NB	NMB	NM	NMS	NS	NVS	ZE	ZE	PS	PS	PS
NVS	NB	NB	NVS	NVS	NVS	NVS	ZE	ZE	PVS	PVS	PVS	PS	PS
ZE	NS	NS	NVS	NVS	NVS	ZE	ZE	ZE	PVS	PVS	PVS	PS	PS
PVS	NS	NS	NVS	NVS	NVS	ZE	ZE	PVS	PVS	PVS	PVS	PB	PB
PS	NS	NS	ZE	NVS	ZE	PVS	PS	PMS	PM	PMB	PB	PB	PB
PMS	NS	NS	ZE	ZE	PVS	PS	PMS	PM	PMB	PB	PB	PB	PB
PM	NS	ZE	ZE	PVS	PS	PMS	PM	PMB	PB	PB	PB	PB	PB
PMB	NVS	ZE	PVS	PS	PMS	PM	PMB	PB	PB	PB	PB	PB	PB
PB	ZE	PVS	PS	PMS	PM	PMB	PB	PB	PB	PB	PB	PB	PB

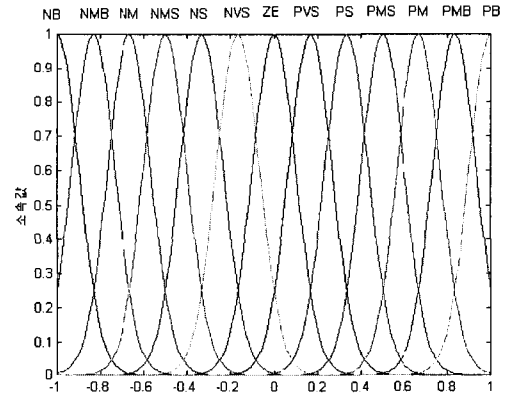


그림 7. 첫 번째 오버슈트에 도달하기 전에 사용된 오차( $e_{flc}$ )와 오차변화율( $\Delta e_{flc}$ ) 소속함수와 플랜트 입력( $u_{flc}$ )소속함수.

Fig. 7. Before arrive at the first overshoot, used to error( $e_{flc}$ ) and change rate of error membership function( $\Delta e_{flc}$ ), plant input membership function ( $u_{flc}$ ).

표 2. 제어규칙2

Table 2. The control rule 2

E ΔE	NB	NMB	NM	NMS	NS	NVS	ZE	PVS	PS	PMS	PM	PMB	PB
NB	NB	NB	NB	NB	NB	NB	NB	NMB	NM	NMS	NS	NVS	ZE
NMB	NB	NB	NB	NB	NB	NB	NMB	NM	NMS	NS	NVS	ZE	PVS
NM	NB	NB	NB	NB	NB	NMB	NM	NMS	NS	NVS	ZE	PVS	PS
NMS	NB	NB	NB	NB	NMB	NM	NMS	NS	NVS	ZE	PVS	PS	PMS
NS	NB	NB	NB	NMB	NM	NM	NM	NM	ZE	PM	PS	PMS	PM
NVS	NB	NB	NMB	NM	NM	NVS	NM	ZE	PM	PM	PMS	PM	PMB
ZE	NB	NMB	NM	NMS	NM	NVS	ZE	PVS	PM	PM	PM	PMB	PB
PVS	NMB	NM	NMS	NS	NM	ZE	PM	PVS	PM	PM	PMB	PB	PB
PS	NM	NMS	NS	NVS	ZE	PM	PM	PM	PM	PMB	PB	PB	PB
PMS	NMS	NS	NVS	ZE	PVS	PS	PMS	PM	PMB	PB	PB	PB	PB
PM	NS	NVS	ZE	PVS	PS	PMS	PM	PMB	PB	PB	PB	PB	PB
PMB	NVS	ZE	PVS	PS	PMS	PM	PMB	PB	PB	PB	PB	PB	PB
PB	ZE	PVS	PS	PMS	PM	PMB	PB	PB	PB	PB	PB	PB	PB

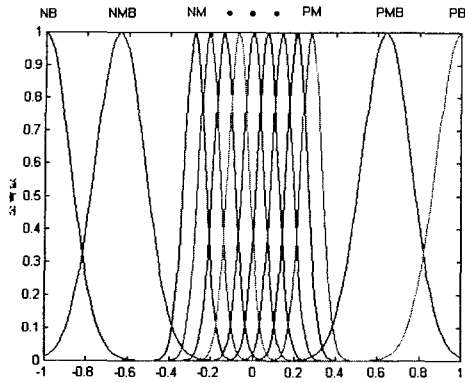


그림 8. 첫 번째 오버슈트 이후의 오차( $e_{fic}$ )와 오차변화율( $\Delta e_{fic}$ )의 소속함수.

Fig. 8. After arrive at the first overshoot, used to error( $e_{fic}$ ) and change rate of error membership function( $\Delta e_{fic}$ ).

### 3. 모의실험

본 논문에서 제안한 제어기의 응답특성을 살펴보기 위해 다음의 2차 플랜트 전달함수에 적용하여 모의 실험하였다.

$$G(s) = \frac{1.2033 \times 10^8}{s^2 + 2.9627s + 6.5775 \times 10^6} \quad (16)$$

그림 9는 TDNN 신경망 제어기와 퍼지보상기를 갖는 TDNN 신경망 제어기의 응답이며, 과도영역에서 큰 오버슈트를 발생시키는 것을 관찰할 수 있다.

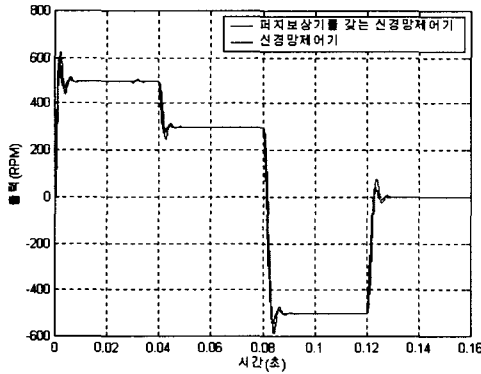


그림 9. TDNN 제어기와 퍼지보상기를 갖는 TDNN 제어기의 응답.

Fig. 9. The response of the TDNN controller and the TDNN controller within fuzzy compensator.

그림 10.은 30ms에서 7rpm 외란을 가했을 때 퍼지보상기를 갖는 TDNN 제어기(1)와 TDNN 제어기(2)의 플랜트 출력 응답을 비교하여 도시하였다. 그림에서 퍼지보상기를 갖는 TDNN 제어기(1)가 TDNN 제어기(2)보다 오버슈트가 작고 빠르게 정상상태로 회복되는 것을 관찰할 수 있다. 그리고 그림 11은 TDNN 신경망제어기의 학습회수에 따른 제곱오차(SSE)의 변화율을 나타낸 것이다. 설정치가 500rpm일 때, 두 제어기 모두 원하는 플랜트 출력에 도달하지만, TDNN 제어

기에 비해 제안한 리커런트 TDNN 제어기가 4.887% 낮은 최대오버슈트와 3.7ms 빠른 정정시간을 가진다.(그림 9, 그림 12, 그리고 표 3 참조) 그림 9와 그림 12는 신경망제어기의 출력을 퍼지보상기가 보상하여 500rpm에 필요한 플랜트 입력으로 적절히 조절되고 있음을 나타내고 있다.

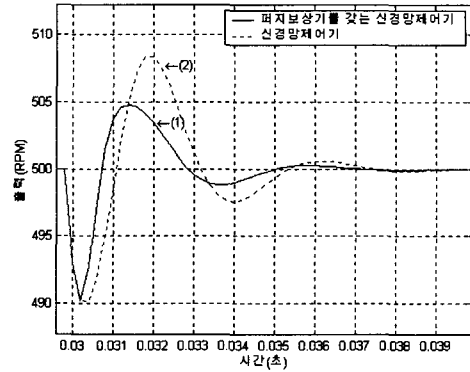


그림 10. 30ms에서 7rpm 외란을 가했을 경우의 퍼지보상기를 갖는 TDNN 제어기(1), TDNN 제어기(2)의 플랜트 출력 응답.  
Fig. 10. For a disturbance of 7rpm at 30ms, the response of the TDNN controller within fuzzy compensator(1) and the TDNN controller(2).

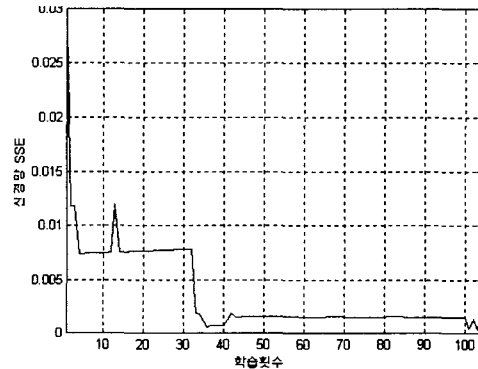


그림 11. TDNN SSE 그래프.  
Fig. 11. The graphics of TDNN SSE.(Sum of Squared Error)

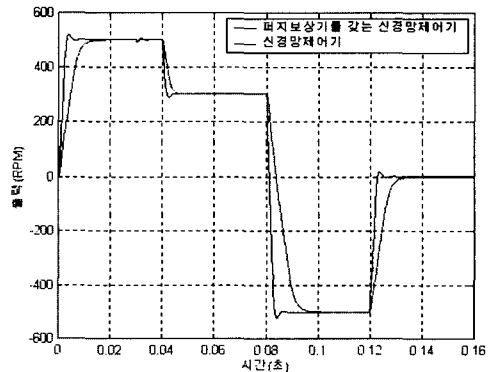


그림 12. 리커런트 TDNN 제어기와 퍼지보상기를 갖는 리커런트 TDNN 제어기의 응답.  
Fig. 12. The response of the Recurrent TDNN controller and the Recurrent TDNN controller within fuzzy compensator.

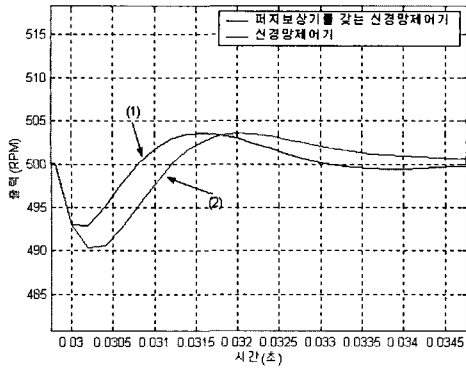


그림 13. 30ms에서 7rpm 외란을 가했을 경우의 퍼지보상기를 갖는 리커런트 TDNN 제어기(1), 리커런트 TDNN 제어기(2)의 플랜트 출력 응답.  
 Fig. 13. For a disturbance of 7rpm at 30ms, the response of the Recurrent TDNN controller within fuzzy compensator(1) and the Recurrent TDNN controller(2).

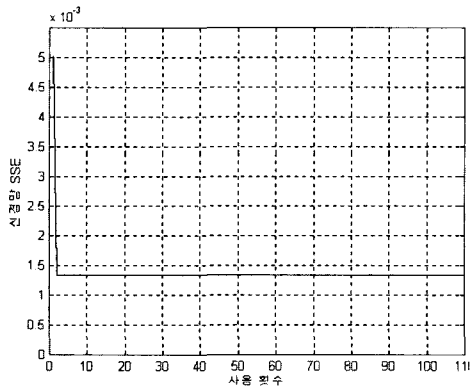


그림 14. 리커런트 TDNN SSE 그래프.  
 Fig. 14. The graphics of Recurrent TDNN SSE.(Sum of Squared Error)

표 3. 응답특성 비교.(500rpm)  
 Table 3. Comparison of a response characteristics.(500rpm)

	지연 시간	상승 시간	정정 시간	최대 오버슈트	외란 인가시 수렴시간
TDNN 제어기	1.7ms	1.81ms	6.7ms	8.94%	37ms
Recurrent TDNN 제어기	1.6ms	2.036ms	3ms	4.053%	2.8ms

그림 13.에서 30ms에서 7rpm 외란을 가했을 때, 리커런트 TDNN 제어기가 TDNN 제어기에 보다 낮은 오버슈트와 3.7ms 빠른 수렴시간을 가짐을 관찰할 수 있다. 리커런트 TDNN 제어기와 TDNN 제어기의 응답 특성을 표 3. 에 비교하였다. 그림 11.에서 나타낸 바와 같이 TDNN의 SSE는 초기엔 0.0285이지만 학습횟수가 110번일 땐 0.000376으로 감소되었다. 그리고 그림14.에서는 리커런트 TDNN의 SEE는 0.0055에서 시작해서 0.00133으로 수렴한다.

### 5. 실험 및 고찰

본 논문에서 제안한 제어기의 성능을 확인하기 위하여 비

선형 유압 서보시스템에 적용하여 그 성능을 확인한다.

그림 15는 DSP 프로세서로 유압 서보모터의 속도제어를 하기 위한 전체 실험장치의 구성도이다.

PC1은 ADSP-21020 계열 개발 소프트웨어를 사용하여 C 언어로 작성하고 제안한 알고리즘을 DSP 어셈블리 언어로 번역한 뒤 실행 파일을 생성시켜 ADSP-21020 EZ-LAB™ DSP 보드로 RS-232 통신 방식으로 다운로드(Down loading)하여 알고리즘을 수행한다.

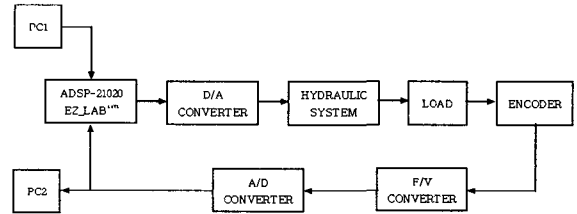


그림 15. 유압 서보 시스템 실험 구성도  
 Fig. 15. The block diagram of overall experimental system.

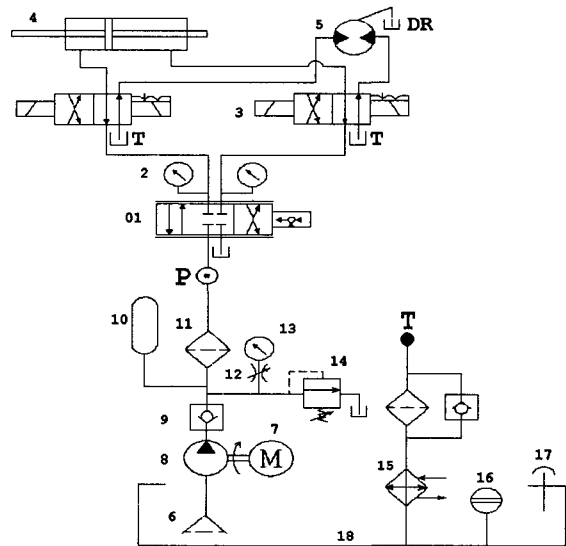


그림 16. 유압 서보시스템의 배치도  
 Fig. 16. The platform of external oil circuit.

- 01 : 직접구동 밸브(Direct Drive Valve)
- 10 : 축압기(Accumulator)
- 02 : 압력 센서
- 11 : 여과기(고압: 3 μm, 저압: 50 μm)
- 03 : 솔레노이드(2sets)
- 12 : 제한기(Restrictor)
- 04 : 유압 실린더
- 13 : 압력 센서
- 05 : 유압 모터
- 14 : Relier Valve
- 06 : 흡입관 여과기
- 15 : 냉각기
- 07 : 유압 펌프 구동용 3상 모터
- 16 : 오일 탱크 유량계
- 08 : 유압 펌프
- 17 : 여과기 고정쇄
- 09 : 역행 방지판(Check Valve)
- 18 : 오일 탱크

유압 서보시스템은 유압모터와 직접구동 밸브(Direct Drive valve), 관성부하로 이루어진다. 유압 서보시스템에서 제어기의 제어 입력 신호를 D/A 변환기를 사용하여 아날로

그 전압으로 바꾼 다음 직접구동 밸브에 인가한다. 직접구동 밸브는 서보 증폭기를 내장하고 있는 일체형으로 직접구동 밸브의 구동신호(제어입력)는  $\pm 10V$ 로 ADSP-21020 EZ-LAB™ DSP 보드에 D/A 변환기를 연결하여 생성하며 직접구동 밸브의 A, B 포트는 유압모터의 포트에 연결되고 유압모터의 축은 커플링을 통해 관성부하를 연결하는 축에 결합된다. 밸브에 의해 제어된 유량은 유압모터로 공급되고 공급된 유량에 비례하여 회전력을 발생시킨다. 관성부하를 고정하는 축은 다시 커플링을 통하여 증분 광부호기로 연결된다. 증분 광부호기는 유압모터의 회전수를 감지하는 부분으로 F/V 변환기를 사용하여 유압모터의 회전수를 전압신호로 변환하고, 변환된 신호는 A/D 변환기를 통과하여 ADSP-21020 EZ-LAB™ DSP 보드로 연결된다.

PC2는 A/D 변환 카드(Lab-PC-1200/AI)로 실험 결과를 기록한다.

부하는 프라이 휠(Fly wheel) 형태의 회전 관성부하  $48.836 \text{ kg/cm}^2$ 을 사용하였고 유압모터의 축에 스프링 커플러를 연결하여 사용하였다. 유압 서보시스템 실험장치는 서보밸브를 100% 열고 유압펌프의 압력을 최대로 하면 250[rpm]으로 모터가 회전하지만, 실험은 설정치가 설정치가  $\pm 200$ [rpm]일 때  $60[\text{kg/cm}^2]$ 의 유압을 유지하여 수행하였다. 이상의 실험 결과는 표 4. 에서 비교 검토하였다. 그림 16.은 실험에 사용한 유압 서보시스템의 배치도이다.

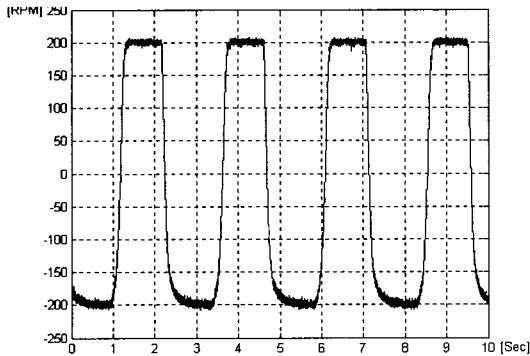


그림 17. 퍼지 보상을 갖는 TDNN 제어기의 실험결과.  
Fig. 17. The experimental result of the TDNN controller within fuzzy compensator.

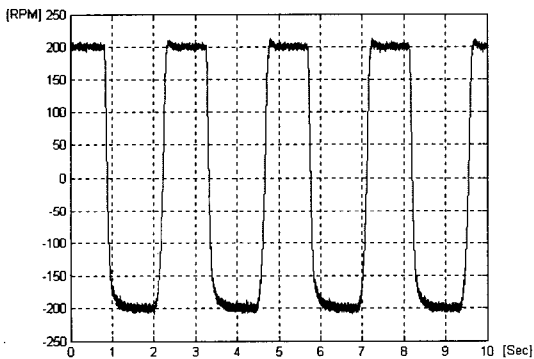


그림 18. 퍼지 보상을 갖는 리커런트 TDNN 제어기의 실험결과.  
Fig. 18. The experimental result of the Recurrent TDNN controller within fuzzy compensator.

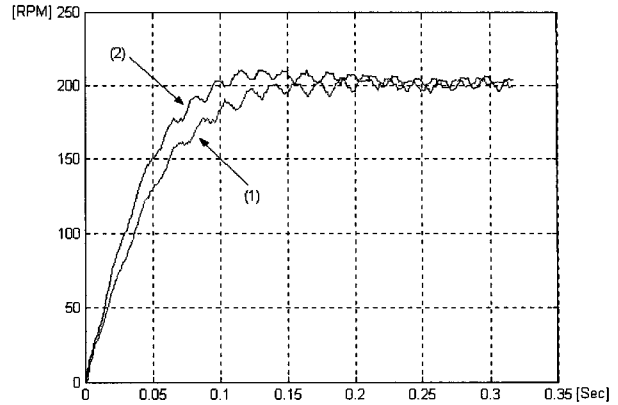


그림 19. 퍼지 보상을 갖는 TDNN 제어기(1)와 퍼지 보상을 갖는 리커런트 TDNN 제어기(2)의 실험결과.  
Fig. 19. The experimental result of the TDNN controller within fuzzy compensator(1) and the Recurrent TDNN controller within fuzzy compensator(2).

표 4. 제안한 제어기의 실험결과  
Table 4. The experimental result of proposed controller.

성능지수	지연시간	상승시간	정정시간	정상상태 오차
제어기				
TDNN 제어기	36.56 ms	92.65 ms	117 ms	Min. : 9.6889(RPM) Max. : 7.2444(RPM)
리커런트 TDNN 제어기	29.48 ms	69.28 ms	79 ms	Min. : 9.911(RPM) Max. : 10.4889(RPM)

그림 17은 퍼지 보상을 갖는 TDNN 제어기의 실험 결과이고, 그림 18은 제안한 RTDNN 제어기의 실험결과이다. 두가지 제어기의 성능을 비교하기 그림 19에 응답의 상승부분을 비교하여 도시하였다. 수치적인 비교는 표 4에 나타나 있다. 실험 결과를 비교하면 RTDNN 제어기가 TDNN 제어기 보다 우수함을 관찰할 수 있다.

## 6. 결 론

본 논문에서는 플랜트를 완전하게 수학적으로 모델링하지 않고, 플랜트의 입출력만을 이용하여 제어기를 구성하였고, 신경망제어기의 성능을 향상시키기 위해 퍼지보상을 이용하였다. 또 학습이 부족한 신경망제어기의 성능을 온라인으로 향상시키기 위해 플랜트의 역모델 신경망을 사용하였다.

플랜트의 역모델 신경망을 학습시킨 결과를 이용하여 주 신경망의 가중치를 변경시킴으로써 원하는 플랜트의 동적 특성을 얻도록 하였으며 2차 플랜트에 제안한 알고리즘을 적용하여 성능을 모의실험한 결과 TDNN 제어기에 비해 상승시간은 비록 길지만, 정상상태에 이르는 정정시간이 짧았고, 오버슈트가 정정시간 편차안에 들어간다. 최대오버슈트 역시 신경망제어기에 비해 낮았으며, 외란의 영향에 대해 제안한

제어기가 더욱 견실하였고 제어기의 학습횟수가 증가할수록, 즉 시간이 경과할수록 SSE가 감소하고 있음을 확인할 수 있었다.

본 논문에서 제안한 리커런트 TDNN 제어기가 TDNN 제어기보다 성능이 개선됨을 모의 실험을 통해 확인할 수 있었다. 그리고, 비선형 유압 서보모터의 실시간 제어를 위해 DSP 프로세서를 이용하여 제어기를 구현하였으며 본 논문에서 제안한 제어기의 성능을 실험을 통해 확인할 수 있었다.

앞으로의 연구과제는 신경망제어기의 신경망수를 자율적으로 조절할 수 있는 SONN(self-organization Neural network)를 도입하는것입니다.

### 참 고 문 헌

[1] 이상배, "퍼지-뉴로제어 시스템", (주)교학사, 1999  
 [2] Peter Vas, "Artificial-Intelligence-Based Electrical Machines and Drives", Oxford University Press, Inc., pp.77~234, 1999  
 [3] E.H.Mamdani, "Twenty years of fuzzy control : Experiences gained and lessons learnt", proc. 2nd IEEE Int. Conf. fuzzy systems, San Francisco, CA, pp339-344, 1993  
 [4] 김용태, "신경망과 퍼지보상기를 이용한 유압 서보모터의 속도제어기 설계", 석사학위논문, 2000  
 [5] Lefteri.H.Tsoukalas, Robert E.Uhrig, "Fuzzy and Neural Approaches in Engineering", John Wiley & Sons, Inc., pp.13~405, 1997  
 [6] Simon Haykin, "Neural Networks, A comprehensive foundation ,Second Edition", Prentice Hall International, Inc., pp.1~255, pp.664~789, 1999  
 [7] Michael A.Arbbib, "The Handbook of Brain Theory and Neural Networks", The MIT Press, pp.19~20, 1995  
 [8] 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  
 [9] Kevin M.Passino, Stephen Yurkovich, "Fuzzy Control", Addison Wesley Longman, Inc., 1998  
 [10] 채석, 오영석, "퍼지이론과 제어", 청문각, 1999.  
 [11] T. Takagi and M. Sugeno, "Derivation of fuzzy control rules from human operator's control actions", in Proc. of the IFAC Symp. on Fuzzy Information, Knowledge Representation and Decision Analysis, Marseille, France, 1983.  
 [12] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller", Int. J. Man-Machine Studies, Vol. 7, No. 1, 1975.  
 [13] Ronald R. Yager, Dimitar P. Filev, "Essentials of Fuzzy Modeling and Control", John Wiley & Sons, Inc., 1994.

### 저 자 소 개



#### 이 상 윤(Sang-Yun Lee)

1986년 금오공과대학교 전자공학과 졸업  
 1991년 금오공과대학교 대학원 전자공학과 (공학석사)  
 1998년 경남대학교 대학원 전자공학과 (공학박사)  
 1995년 3월~2001년 8월 경남대학교 시간강사

2000년~2001년 4월 (주) 피엘콤 부소장  
 2001년 6월~2001년 8월 (주) 한국부식연구소 부소장  
 2000년~현재 한국신호처리·시스템학회 평이사  
 2001년 9월~현재 경남대학교 기계자동화공학부 연구전임강사  
 2002년 5월~현재 (주)소라 인터내셔널 코리아 감사

연구분야 : 시스템지능제어 및 자동화, 마이크로프로세서응용



#### 신 위 재(Wee-jae Shin)

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