

퍼지보상기를 갖는 신경망제어기 설계

김용태* · 이상윤 · 신위재
경남대학교 전자공학과

Design on Neural Network Controller with a Fuzzy Compensator

Yong-Tae Kim*, Sang-Yun Lee, Wee-Jae Shin
Department of Electronic Engineering
Kyungnam University
email : cwee@kyungnam.ac.kr

요약

본 논문에서는 신경망제어기의 출력을 보상하는 퍼지보상기를 갖는 신경망제어기에 관하여 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 동적 특성을 계속해서 학습할 수 있도록 시간이 경과함에 따라 신경망제어기의 성능이 향상되도록 하였다. 이것을 확인하기 위해서, 2차 플랜트에 적용하여 제안한 제어기의 성능을 확인하였다.

I. 서론

산업현장에는 비선형시스템들이 존재하며 이를 제어하기 위한 방법으로 퍼지제어방식과 신경망제어방식이 사용된다. 퍼지제어방식은 소속함수와 제어규칙을 기반으로 하여 비선형성이 강하고 불확실하며 복잡한 시스템을 비교적 쉽게 제어할 수 있으나, 제어규칙을 작성하기가 힘들며, 전문가의 주관적인 관점에 따라 크게 달라질 수 있다. 신경망은 비선형시스템을 모델링하고 근사화시킬 수 있으며, 비선형 시그모이드 활성화함수를 사용하여 전통적 제어법으로 접근하지 못하는 난해한 비선형 제어문제를 해결하기 위한 비선형

사상능력을 제공해 준다. 하지만, 학습시간이 많이 소요되고, 학습에 필요한 플랜트의 입출력데이터를 구하기 어렵다.[1][2][3] 또한, 학습이 완료되어진 신경망을 사용하더라도 예상치 못한 외란으로 인해 플랜트의 출력이 좋지 못하다.

본 논문에서는 이와 같이 학습된 신경망제어기를 사용하더라도 플랜트의 출력이 좋지 못한 결과를 가져올 때, 이를 퍼지보상기를 사용하여 원하는 결과를 얻을 수 있도록 하는 신경망제어기를 설계하여 그 성능을 확인하였다.

II. 신경망제어기 설계

퍼지보상기를 갖는 신경망제어기는 주 제어기인 신경망제어기와 신경망제어기의 출력을 보상시켜주는 퍼지보상기, 플랜트의 역모델을 학습하는 역모델 신경망(Inverse Model Neural Network: IMNN)으로 구성되고, [그림 1]에 나타내었다.

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

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 $u_{flc}(n)$ 을 계산한다.

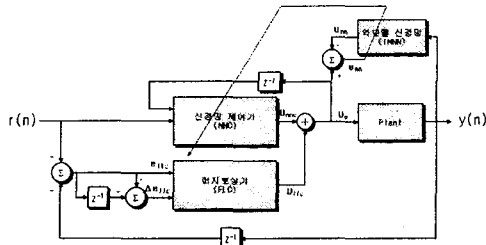


그림 1. 퍼지보상기를 갖는 신경망제어기 구조

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

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

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

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

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

2.1 신경망 구조와 학습법

신경망 제어기에는 시간지연 신경망(Time Delayed Neural Network: TDNN)을 사용하였으며, 역모델 신경망에는 NARX(nonlinear autoregressive with exogenous inputs) 신경망을 사용하였다.[1][4] 두 신경망 모두 시간지연을 가지기 때문에, 신경망 구조를 차분방정식 또는 순환방정식으로 기술할 수 있다.[3] 따라서, 시간에 따라 변하는 비선형 동적시스템을 모델링하기에 유리하다. 그리고, 플랜트 역모델을 On-line으로 학습시킬 수 있기 때문에, 학습이 부족한 신경망제어기의 성능을 On-line으로 보완시킬 수 있다.

학습알고리즘은 일반적으로 신경망을 학습시키는데 사용하는 오류 역전파 학습 알고리즘을 사용하였다. 지역 극소점에 빠지는 것을 방지하기 위하여 모멘텀법을 사용하였으며 가중치 조절식은 다음과 같다.[1][3]

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

$$\begin{aligned} \Delta W_{jk}(n+1) &= 2\eta e_{nn}(n)y(n)(1-y(n))h(n) \\ &\quad + \alpha\Delta W_{jk}(n) \end{aligned} \quad (5)$$

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

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

η : 학습률, α : 모멘텀 상수

$x(n)$: 입력층 뉴런값

$h(n)$: 은닉층 뉴런값

$y(n)$: 출력층 뉴런값

$W_{ij}(n)$: 입력층과 은닉층사이의 가중치

$W_{jk}(n)$: 은닉층과 출력층사이의 가중치

$$\Delta W = W_{new} - W_{old}$$

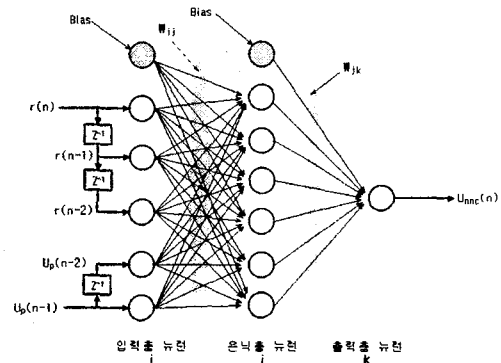


그림 2. 신경망제어기로 사용된 시간지연신경망(TDNN)

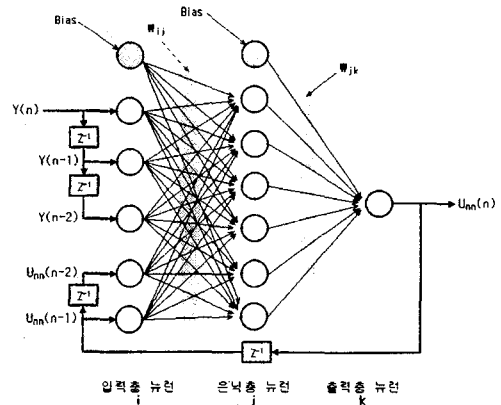


그림 3. 역모델 신경망에 사용된 NARX 신경망

플랜트의 역모델을 구하기 위해서 퍼지제어기를 사용하여 플랜트의 역모델을 학습하였다.

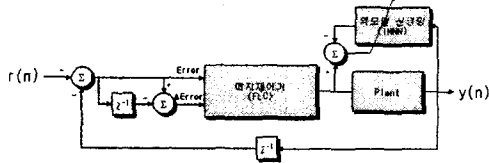


그림 4. 퍼지제어기를 사용한 플랜트 역모델 학습 구조

2.2 퍼지보상기[5]

첫 번째 오버슈트가 발생하기 전에는 제어규칙 1을 사용하여 오버슈트를 감소시킨다. 이때, 사용한 오차 e_{flc} 와 오차변화율 Δe_{flc} , 플랜트 입력 u_{flc} 의 소속함수는 [그림 5]와 같다.

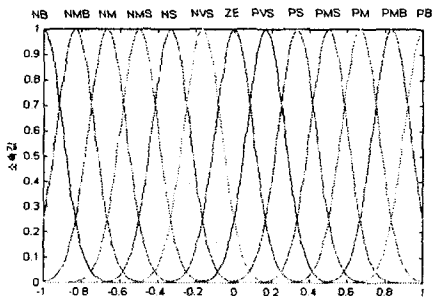


그림 5. 첫 번째 오버슈트에 도달하기 전에 사용된 오차(e_{flc})와 오차변화율(Δe_{flc}) 소속함수와 플랜트 입력(u_{flc}) 소속함수

첫 번째 오버슈트 발생 이후에는 제어규칙2를 사용하게 되고, 이때 오차 e_{flc} 와 오차변화율 Δe_{flc} 의 소속함수는 [그림 6]과 같으며 플랜트 입력 u_{flc} 의 소속함수는 [그림5]와 같다. 제어규칙2는

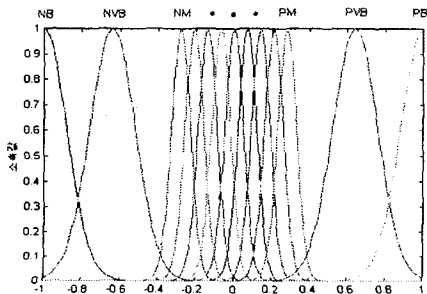


그림 6. 첫 번째 오버슈트 이후의 오차(e_{flc})와 오차변화율(Δe_{flc})의 소속함수

설정치에 더 빨리 도달할 수 있도록 해준다.

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

표 1. 제어규칙1

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	ZE	PM	PS	PMS	PM	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

표 2. 제어규칙2

IV. 모의 실험 및 고찰

모의실험에 사용한 2차 플랜트의 전달함수는 다음과 같다.

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

식(6)을 샘플링 시간 0.2ms로 이산화하여 차분방정식을 구한다.

$$y(k+1) = 1.39711y(k) - 0.538894y(k-1) + 2.59383u(k) \quad (7)$$

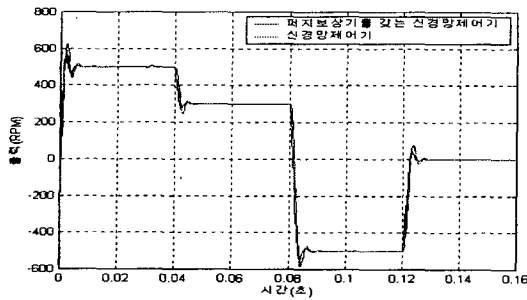


그림 7. 가변속일 때의 신경망제어기와 퍼지보상기를 가진 신경망제어기의 2차 플랜트 응답

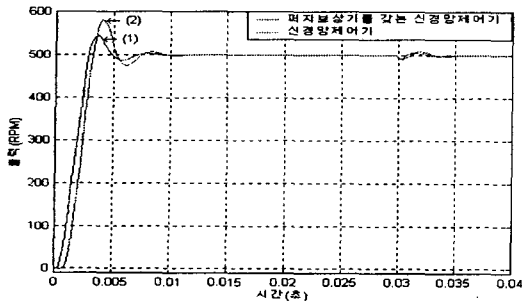


그림 8. 500rpm일 때의 퍼지보상기를 갖는 신경망제어기 (1)과 신경망제어기(2)의 2차 플랜트 응답

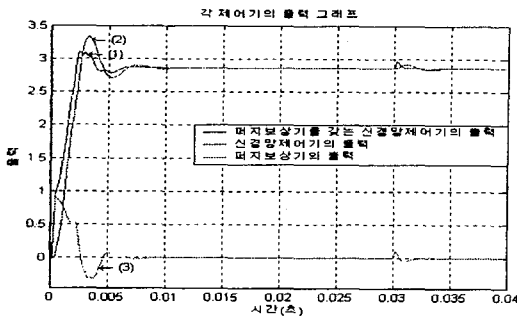


그림 9. 500rpm일 때의 퍼지보상기를 갖는 신경망제어기 (1), 신경망제어기(2), 퍼지보상기(3)의 출력 그래프

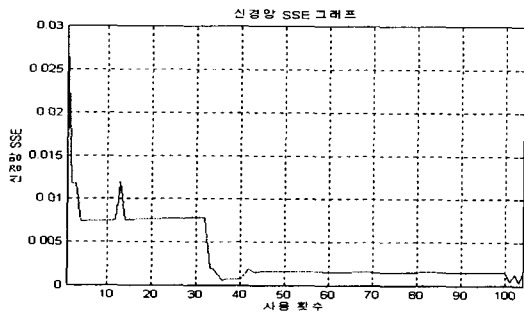


그림 10. 신경망 SSE(Sum of Squared Error) 그래프

	지연시간	상승시간	정정시간	최대오버슈트
신경망제어기	2.21ms	1.63ms	8.7ms	16.29%
퍼지보상기를 가지는 신경망제어기	1.7ms	1.81ms	6.7ms	8.94%

표 3. 성능지수를 통한 응답특성 비교 (500rpm)

[그림 7]은 가변속일 때의 2차 플랜트 응답을 나타내고 있으며, 두 제어기 모두 원하는 출력에 도달하지만 신경망제어기에 비해 제안한 제어기가 7.35% 낮은 최대오버슈트와 2ms 빠른 정정시간을 가진다.([그림 8]과 [표 3] 참조) [그림 9]에서 살펴보는 바와 같이 신경망제어기의 출력을 퍼지보상기가 보상하여 원하는 플랜트의 출력에 필요한 플랜트 입력으로 조절되고 있음을 알 수 있다. 그리고, [그림 10]에 나타난 바와 같이 신경망 SSE는 초기엔 0.0285이지만 사용횟수가 110번일 땐 0.000376으로 감소되었다.

V. 결론

2차 플랜트에 적용하여 퍼지보상기를 가진 신경망제어기의 성능을 모의실험한 결과 신경망제어기에 비해 지연시간은 비록 길지만, 정상상태에 이르는 정정시간과 상승시간은 짧았다. 그리고, 최대오버슈트 역시 신경망제어기에 비해 좋은 성능을 보이고 있다. 또한, 제어기의 사용횟수가 증가하면, 즉 시간이 경과할수록 SSE가 감소하고 있음을 알 수 있는데, 이것은 신경망제어기가 시간이 경과함에 따라 성능이 향상되고 있음을 보여준다.

VI. 참고문헌

- [1] Lefteri H. Tsoukalas, Robert E.Uhrg, "Fuzzy and Neural Approaches in Engineering", John Wiley & Sons, Inc. pp.238-250, p333-382, 1997
- [2] Peter Vas, "Artificial-Intelligence-based Electric Machines and Drives", Oxford University Press, pp.152 - 170, 1999
- [3] 이상배, "퍼지-뉴로제어 시스템", (주)교학사, pp.153-194, 1999
- [4] Simon Haykin, "Neural Networks", Prentice-Hall, Inc. pp.732-747, 1999
- [5] Ronald R.Yager, Dimitar P.Filev, "Essentials of Fuzzy Modeling and Control", John Wiley & Sons, Inc. pp.109-202, 1994