

뉴럴 네트워크를 이용한 유도 전동기의 속도 제어

이동빈* 유창완 홍대승 고재호 임화영
 광운 대학교 제어 계측 공학과

The Speed Control of an Induction Motor Based on Neural Networks

Lee Dong-Bin*, Ryu Chang-Wan, Hong Dae-Seung, Ko Jae-Ho, Yim Wha-Yeong
 Dept. of Control and Instrumentation Engineering KwangWoon Univ.

Abstract - This paper presents an feed-forward neural network design instead PI controller for the speed control of an Induction Motor. The design employs the training strategy with Neural Network Controller(NNC) and Neural Network Emulator(NNE). Emulator identifies the motor by simulating the input and output map. In order to update the weights of the Controller, Emulator supplies the error path to the output stage of the controller using backpropagation algorithm. and then Controller produces an adequate output to the system due to neural networks learning capability. Therefore it becomes adjustable to the system with changing characteristics caused by a load. The speed control based on neural networks for induction motor is implemented by a vector controlled induction motor. The simulation results demonstrate that actual motor speed with neural network system well follows the reference speed minimizing the error and is available to implement on the vector control theory.

1. 서 론

유도 전동기는 전력 전자의 발달 및 벡터 제어의 이론에 의하여 고속, 높은 정도를 요하는 분야까지 적용이 가능해졌다. 산업 현장에서 속도 제어기는 PI나 PID 제어를 많이 사용하고 있으며 제어 이득이 적절히 조정될 경우 좋은 운전 특성을 가질 수 있지만 최적의 파라미터를 구하기가 어렵고 시스템의 부하 등 특성이 변할 때 민감하지 못하는 단점이 있다. 따라서 학습 능력을 갖고 있어 적응성을 갖는 뉴럴 네트워크 제어를 설계하여 유도 전동기의 외부 환경이나 자체 시스템 특성의 변화에 스스로 적응할 수 있도록 하는 연구가 진행되고 있다.^[1,2,3] 뉴럴 네트워크를 이용한 대부분의 제어기 경우, 학습에 필요한 지도 입력력 값을 구하기가 어렵고 웨이트를 근사화하기 위해 플랜트의 복잡한 비선형 수식을 미분하기가 어렵다. 이 논문에서는 벡터 제어 유도 전동기의 속도 제어를 뉴럴 네트워크에 기반한 NNC와 NNE로 구성된 제어기를 설계하였고 NNE는 유도 전동기와 병렬로 연결하여 플랜트의 입출력 쌍을 모의함으로써 모터의 근사화하는 학습 전략을 갖고 있다.^[2,6] NNC의 웨이트를 변화하기에 필요한 오차항을 에뮬레이터를 통하여 역전파함으로써 플랜트를 제어할 수 있도록 하였다. 또한 유도 전동기에 부하를 인가함으로써 유도 전동기의 실제속도와 기준속도와 오차를 평가 함수를 이용하여 전체 오차의 합이 최소가 되도록 하는 역전파 학습 알고리즘을 이용하여 시스템의 특성이 변하는 환경에 스스로 적응할 수 있도록 실현하였다.

2. 뉴럴 네트워크를 이용한 속도 제어

일반적으로 속도 제어기는 실제 속도 $w_r(i)$ 가 기준 속도 $x(i)$ 를 잘 추종하도록 제어하는데 여기서는 학습을 통하여 속도 오차가 최소가 되도록 하는 피드 포워드 뉴럴 네트워크 제어를 설계하고 원하는 플랜트 출력이 나오도록 하는 최적의 입력값 $u(i)$ 를 생성하는 제어 시스템이다. 제안한 전방향 뉴럴 네트워크를 이용한 속도 제어 시스템에 대한 구조를 그림 1에 나타내었다.

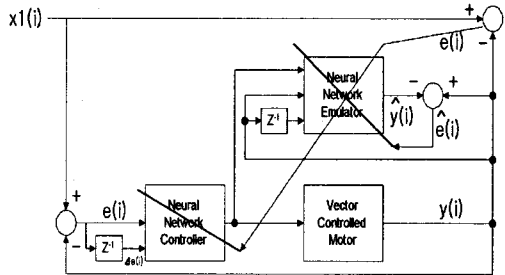


그림 1 뉴럴 네트워크를 이용한 유도 전동기 속도 제어 시스템

2.1 뉴럴 네트워크 시스템의 구성

네트워크는 학습시키고자 하는 입력 성분들과 그에 해당하는 출력 소자들 그리고 이를 연결시켜주는 회로망으로서 제어기 및 에뮬레이터로 구성되었고 플랜트는 벡터 제어하는 필드 제어기와 인버터 그리고 유도 전동기로 이루어진 유도 전동기이다. 제어기의 구조는 입력과 출력층, 은닉층의 3개 층으로 그림2와 같다. 입력뉴런은 3개로서 기준속도 x 와 기준속도와 실제 속도와의 오차인 e 그리고 그 오차의 변화량을 받고 있고 출력뉴런은 u 로서 유도 전동기로 하여금 원하는 속도를 내도록 명령을 내리는 토크 전류 I_{qs} 가 되는 MISO 네트워크이다. 에뮬레이터의 구조는 그림 3으로서 제어기와 동일한 구조로 이루어져 있다. 입력이 3개로서 모터의 입출력 쌍을 모의하여 근사화 하기 위한 u 와 실제 모터의 출력 y 를 피드백 받고 있다. 은닉층에는 각 5개의 노드(뉴런)로 구성된 형태이다. 뉴런의 개수는 성능에 적합하도록 경험을 통하여 선정하였다.

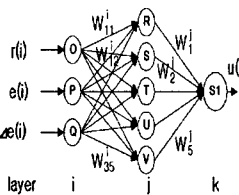


그림 2 신경망 제어기 (Neural Network Controller)

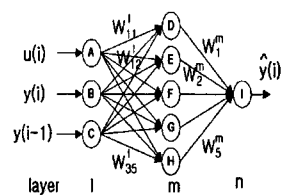


그림 3 신경망 에뮬레이터 (Neural Network Emulator)

2.2 뉴럴 네트워크 시스템의 학습 알고리즘

뉴럴 네트워크의 에블레이터는 모터 시스템과 병렬로 연결하고 모터의 입, 출력을 입력으로 받아 잘 모의하도록 함으로서 에블레이터를 플랜트로 간주하도록 한다. 실제 플랜트와 에블레이터간의 속도 오차의 전체 제곱의 합을 최소로 하는 역전과 학습 알고리즘을 써서 한 번에 원하는 방향으로 에블레이터의 가중치를 변경하도록 하였다. 동일한 방법으로 뉴럴 네트워크 제어기의 각 연결 요소의 가중치를 조정하나 유도 전동기 출력과 기준 속도간의 오차 e 를 에블레이터를 통하여 차례로 역전파시켜 제어기 출력단에서 오차항을 구하는 데에 사용하였다.

가중치 조절은 전방향 네트워크에서 보통 사용되는 역전과 학습 알고리즘을 이용하였고 입,출력이 P쌍인 목표 상태와 실제 상태간의 오차 식(1)의 전체 자승의 합과 관련한 평가 함수로 식(2)를 사용한다.

$$e(k) = (y'_k - \hat{y}_k) \quad (1)$$

$$E = \frac{1}{2} \sum_{k=1}^P e_k^2 \quad (2)$$

가중치의 평가 함수가 최소가 되는 조건으로 에블레이터 노드 i, j 간의 가중치 $W_{i,j}$ 가 식(3)의 관계가 되도록 구하였다.

$$\frac{\partial E}{\partial W_{i,j}} = 0 \quad (3)$$

최급 강하법을 써서 가중치 변분을

$$\Delta W_{i,j} = -\eta \frac{\partial E}{\partial W_{i,j}} \quad \eta > 0 \quad (4)$$

와 같이 계산하였다. 여기서 η 는 학습율이다. 지역 최저치(local minimum)에 빠지는 것을 벗어나기 위해 식(5)와 같이 모멘텀을 썼다.

$$\Delta W(k+1) = a\Delta W(k+1) + (1-a)\Delta W(k) \quad 0 \leq a \leq 1 \quad (5)$$

각 노드의 활성화함수는 ± 1 을 갖고 있는 tanh를 적용하였다. 따라서 일반화된 가중치 조절식은 따라서 아래와 같이 주어진다.

$$W(k+1) = W(k) + \Delta W(k+1) \quad (6)$$

2.2.1 학습 절차

본 논문의 속도 제어 시스템에서 뉴럴 네트워크 에블레이터의 학습은 유도 전동기의 실제 출력 y 와 에블레이터의 출력 \hat{y} 과의 전체 오차 \hat{E} 으로부터

$$\Delta W_m = -\frac{\partial \hat{E}}{\partial W_m} \quad (7)$$

여기서 우변은 일반화된 델타 규칙에 의해 구해지고 체인 룰로서 다음과 같이 이루어진다.

$$-\frac{\partial \hat{E}}{\partial W_m} = -\delta_n \frac{\partial I_n}{\partial W_m} \quad (8)$$

여기서 δ_n 은 n 번 째 layer, 즉 출력층의 오차항이다.

다음으로 에블레이터 은닉층에서의 오차항은 출력층의 오차를 이용하여 구할 수 있고 이 때 1층의 가중치 조절은 위와 동일한 방법으로 다음과 같이 구할 수 있다.

$$-\frac{\partial \hat{E}}{\partial W_i} = -\delta_m O_i \quad (9)$$

뉴럴 네트워크 제어기의 가중치를 변화하기 위해서는 먼저 에블레이터가 정확히 실제 모터를 근사화 해야 한다. 에블레이터가 모터로서 학습될 때에 제어기를 학습시키되 제어기가 한 번 학습될 때마다 모터의 입,출력 쌍이 변화므로 에블레이터가 모터와 동일하도록 학습을 시키는 것이 중요하다. 따라서 기준 속도 x 와 유도 전동기의 실제 출력 y 와의 전체 오차 E 로부터

$$\Delta W_m = -\frac{\partial \hat{E}}{\partial W_m} = -\frac{\partial \hat{E}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_n} \frac{\partial O_n}{\partial I_n} \frac{\partial I_n}{\partial W_m} \quad (10)$$

여기서 우변은 마찬가지로 델타 규칙과 체인 룰로서

다음과 같이 구한다.

$$-\frac{\partial E}{\partial W_m} = -\delta_n \frac{\partial I_m}{\partial W_m} \quad (11)$$

여기서 δ_n 은 n 번 째 layer, 즉 출력층의 오차항이다.

에블레이터 은닉층에서의 오차항은 출력층의 오차를 이용하여 구할 수 있고

$$-\frac{\partial E}{\partial W_i} = -\delta_m O_i \quad (12)$$

제어기의 가중치를 조정하기 위해서 먼저 에블레이터 입력층과 제어기의 출력층에서의 오차항을 구하면 에블레이터의 오차항이 역전파되므로 은닉층의 각 노드의 오차항의 합이 된다.

$$\delta_i = \sum \delta_m \quad (13)$$

따라서 제어기 출력층에서의 가중치 조절은

$$\Delta W_j = -\frac{\partial E_u}{\partial W_j} \quad (14)$$

여기서 우변은 마찬가지로 델타 규칙과 체인 룰로서 다음과 같이 구한다.

제어기 은닉층에서의 오차항은 출력층의 오차를 이용하여 구할 수 있다.

$$-\frac{\partial E_u}{\partial W_i} = -\delta_i \frac{\partial O_k}{\partial I_k} \frac{\partial I_k}{\partial O_k} \frac{\partial O_j}{\partial I_j} \frac{\partial I_j}{\partial W_i} \quad (15)$$

2.2 시뮬레이션 결과

원하는 속도를 내기 위한 토크분 전류 I_{qs} 가 뉴럴 네트워크로 구성된 제어기에서 나온 u 값이고 벡터 제어기인 펄드 기준 제어기에 들어간다. 벡터 제어기는 정상 상태에서 구동하며 일정 자속이 분포되는 형태이며 회전자 자속 기준으로서 d, q 전류로부터 슬립 속도를 계산한 것과 엔코더로부터 측정된 회전자 속도를 더하여 동기시켜 주는 간접 벡터 제어 방식이다. 모터에 대한 파라미터를 표1에 나타내었고 회전자 자속 및 유도 전동기의 해는 다음과 같다.

$$\lambda_{dre} = E' \frac{L_r}{\omega_r * L_m} \quad (16)$$

$$T_e = \frac{2}{P} J_m \frac{d}{dt} \omega_r + \frac{2}{P} B_m \omega_r + T_L \quad (17)$$

시뮬레이션 그림은 좌측은 무부하시를, 우측은 부하시로 구분하였다. 부하는 정격 토크에 해당하는 힘을 가해 주었다. 그림4는 기준 속도가 1000rpm이고 학습 되지 않은 임의의 상태로써 모터에 그림6처럼 임의의 u 값을 주었을 경우 나타나는 기준, 실제 그리고 에블레이터의 속도를 나타내었다. 그림5는 기준속도와 실제 속도간의 오차와 그 오차의 변화량을 나타내었다. 에블레이터를 20회 학습시 전체 오차 자승의 합의 변화량이 그림 7에 나타나 있다. 약 15에 정도 학습하면 최소점에 다다른다. 그림8, 10은 각각 50회 학습을 한 후 모터의 실제 속도와 에블레이터 속도를 보여 주고 있고 그 오차와 오차의 변화량을 그림11에서 보여준다. 이 때 모터에 가해지는 u 값을 그림9에서 보여준다. 그림12에서 보듯이 약 25회 정도면 오차 최소값에 도달한다. 부하를 가한 후에도 기준 속도를 잘 추종하고 있는 것을 보면 본 논문에서 뉴럴 네트워크를 적용한 속도 제어가 부하 등에 강건한 적응성을 가진 시스템임을 보여준다.

모터 정격	5hp	Rs(Ω)	0.2
정격 전압	220V	Rr(Ω)	0.18
극 수	4p	Ls(mH)	180
정격 전류	13.5	Lr(mH)	180
회 전 수	1748rpm	Lm(mH)	176
주 파 수	60Hz	J(kgm ²)	0.2

표1 유도 전동기 파라미터

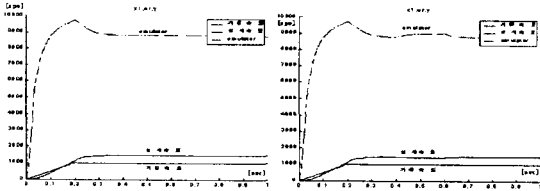


그림 4 초기 상태의 기준, 실제, 에뮬레이터 속도

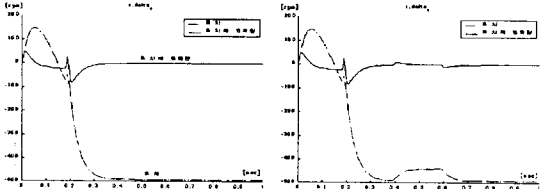


그림 5 기준속도와 실제속도간의 오차와 오차변화량

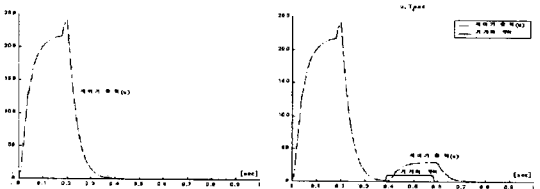


그림 6 제어기 출력 u (I_{qs} 전류)

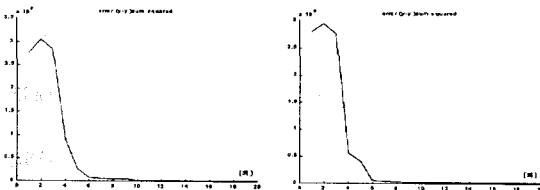


그림 8 에뮬레이터 학습시 전체 오차의 변화

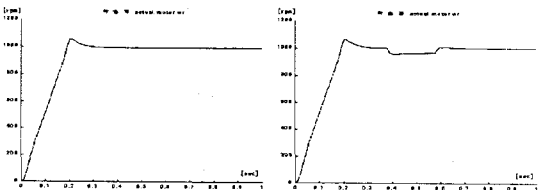
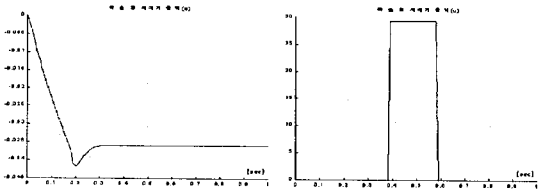


그림 9 학습을 한 후 실제 속도 (50회)



무부하 부하
그림 10 학습후 제어기 출력 $u_{\text{값}}$ (I_{qs} 전류)

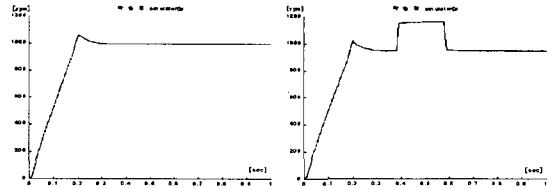


그림 11 학습 후 에뮬레이터

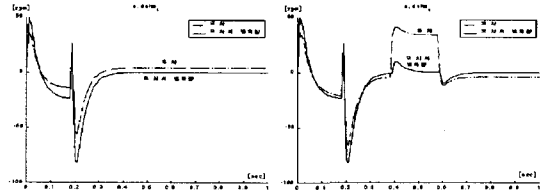
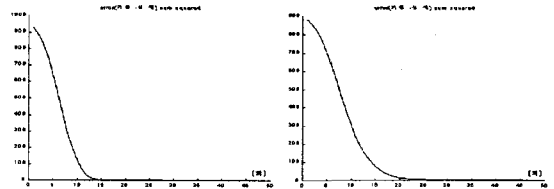


그림 12 학습시 오차와 오차의 변화량



무부하 부하
그림 13 학습시 전체 오차의 변화

3. 결 론

본 논문에서는 유도 전동기 시스템에 뉴럴 네트워크를 적용한 속도 제어가 부하와 같은 시스템의 특성이 변하는 상황에서도 기준 속도를 잘 추종하고 있음을 보여주고 있다. 유도 전동기는 상태 변수와 파라미터들이 서로 연관되어 있는 다변수 비선형 시스템으로서 속도 제어가 어렵지만 본 논문에서는 복잡한 시스템의 모델링이나 수식적 해석을 줄이면서 또한 플랜트의 입, 출력 데이터를 이용한 에뮬레이터 및 제어기를 적용한 뉴럴 네트워크 속도 제어 시스템은 학습을 많이 하지 않아도 빨리 최소점에 이르고 부하 등에 적응성을 갖아 시스템에 잘 적용될 수 있음을 보여준다. 차후에는 실제 시스템에 적용하고 온라인으로 할 수 있는 방법을 고찰하도록 한다.

[참 고 문 헌]

- [1] 김세찬, 원충연, "신경회로망을 이용한 유도전동기의 속도 제어", KIEE, Vol.45, No.1, p42~p53, 1996
- [2] T.Aoyama, S.Omatu, "Design of a Self-Tuning PID Control System by Neural Networks", 전학론C, Vol.116 No.11, p1197~1201, 평성 8년, 1998
- [3] S.Omatu, Y.Kishida, M.Yoshioka, "Neuro-Control for Single-Input Multi-Output systems, International conference on knowledge-based Intelligent Electronic Systems, Vol.1, p202~205, 1998
- [4] L.Tsoukalas, R.Uhrig, "Fuzzy and Neural Approaches in Engineering", John Wiley & Sons, 1997
- [5] A.Cichocki, R.Unbehauen, "Neural Networks for Optimization and Signal Processing, John Wiley, 1993
- [6] K.J.Hunt, D.Sbarbaro, "Neural Networks for Control System-A Survey", Automatica, Vol.28, No.6, pp1083~1112, 1992
- [7] D.W.Novotny, T.A.Lipo, "Vector Control and Dynamics of AC Drives", Clarendon Press Oxford, 1996
- [8] C.M.Ong, "Dynamic simulation of Electric Machinery", Prentice Hall PTR, 1998