

The Speed Control of Induction Motor Using Neural Networks

金世燦·元忠淵**
(Sei-Chan Kim · Chung-Yeon Won)

Abstract - This paper presents a speed control system of vector controlled induction motor using neural networks. The main feature of proposed speed control system is a Neural Network Controller(NNC) which supplies torque current to induction motor and Neural Network Emulator(NNE) which captures the forward dynamics of induction motor. A back propagation training algorithm is employed to train the NNE and NNC. In order to determine the NNC output error, plant(induction motor) output error can be back propagated through the NNE. The NNC and NNE for speed control of vector controlled induction motor is carried out by TMS320C30 DSP and IGBT current regulated PWM inverter. Through computer simulation and experimental results, it is verified that proposed speed control system is robust to the load variation.

Key words : Neural Network Controller, Neural Network Emulator, Back propagation training algorithm.

1. 서 론

유도전동기는 오랫동안 산업용으로 많이 사용되어져 왔지만 비선형이며 고도의 상호결합된 다변수 제어구조로 인하여 제어에 어려움이 있다. 이러한 제어의 어려움 때문에 주로 정속 운전용으로 사용되어져 왔고, NC선반, 각종 공작기계, 로봇 등과 같이 정밀하고 빠른 응답특성이 요구되는 서어보 운전용 전동기로는 직류전동기가 주로 쓰여져 왔다. 그러나 유도전동기는 견고하고 보수 유지가 용이하며 출력에 비하여 소형 경량이라는 장점이 있다. 또한 고속운전이 가능하며 관성이 작아 적절한 제어 방법이 적용 된다면 서어보 운전용으로 직류 전동기보다 유리한 특성을 갖는다[1]. 또한 마이크로프로세서의 비약적인 발달은 실시간에 많은 계산을 필요로 하는 벡터제어기법과 현대제어이론의 적용을 가능하게 하여 여러 분야에서 서어보 액추에이터로서 유도전동기의 사용이 점차 확대되고 있다[2].

최근까지 서어보 운전을 위한 제어 방법으로 산업현장에서 널리 쓰이고 있는 제어기는 비례적분제어기(PI), 또는 비례적분미분(PID) 제어기이다. PI혹은 PID제어기는 제어상수값이 적절히 조절될 경우 좋은 운전 특성을 얻을 수 있으며 제어 알고리즘이 간단하여 손쉽게 구현할 수 있고 아날로그 제어기로도 구성이 가능하다. 그러나 제어성능이 시스템의 파라미터 변동에 민감하며 부하외란에 대해 빠른 과도응답을 갖지 못하여 견실성이 떨어진다. 이러한 문제점을 해결하기 위하여 최근 학습기능을 가지고 있어서 외부환경이나 시스템 자체에 관한 지식이 없어도 제어환경의 변화에 스스로 적응할 수 있는 신경회로망을 제어분야에 응용하기 위한 연구가 활발히 진행 되고 있다

[3][4]. 신경회로망을 제어기에 응용할 경우 대부분 학습 알고리즘으로써 지도학습법을 사용하게 되는데 이 경우 신경회로망을 학습시키기 위해서는 필수적으로 지도 입력력 패턴이 필요하게 된다. 그러나 많은 경우에 있어서는 지도입력에 대한 지도 출력값을 구할 수 없다. 따라서 이러한 지도 출력력 패턴을 구하는 문제는 신경회로망을 제어기로 사용하는 경우에 있어서는 매우 중요한 문제이며 최근 이러한 문제를 해결하기 위한 연구가 진행되고 있다[5][6].

본 논문에서는 제어기 출력단에서의 오차항을 구하기 위하여 유도전동기의 입력력 관계를 모의할 수 있는 신경회로망 에뮬레이터를 구성하였다. 유도전동기의 실제 속도와 기준 속도와 오차는 이 신경회로망 에뮬레이터를 통하여 역전파 된다. 또한 빠른 명령주기를 가지는 DSP보드(TMS320C30)를 이용하여 매 샘플링 마다 오차를 역전파 시킴으로써 온라인 실시간 학습이 가능하게 하였다.

2. 신경회로망을 이용한 유도전동기 속도제어 시스템

본 논문에서 사용된 신경회로망 제어기의 목적은 유도전동기의 실제속도 $y(k)$ 가 기준속도 $r(k)$ 를 잘 추종할 수 있도록, 학습을 통하여 속도오차 $e(k) = (r(k) - y(k))$ 가 최소가 되도록 하는 제어값 $u(k)$ 을 생성하는 것이다. 그러나 신경회로망 제어기를 학습시키기 위해서는 제어기 출력단에서의 오차평가함수

$$ue(k) = \frac{1}{2}(u_r(k) - u(k))^2 \quad (1)$$

;여기서 $u_r(k)$ 는 지도출력값, $u(k)$ 는 실제 출력값

를 구하여야 한다. 하지만 신경회로망 제어기를 이용하여 유도전동기 속도 제어기를 구성하였을 경우 제어기 출력단에서의 오차항 $ue(k)$ 는 지도출력 $u_r(k)$ 가 존재하지 않으므로 구할 수 없다. 본 논문에서는 신경회로망 제어기의 지도출력 $u_r(k)$ 를 구하지 않고, 신경회로망 에뮬레이터를 통하여 유도전동기 출력

* 正 會 員 : 成均館大 工大 電氣工學科 時間講師

** 正 會 員 : 成均館大 工大 電氣工學科 副教授 · 工博

接受日字 : 1995年 6月 29日

1次 修正 : 1995年 8月 9日

2次 修正 : 1995年 11月 6日

3次 修正 : 1995年 12月 20日

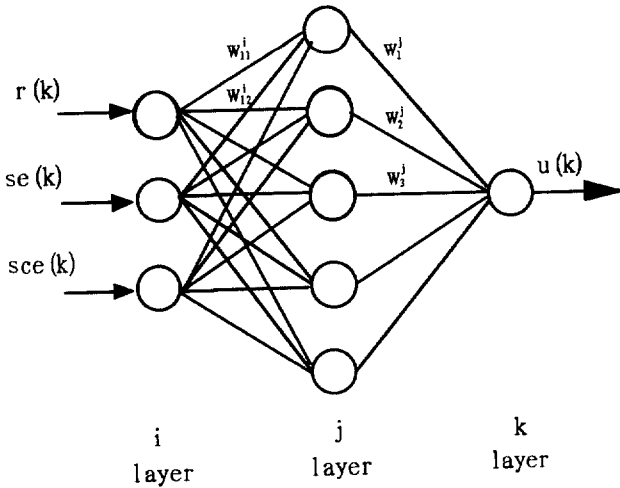


그림 1 신경회로망 제어기
Fig. 1 Neural network controller

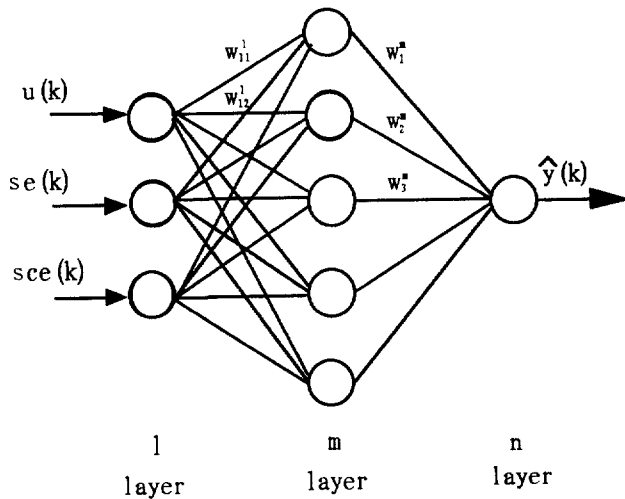


그림 2 신경회로망 에뮬레이터
Fig. 2 Neural network emulator

단에서의 오차항을 역전과 함으로써 신경회로망 제어기의 출력단에서의 오차항을 구하였다. 신경회로망 에뮬레이터의 역할은 유도전동기의 forward dynamics를 학습하고 유도전동기 출력단에서의 오차항을 역전파시켜 제어기 출력단에서의 오차항을 계산할 수 있는 오차 역전과 경로를 제공하는 것이다.

2.1 신경회로망 제어기와 에뮬레이터의 구성

구성된 신경회로망 제어기와 에뮬레이터는 그림 1과 그림 2에서 각각 보여주고 있다. 여기서 신경회로망 에뮬레이터와 신경회로망 제어기의 은닉층 뉴런의 갯수는 모두 최소의 갯수를 가지면서 가장 우수한 성능을 얻을 수 있도록 경험적으로 얻어졌다. 그림 1에서 신경회로망 제어기는 3개의 입력층 뉴런, 5개의 은닉층 뉴런, 1개의 출력층 뉴런을 가지는 3계층의 다층퍼셉트론으로 구성된다. 신경회로망 제어기의 입력으로는 속도기준값 $r(k)$, 기준속도와 실제속도의 오차

$$se(k) = r(k) - y(k) \tag{2}$$

그리고 오차의 변화율

$$sce(k) = se(k) - se(k-1) \tag{3}$$

을 선택하였다.

한편 신경회로망 에뮬레이터는 그림 2와 같이 3개 입력층 뉴런, 5개의 은닉층 뉴런, 한개의 출력층 뉴런을 가지는 3층의 다층 퍼셉트론으로 구성된다. 신경회로망 에뮬레이터의 입력으로는 신경회로망 제어기의 출력값 $u(k)$, 그리고 신경회로망 제어기의 입력으로 사용된 기준속도와 실제속도의 오차 $se(k)$, 그리고 오차의 변화율 $sce(k)$ 을 선택하였다. 이렇게 하면 신경회로망의 구조가 간단하게 되어 연산시간이 짧아지며, 학습방법으로 일반적인 오차역전파 알고리즘의 이용이 가능해진다. 따라서 하드웨어를 구성하여 제어를 할 경우 샘플링 타임을 줄일 수 있다. 또한 제어기와 에뮬레이터의 입력층의 입력범위가 ± 1 의 값을 가지므로 활성화함수로서 양방향 시그모이드 함수인 $\tanh(\cdot)$ 를 사용하였다.

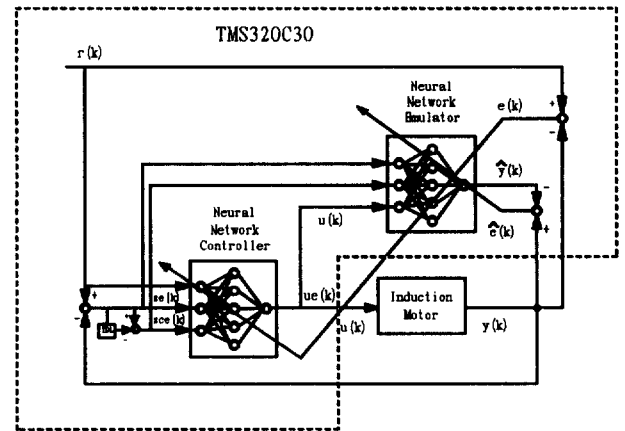


그림 3 신경회로망을 이용한 유도전동기 속도제어시스템
Fig. 3 Speed control system of induction motor using neural network

구성된 전체 제어 시스템은 그림 3에서 보여준다.

여기서 TDL은 tapped delayed line을 표시하며 점선부분은 하드웨어 구성시 DSP에서 소프트웨어적으로 처리되는 부분이다.

2.2 신경회로망 제어기와 에뮬레이터의 학습

신경회로망 제어기와 에뮬레이터의 학습과정은 다음과 같다. 먼저 신경회로망 에뮬레이터를 학습시키기 위하여 에뮬레이터의 출력과 유도전동기의 실제 출력과의 오차함수를 구하면,

$$\hat{e}(k) = \frac{1}{2}(y(k) - \hat{y}(k))^2 \tag{4}$$

; 여기서 $y(k)$ 은 유도전동기 실제 속도, $\hat{y}(k)$ 는 신경회로망 에뮬레이터 출력이다.

학습은 오차식 (4)를 최소화 하도록 다음과 같이 일반화된 델타규칙에 의해 이루어진다[7].

$$\Delta W_i^m(k) \propto -\frac{\partial \hat{e}(k)}{\partial W_i^m} \tag{5}$$

(5)의 우변은 체인 규칙에 의해 다음과 같이 나뉘어질 수 있다.

$$-\frac{\partial \hat{e}(k)}{\partial W_j^m} = -\frac{\partial \hat{e}(k)}{\partial net^n} \frac{\partial net^n}{\partial W_j^m} \quad (6)$$

여기서

$$\frac{\partial net^n}{\partial W_j^m} = \frac{\partial}{\partial W_j^m} \sum_j OUT_j^m W_j^m = OUT_j^m \quad (7)$$

가 되고, 다음을 정의하면

$$\delta^n = -\frac{\partial \hat{e}(k)}{\partial net^n} \quad (8)$$

식(6)은 음과 같이 표시된다.

$$-\frac{\partial \hat{e}(k)}{\partial W_j^m} = \delta^n OUT_j^m \quad (9)$$

그러므로 $\hat{e}(k)$ 에 대한 기울기가 감소 되기 위해서는 가중치 조절은 다음과 같이 이루어져야 한다.

$$\Delta W_j^m = \eta \delta^n OUT_j^m \quad (10)$$

; $\eta (0 < \eta < 1)$ 는 학습율

여기서 δ^n 는 n번째층의 뉴런의 오차항이며 다음과 같이 구해진다.

$$\delta^n = -\frac{\partial \hat{e}(k)}{\partial net^n} = -\frac{\partial \hat{e}(k)}{\partial OUT^n} \frac{\partial OUT^n}{\partial net^n} \quad (11)$$

윗식에서 우변의 두번째항은

$$\frac{\partial OUT^n}{\partial net^n} = f'(net^n) \quad (12)$$

$$OUT^n = f(net^n) = \tanh(net^n) \quad (13)$$

이 된다. 한편 (11)식 우변의 첫번째 항은 출력층의 경우 오차함수 $\hat{e}(k)$ 에 대한 정의식 (4)를 이용하면 다음과 같이 구할 수 있다.

$$\frac{\partial \hat{e}(k)}{\partial OUT^n} = -(y(k) - \hat{y}(k)) \quad (14)$$

그러므로 출력층의 뉴런에서의 오차항은 다음과 같이 표현된다.

$$\begin{aligned} \delta^n &= f'(net^n)(y(k) - \hat{y}(k)) \\ &= (1 - (\hat{y}(k))^2)(y(k) - \hat{y}(k)) \end{aligned} \quad (15)$$

은닉층의 경우에는 지도출력값을 알 수 없으므로 이미 알고

있는 출력층에서의 오차항을 이용하면 다음과 같이 구할 수 있다.

$$\begin{aligned} \frac{\partial \hat{e}(k)}{\partial OUT_j^m} &= \frac{\partial \hat{e}(k)}{\partial net^n} \frac{\partial net^n}{\partial OUT_j^m} \\ &= \frac{\partial \hat{e}(k)}{\partial net^n} \frac{\partial}{\partial OUT_j^m} \sum_j OUT_j^m W_j^m \\ &= \frac{\partial \hat{e}(k)}{\partial net^n} W_j^m = \delta^n W_j^m \end{aligned} \quad (16)$$

위의 결과를 식(11)에 대입하면 은닉층의 경우 오차항은 다음과 같이 표시된다.

$$\begin{aligned} \delta_j^m &= f'(net_j^m) \delta^n W_j^m \\ &= (1 - (OUT_j^m)^2) \delta^n W_j^m \end{aligned} \quad (17)$$

가 된다.

일반적으로 보다 빠른 학습속도를 위해 식(10)에 모멘텀항이 추가 되는데 이 경우 가중치 조절은 다음과 같이 표현된다.

$$\Delta W_j^m(k+1) = \eta \delta^n OUT_j^m + \alpha \Delta W_j^m(k) \quad (18)$$

$$W_j^m(k+1) = W_j^m(k) + \Delta W_j^m(k+1) \quad (19)$$

; 여기서 n은 학습반복횟수, $\alpha (0 < \alpha < 1)$ 는 관성항이다.

여기서 구성된 신경회로망 에뮬레이터가 플랜트 출력단에서의 오차항을 정확히 역전파 시킬 수 있기 위해서는 에뮬레이터가 플랜트의 입출력을 잘 모의할 수 있도록 하는 예비학습단계가 필요하다. 이를 위하여 본 연구에서는 한 주기의 기준입력 기간동안 임의의 제어입력패턴(u(k))을 1회 인가하여 예비학습을 실시하였다. 신경회로망 에뮬레이터가 학습된 후 제어기 출력단에서의 오차항을 구하기 위하여 플랜트 출력단에서의 오차항 $e(k)$ 를 구하면

$$e(k) = \frac{1}{2} (r(k) - y(k))^2 \quad (20)$$

; 여기서 $y(k)$ 는 유도전동기 실제속도, $r(k)$ 는 기준속도.

이 오차항은 오차 역전파 알고리즘에 의해 신경회로망 에뮬레이터를 통하여 역전파 된다.

은닉층에서의 오차항은

$$\begin{aligned} \delta_j^m &= f'(net_j^m) \delta^n W_j^m \\ &= (1 - (OUT_j^m)^2) \delta^n W_j^m \end{aligned} \quad (21)$$

$$\begin{aligned} \delta^n &= f'(net^n)(r(k) - y(k)) \\ &= (1 - (y(k))^2)(r(k) - y(k)) \end{aligned} \quad (22)$$

이 되고 신경회로망 에뮬레이터 입력층에서 신경회로망 제어기의 출력 $u(k)$ 와 연결된 노드에서의 오차항은 은닉층의 각 노드에서 역전파 되어온 오차항의 합이 된다.

$$\delta_j^i = \sum_j \delta_j^m \quad (23)$$

여기서 구해진 오차항 δ_j^i 은 다시 신경회로망 제어기를 학습시키기 위한 오차항으로 사용된다.

신경회로망 제어기 출력층의 가중치 변화량은

$$\Delta W_j^i \propto -\frac{\partial ue(k)}{\partial W_j^i} \quad (24)$$

이 되고, 여기서 식(24)는 체인룰(chain rule)에 의해

$$-\frac{\partial ue(k)}{\partial W_j^i} = -\frac{\partial ue(k)}{\partial net^k} \frac{\partial net^k}{\partial W_j^i} \quad (25)$$

이 되고 여기서

$$\frac{\partial net^k}{\partial W_j^i} = \frac{\partial}{\partial W_j^i} \sum_j OUT_j^i; W_j^i = OUT_j^i \quad (26)$$

이 된다.

다음을 정의하면

$$\delta^k = -\frac{\partial ue(k)}{\partial net^k} = -\frac{\partial ue(k)}{\partial u(k)} \frac{\partial u(k)}{\partial net^k} \quad (27)$$

여기서

$$\frac{\partial u(k)}{\partial net^k} = f'(net^k) \quad (28)$$

으로 계산된다.

그러나 $ue(k)$ 를 알지 못하므로 $\frac{\partial ue(k)}{\partial u(k)}$ 는 계산이 불가능하다. 따라서 에블레이터를 통하여 역전파된 오차값을 이용하여

$$\frac{\partial ue(k)}{\partial u(k)} = \delta^{k+1} \quad (29)$$

으로 두면

$$\delta^k = f'(net^k) \delta^{k+1} \quad (30)$$

$$\delta^{k+1} = \delta_j^i = \sum_j \delta_j^m \quad (31)$$

이 되며 이것을 사용하면 제어기 출력단에서의 가중치 변화량은

$$\begin{aligned} \Delta W_j^i &= -\frac{\partial ue(k)}{\partial W_j^i} = -\frac{\partial ue(k)}{\partial net^k} \frac{\partial net^k}{\partial W_j^i} \\ &= -\frac{\partial ue(k)}{\partial u(k)} \frac{\partial u(k)}{\partial net^k} \frac{\partial net^k}{\partial W_j^i} \\ &= -\delta^{k+1} f'(net^k) OUT_j^i \end{aligned} \quad (32)$$

으로 구하여 진다. 그리고 은닉층에서의 오차항 역시 유사하게 구할 수 있다.

3. 시뮬레이션 결과

그림 4는 시뮬레이션을 위하여 구성된 유도전동기 속도제어 시스템의 구조를 나타낸다.

여기서 $u(k)$ 는 신경회로망 제어기의 출력인 토오크분 전류 지령치 (i_{qs})이며 벡터제어기의 입력으로 이용되며 벡터제어기에서 2/3상변환을 거쳐 3상 기준 전류 i_a^*, i_b^*, i_c^* 를 출력한다. 신경회로망 제어기의 입력은 기준 속도값 $\omega_{ref}(k)$, 그리고 기준 속도와 실제속도 $\omega_r(k)$ 의 오차 $se(k)$, 그리고 오차의 변화율 $sce(k)$ 이 된다.

한편 신경회로망 에블레이터의 입력으로는 신경회로망 제어기의 출력값 $u(k)$, 그리고 신경회로망 제어기의 입력으로 사용된 기준속도 $\omega_{ref}(k)$ 와 실제속도 $\omega_r(k)$ 의 오차 $se(k)$, 그리고 오차의 변화율 $sce(k)$ 이 된다.

그리고 유도전동기는 식 (33)~(37)과 같은 5차 비선형 연립 미분 방정식으로 표현되며, 시뮬레이션을 위하여 Runge-Kutta 방법을 이용하여 해를 구하였다. 또한 전류변환장치로서 3상 전류제어형 인버터를 사용하였고 전류제어기로는 히스테리시스 전류제어기를 사용하였다.

$$\frac{d}{dt} i_{qs} = \frac{-1}{L_s L_r - L_m^2} (R_s L_r i_{qs} + w_r L_m i_{ds} - R_r L_m i_{qr} + w_r L_r L_m i_{dr} - L_r v_{ds}) \quad (33)$$

$$\frac{d}{dt} i_{ds} = \frac{-1}{L_s L_r - L_m^2} (-w_r L_m^2 i_{qs} + R_s L_r L_m i_{ds} - w_r L_r L_m i_{qr} + R_r L_m i_{dr} - L_r v_{ds}) \quad (34)$$

$$\frac{d}{dt} i_{qr} = \frac{-1}{L_s L_r - L_m^2} (R_s L_m i_{qs} - w_r L_s L_m i_{ds} + R_r L_s i_{qr} - w_r L_r L_s i_{ds} - L_r v_{qs}) \quad (35)$$

$$\frac{d}{dt} i_{dr} = \frac{-1}{L_s L_r - L_m^2} (w_r L_s L_m i_{qs} - R_s L_m i_{ds} + w_r L_r L_s i_{qr} + R_r L_s i_{dr} - L_m v_{ds}) \quad (36)$$

$$\frac{d}{dt} w_r = \left(\frac{p}{2}\right)^2 \frac{3}{2} \frac{L_m}{J} (i_{qs} i_{dr} - i_{ds} i_{qr}) - \frac{B}{J} w_r - \frac{p}{2} \frac{1}{J} T_l \quad (37)$$

그림 5는 시뮬레이션 순서도이다. 먼저 에블레이터의 예비학습단계가 완료된 후, 신경회로망 제어기는 주어진 기준 속도값을 추종할 수 있도록 매 샘플링(2ms) 순간마다 학습에 의하여

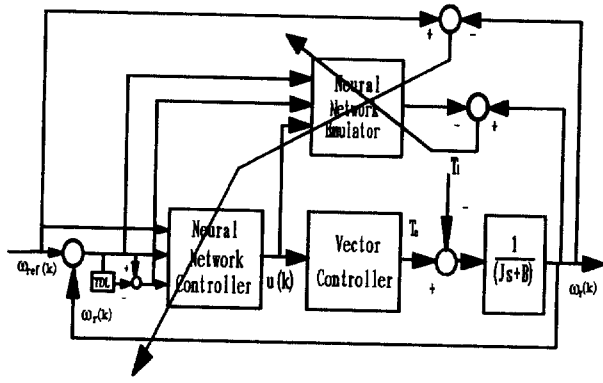


그림 4 유도전동기 속도제어 시스템
Fig. 4 Induction motor speed control system

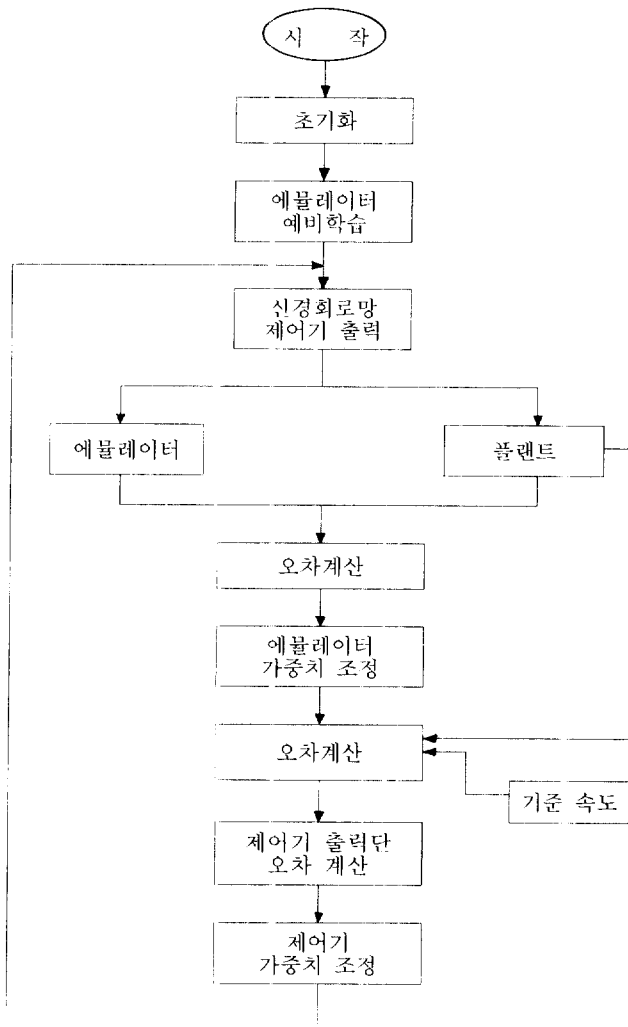


그림 5 시뮬레이션 순서도
Fig. 5 Flow chart for simulation

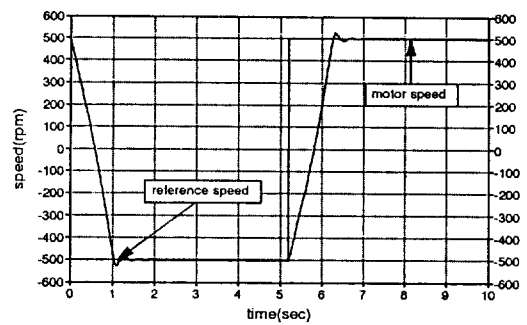
가중치를 변화 시키게 된다. 그와 동시에 신경회로망 에블레이터 역시 매 샘플링 순간마다 플랜트의 입출력관계를 모의 할 수 있도록 학습 된다. 신경회로망 에블레이터와 신경회로망 제어기의 은닉층 갯수는 모두 최소의 갯수를 가지면서 가장 우수

한 성능을 가지도록 경험적으로 얻어졌고 학습을 또한 시행착오법에 의하여 구하였다.

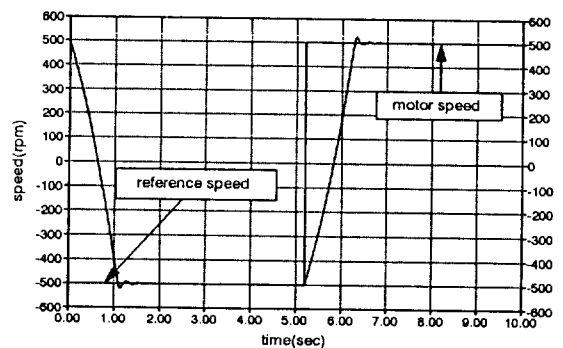
시뮬레이션에서 사용된 신경회로망 제어기와 신경회로망 에블레이터는 각각 입력층 뉴런 3개, 은닉층 뉴런 5개, 출력층 뉴런 1개씩으로 구성된다. 그리고 신경회로망 제어기의 학습율은 0.1, 에블레이터의 학습율은 0.3으로 하였다.

그림 6(a),(b)는 무부하에서 -500rpm ~ 500rpm으로 변화하는 기준속도를 5회, 10회 반복 인가하여 학습하였을 경우 유도전동기 실제속도와 기준속도를 보여준다. 5회만의 반복학습시에도 약간의 오버슈터가 존재하지만 기준속도를 잘 추종하고 있다. 그림 7(a),(b)는 에블레이터 출력과 유도전동기 실제 속도를 보여준다. 본 논문에서와 같이 에블레이터를 이용하여 제어기 출력단에서의 오차항을 구하게 되는 경우 에블레이터의 성능은 제어기의 성능에 직접적인 영향을 미치게 되는데, 본 논문에서 제안된 에블레이터의 구성방법은 플랜트의 입출력 관계를 잘 모의하여 주고 있음을 알 수 있다.

그림 8(a),(b)는 유도전동기 실제 속도와 제어입력 $u(k)$ 을 보여준다. 그림 9(a),(b)는 $49 \times \sin(\theta)$ [N.cm]로 변화하는 부하를 인가하였을 경우 -500rpm ~ 500rpm으로 변화하는 기준속도를 5회, 10회 반복하여 인가하여 학습하였을 경우 유도전동기 실제속도와 기준속도를 보여준다. 역시 5회만의 학습으로도 기준속도를 잘 추종함을 알 수 있다. 그림 10(a),(b)는 부하 인가시



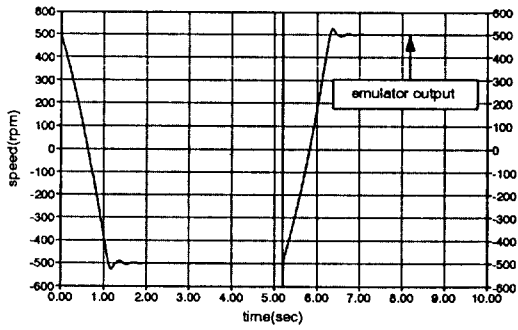
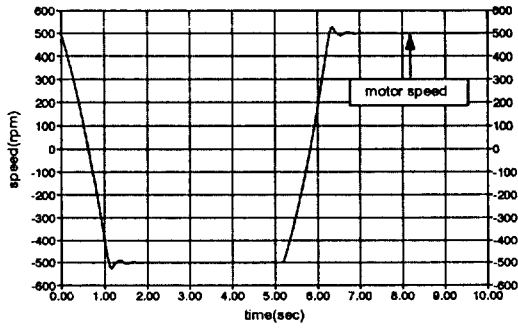
(a)



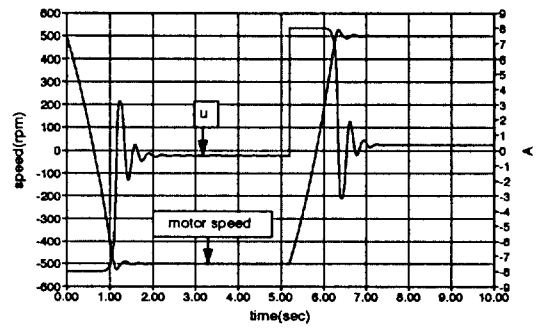
(b)

그림 6 무부하시 유도전동기 속도와 기준속도
(a) 5회 학습 (b) 10회 학습

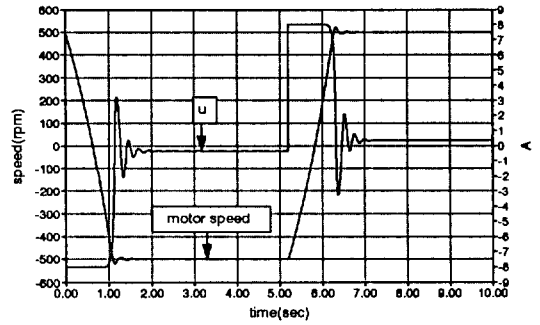
Fig. 6 Speed of induction motor and reference speed under no load
(a) after 5 training (b) after 10 training



(a)



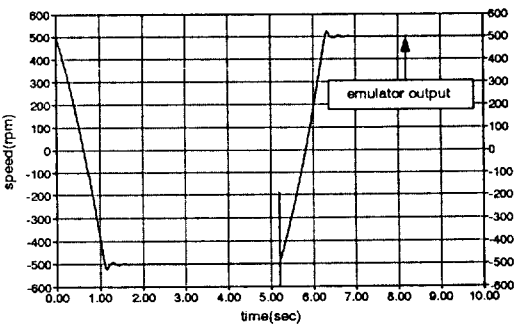
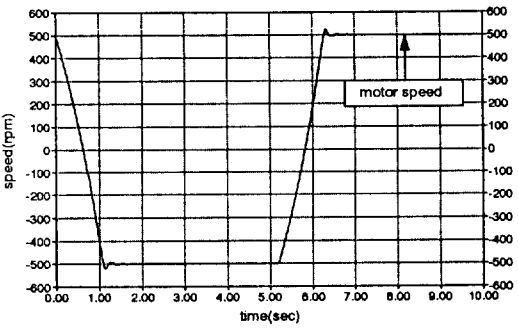
(a)



(b)

그림 8 무부하시 유도전동기 속도와 제어입력
(a) 5회 학습 (b) 10회 학습

Fig. 8 Speed of induction motor and control input under no load
(a) after 5 training (b) after 10 training



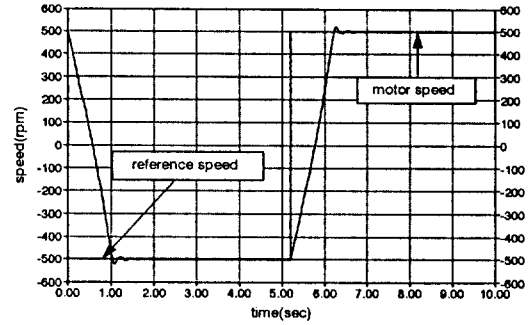
(b)

그림 7 무부하시 유도전동기 속도와 에뮬레이터 출력
(a) 5회 학습 (b) 10회 학습

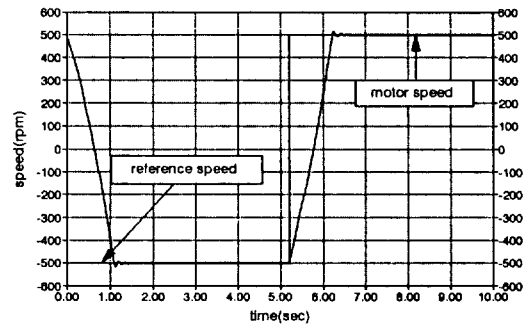
Fig. 7 Speed of induction motor and emulator output under no load
(a) after 5 training (b) after 10 training

에뮬레이터 출력과 유도전동기 실제 속도를 보여준다.

부하 인가시에도 신경회로망 에뮬레이터는 플랜트의 입력력 관계를 잘 모의하고 있음을 알 수 있다. 그림 11(a),(b)는 유도



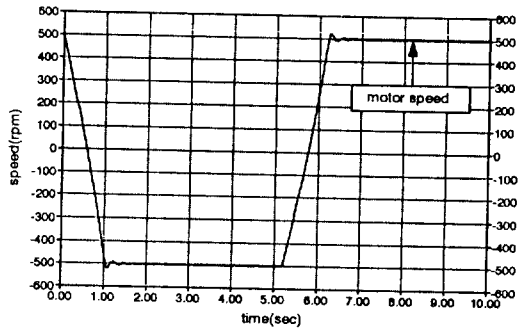
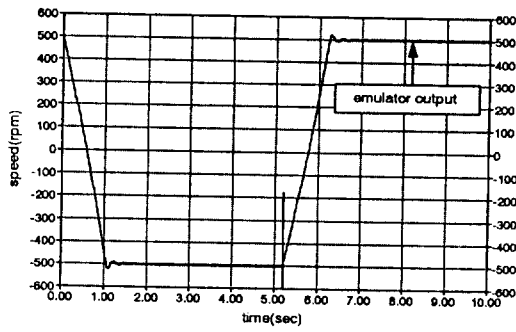
(a)



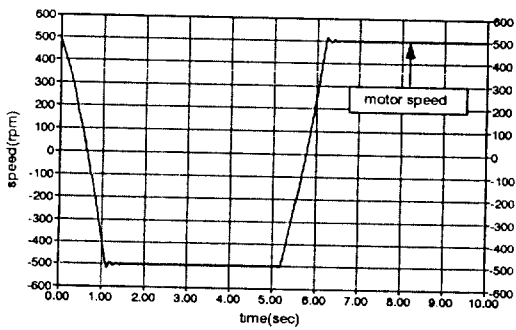
(b)

그림 9 부하인가시 유도전동기 속도와 기준속도
(a) 5회 학습 (b) 10회 학습

Fig. 9 Speed of induction motor and reference speed under load
(a) after 5 training (b) after 10 training



(a)



(b)

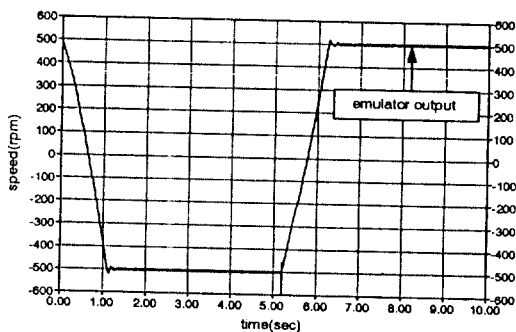
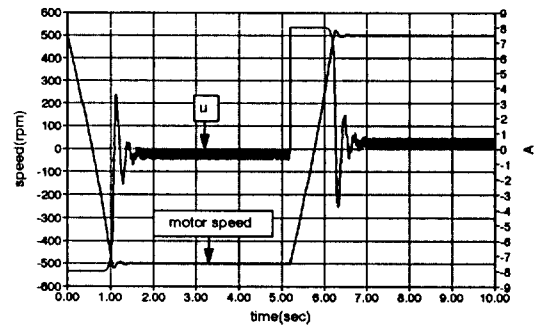


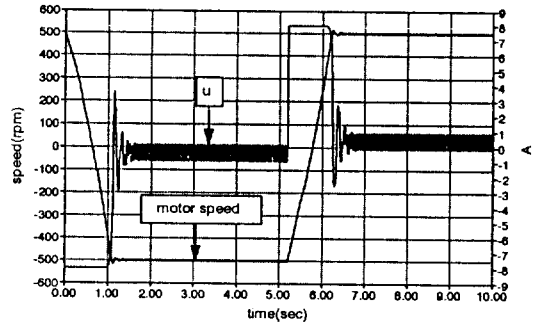
그림 10 부하인가시 유도전동기 속도와 에뮬레이터 출력
(a) 5회 학습 (b) 10회 학습

Fig. 10 Speed of induction motor and emulator output under load
(a) after 5 training (b) after 10 training

전동기 실제 속도와 제어입력 $u(k)$ 을 보여준다. 그림 12는 그림 11을 확대한 파형이다. 제어입력은 부하의 변동에 적응하기 위하여 $\sin(\theta)$ 로 변화하는 값을 출력함을 알 수 있다.



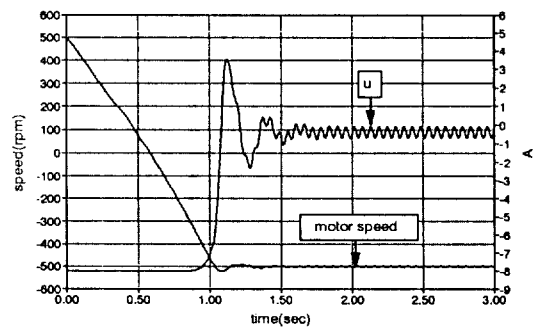
(a)



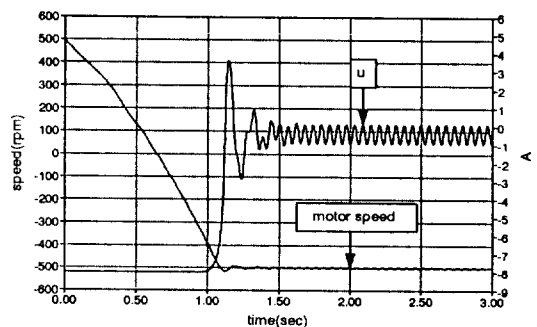
(b)

그림 11 부하인가시 유도전동기 속도와 제어입력
(a) 5회 학습 (b) 10회 학습

Fig. 11 Speed of induction motor and control input under load
(a) after 5 training (b) after 10 training



(a)



(b)

그림 12 그림11을 확대한 파형
Fig. 12 Enlarged waveform of Fig. 11

4. 실험결과 및 검토

그림 13은 유도전동기 속도제어 시스템의 하드웨어 구성도를 나타낸다.

유도전동기 속도제어 시스템의 하드웨어는 고성능 DSP TMS 320C30보드와 IGBT 전류제어형 인버터로 구성된다[8]. 모든 제어 알고리즘은 DSP인 TMS320C30에 의해 실시간으로 처리되며 제어프로그램은 C-언어와 어셈블러로 작성되고 IBM-PC486에서 에뮬레이터를 통하여 다운로드된 후 실행된다. 전력회로의 인버터 스위칭 소자로는 FUJII사의 1500[V],75[A]의 정격을 가지는 6MB150-120을 사용하였다. 또한 3상 기준 전류를 출력하기 위하여 4채널 12bit D/A converter인 AD394를 이용하였다. 전류제어기는 히스테리시스 전류제어기를 사용하였다.

그림 14는 전체 제어프로그램 순서도를 보여준다. 그림 14(a)의 C-언어로 작성된 주 프로그램에서는 먼저 내부 레지스터 및 각종 변수를 초기화 시키고 속도 측정 인터럽터를 위한 내부 타이머0 인터럽터를 Start시킨후 2[ms]의 주기로 무한 반복되는 신경회로망 속도제어 루프로 들어간다. 그림 14(b)는 내부 타이머0를 이용하여 매 2ms마다 속도 측정 인터럽트를 인에이블 시켜주는 것을 보여 준다. 제어 루프에서는 속도 측정 루틴에서 측정된 속도정보로부터 회전자 위치 θ 를 계산하고, 신경회로망 제어기의 출력을 구한후 이 값을 신경회로망 에뮬레이터와 유도전동기의 입력값으로 이용한다. 그 다음 유도전동기의 실제출력과 신경회로망 에뮬레이터 출력과의 오차를 계산하고 이를 이용하여 에뮬레이터를 학습시킨 후, 다시 기준속도와 유도전동기 실제속도와의 오차항을 에뮬레이터를 통하여 역전과 시켜 신경회로망 제어기의 출력층에서의 오차항을 계산한후 이 값을 이용하여 신경회로망 제어기를 학습 시키게 된다. 한편 그림 14(c)의 속도측정 루틴에서는 1회전당 1000펄스를 발생하는 엔코더펄스를 4배배하여 M/T 속도측정방법을 이용하여 속도를 계산하며 속도 측정 주기는 2[ms]+ Δt 가 된다. 그림 14(c)에서 m1,m2는 각각 encoder pulse와 clock pulse이다. 또

한 회전방향검출 회로를 통하여 회전방향을 검출하여 정역회전이 가능하게 하였다. 그리고 그림 15와 같이 길이 L인 강체의

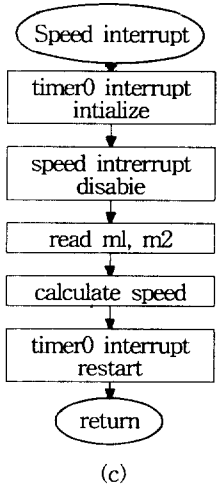
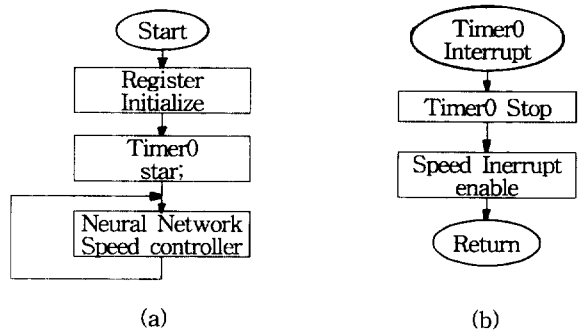


그림 14 제어 프로그램 순서도
Fig. 14 Flow chart of control program

표 1 시뮬레이션과 실험에 사용된 전동기 상수 및 정격

전압 : 220[V]	출력 : 2.2[Kw]
극수 : 4[극]	회전수 : 1735[rpm]
주파수 : 60[Hz]	
$R_s : 0.687[\Omega]$	$L_s : 0.08397[H]$
$R_r : 0.842[\Omega]$	$L_r : 0.08525[H]$
$M : 0.08136[H]$	$J : 0.03Kg \cdot m^2$
$B : 0.01[kg \cdot m^2/sec]$	

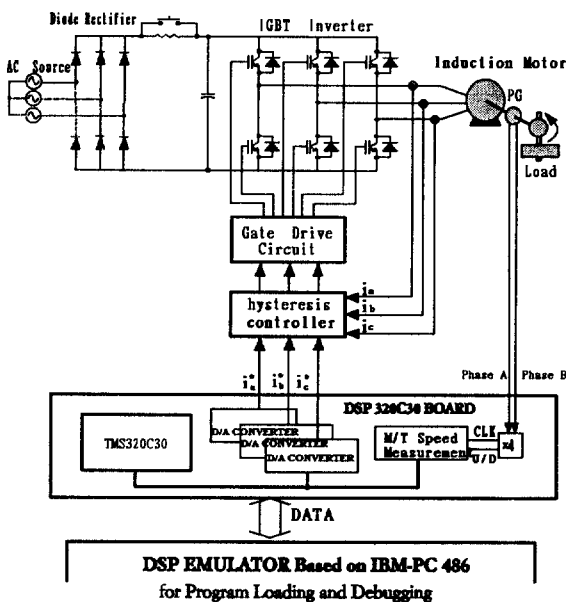


그림 13 하드웨어 구성도
Fig. 13 Block diagram of hardware

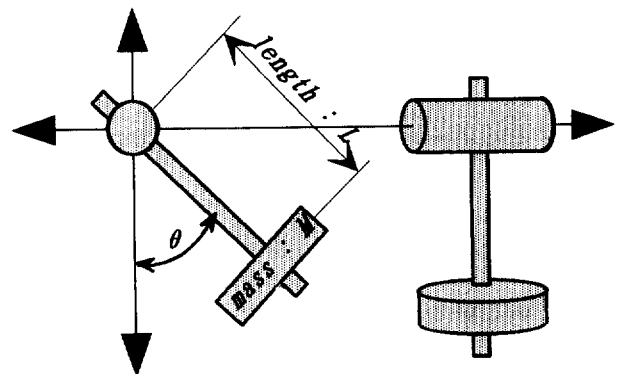
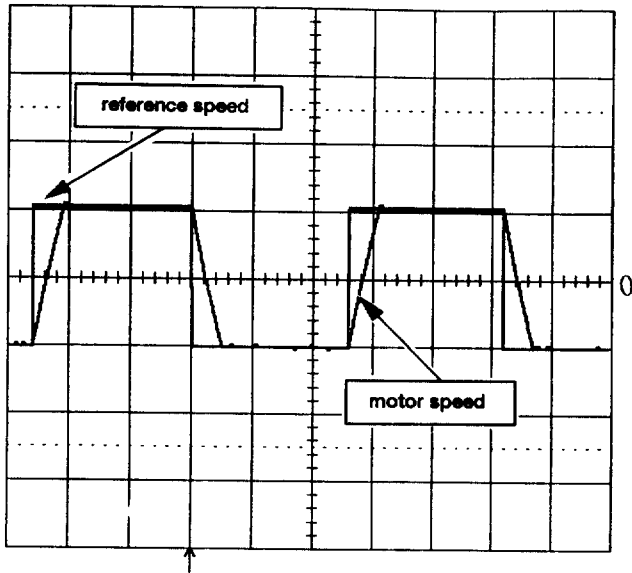
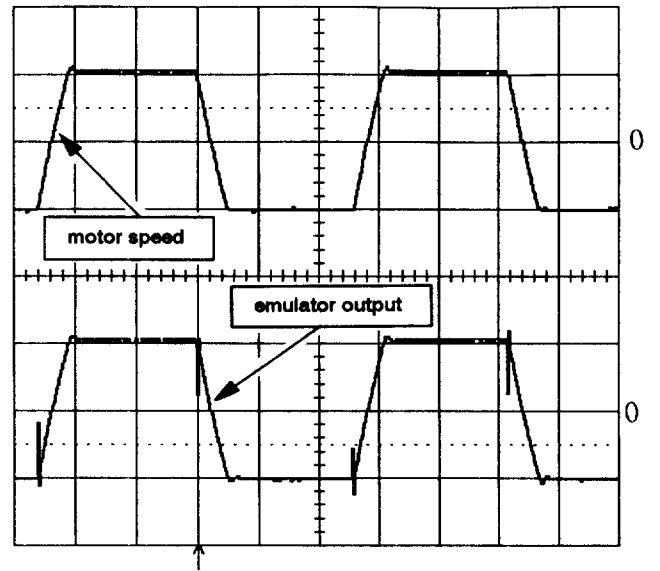


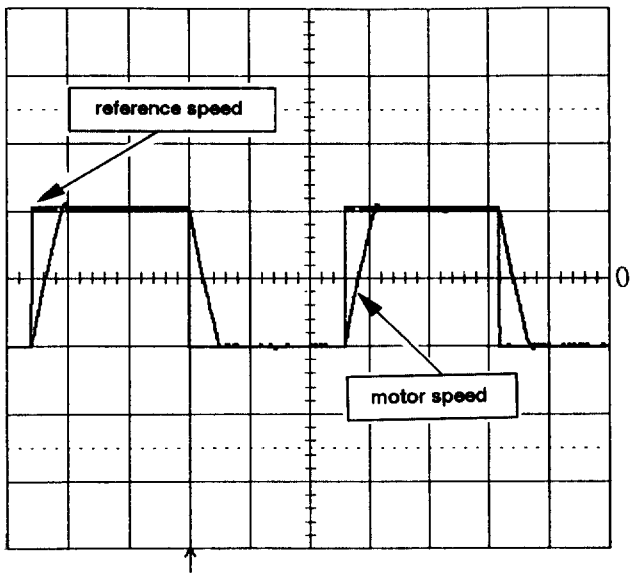
그림 15 전동기 부하
Fig. 15 Load of motor



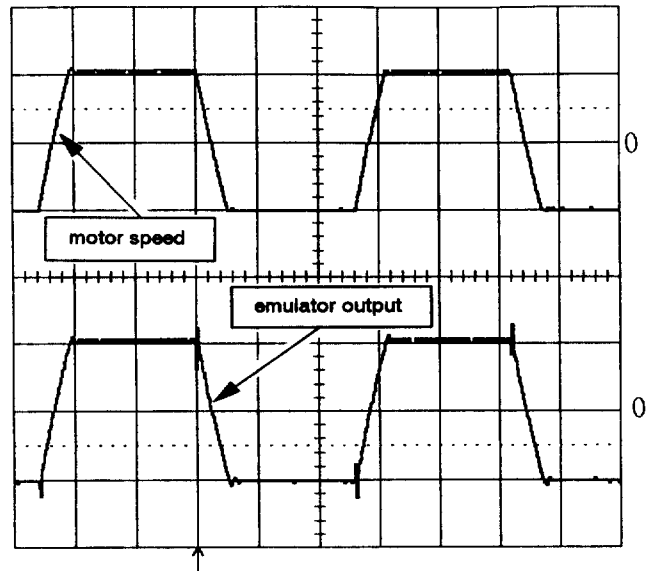
(a)



(a)



(b)



(b)

그림 16 무부하시 유도전동기 속도와 기준속도
(a) 5회 학습 (b) 10회 학습
(X 축 : 2 sec/div , Y축 : 500 rpm/div)

Fig. 16 Speed of induction motor and reference speed under no load
(a) after 5 training (b) after 10 training
(X axis : 2 sec/div , Y axis : 500 rpm/div)

그림 17 무부하시 유도전동기 속도와 에뮬레이터 출력
(a) 5회 학습 (b) 10회 학습
(X 축 : 2 sec/div , Y축 : 500 rpm/div)

Fig. 17 Speed of induction motor and emulator output under no load
(a) after 5 training (b) after 10 training
(X axis : 2 sec/div , Y axis : 500 rpm/div)

한쪽끝에 무게 M인 디스크부하를 장착하고 강체의 다른쪽 끝을 전동기 축에 연결한 장치를 구성하여 전동기 축이 회전함에 따라 정현적으로 변하는 부하에 대하여 실험을 행하였다. 표 1은 시뮬레이션과 실험에서 사용된 유도전동기 파라미터이다.

그림 16(a),(b)는 무부하에서 -500rpm ~ 500rpm으로 변화하는 기준속도를 5회, 10회 반복하여 인가하여 학습하였을 경우

유도전동기 실제속도와 기준속도를 보여준다. 5회 학습의 경우에도 기준속도를 잘 추종함을 알 수 있다. 그림 17(a),(b)는 에뮬레이터 출력과 유도전동기 실제 속도이며 에뮬레이터가 유도전동기의 입출력 관계를 잘 모의 하고 있음을 알 수 있다.

그림 18(a),(b)는 유도전동기 실제 속도와 제어입력 $u(k)$ 을 보여준다. 그리고 그림 19(a),(b)는 $t=0$ 에서 $49 \times \sin(\theta)$ [N.cm]

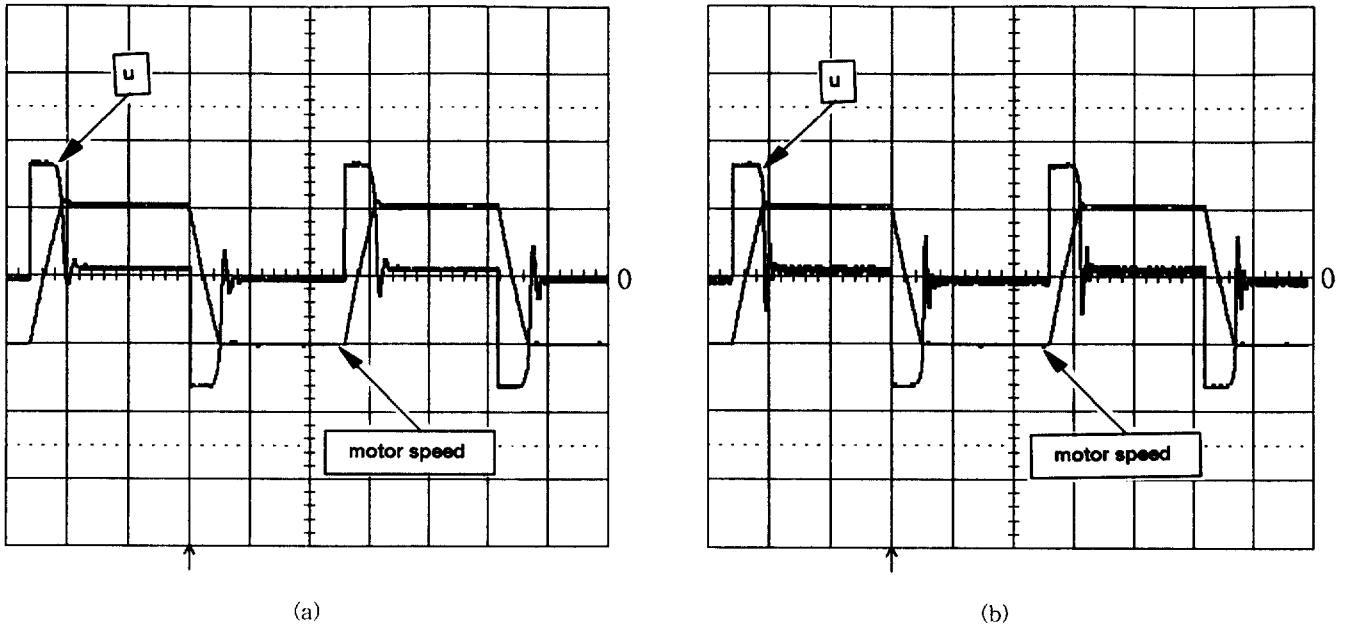


그림 18 무부하시 유도전동기 속도와 제어입력
 (a) 5회 학습 (b) 10회 학습
 (X 축 : 2 sec/div , Y축 : 500 rpm/div, 5A/div)
Fig. 18 Speed of induction motor and control input under no load
 (a) after 5 training (b) after 10 training
 (X 축 : 2 sec/div , Y축 : 500 rpm/div, 5A/div)

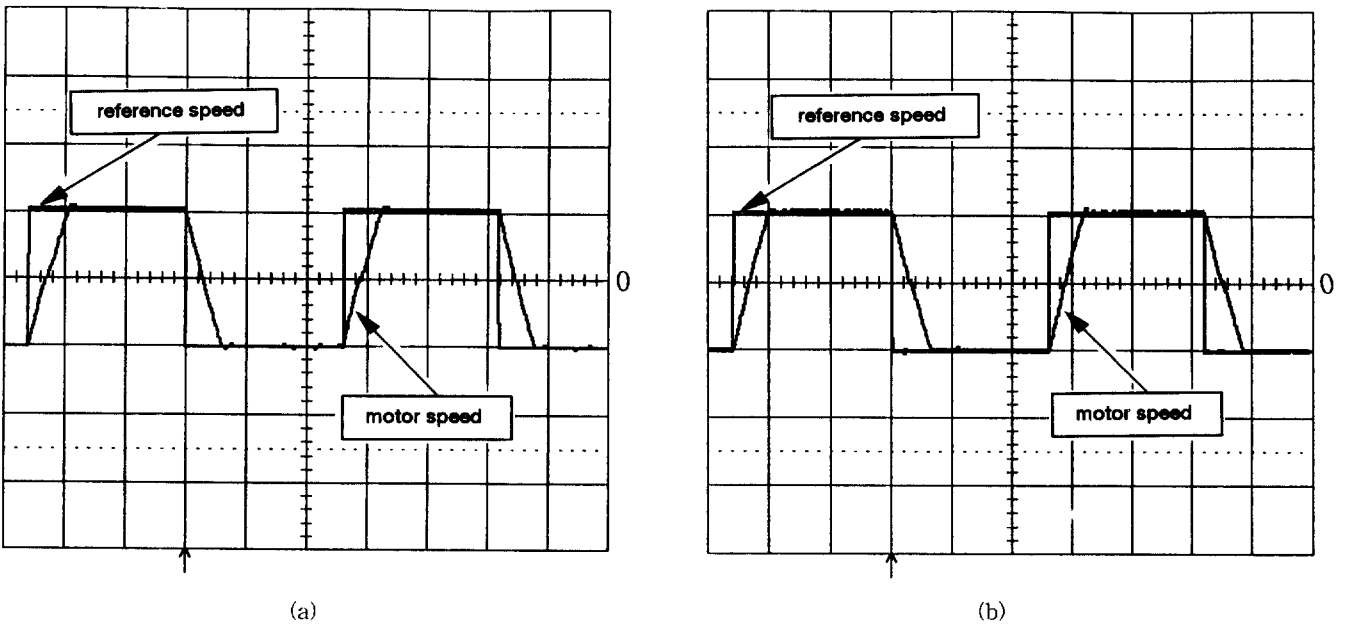


그림 19 부하시인가시 유도전동기 속도와 기준속도
 (a) 5회 학습 (b) 10회 학습
 (X 축 : 2 sec/div , Y축 : 500 rpm/div)
Fig. 19 Speed of induction motor and reference speed under load
 (a) after 5 training (b) after 10 training
 (X axis : 2 sec/div , Y axis : 500 rpm/div)

로 변화하는 부하를 인가하고 -500rpm ~ 500rpm으로 변화하는 기준속도를 5회, 10회 반복하여 학습하였을 경우 유도전

동기 실제속도와 기준속도를 보여준다. 그림 20(a),(b)는 예블레이터 출력과 유도전동기 실제 속도를 보여준다. 그리고 그림

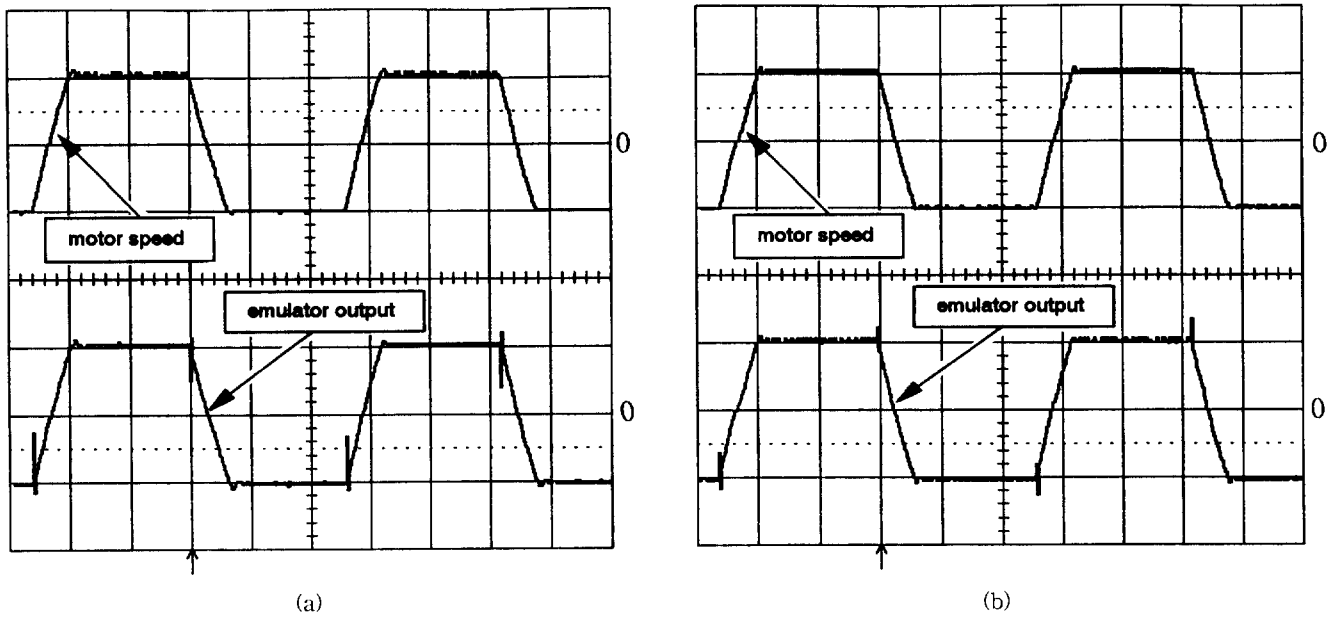


그림 20 부하인가시 유도전동기 속도와 에뮬레이터 출력
 (a) 5회 학습 (b) 10회 학습
 (X 축 : 2 sec/div , Y축 : 500 rpm/div)

Fig. 20 Speed of induction motor and emulator output under load
 (a) after 5 training (b) after 10 training
 (X axis : 2 sec/div , Y axis : 500 rpm/div)

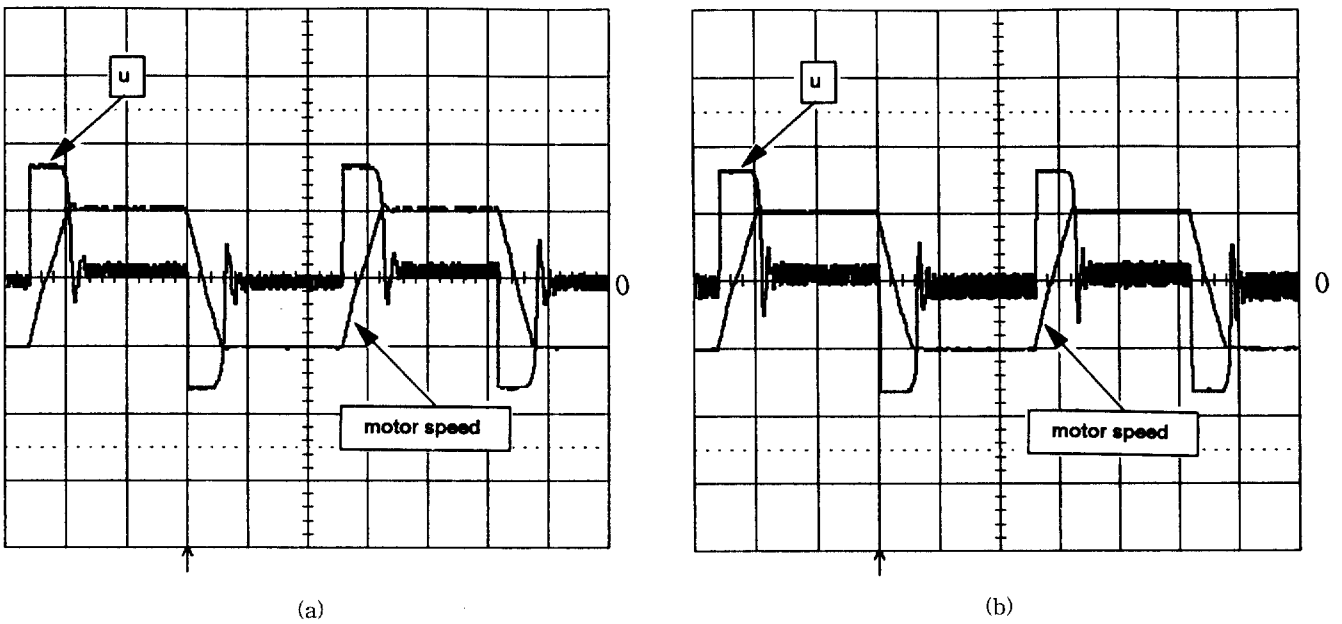
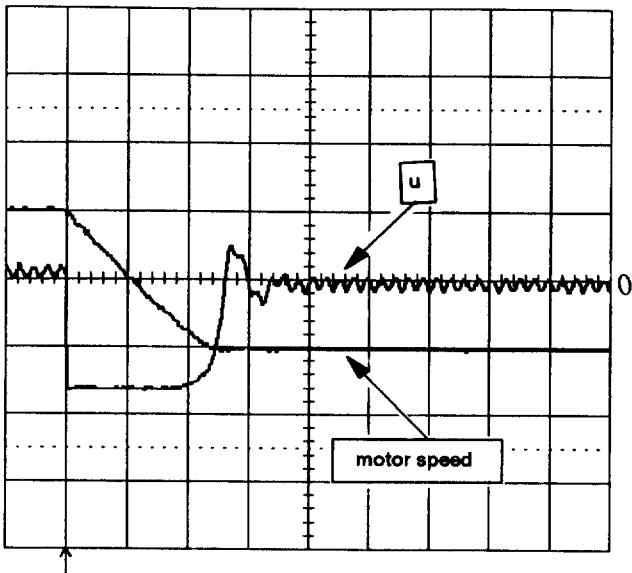


그림 21 부하인가시 유도전동기 속도와 제어입력
 (a) 5회 학습 (b) 10회 학습
 (X 축 : 2 sec/div , Y축 : 500 rpm/div, 5 A/div)

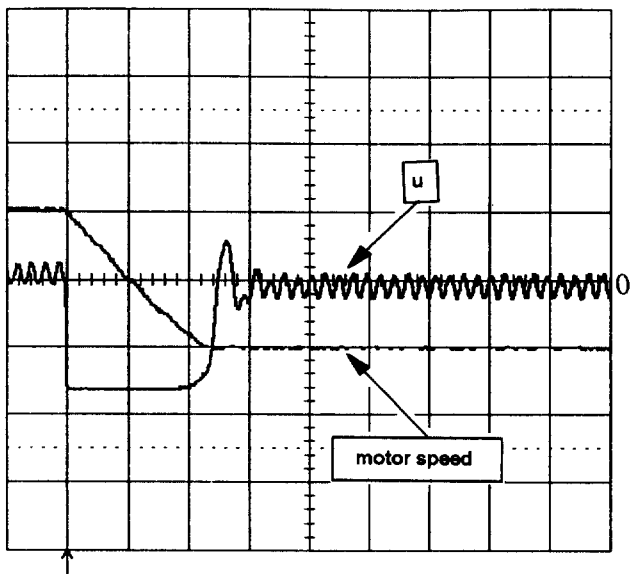
Fig. 21 Speed of induction motor and control input under load
 (a) after 5 training (b) after 10 training
 (X axis : 2 sec/div , Y axis : 500 rpm/div, 5 A/div)

21(a),(b)는 유도전동기 실제 속도와 제어입력 $u(k)$ 을 보여준다.
 부하 인가시에도 역시 기준속도를 잘 추종하고 있으며 부하

의 변동에 대하여 강인성을 보여 주고 있다.



(a)



(b)

그림 22 그림 21을 확대한 파형
 (X 축 : 0.5 sec/div , Y축 : 500 rpm/div, 5 A/div)
 Fig. 22 Enlarged waveform of Fig.21
 (X axis : 0.5 sec/div , Y axis : 500 rpm/div, 5 A/div)

5. 결 론

본 논문에서는 신경회로망에 의한 백터제어 유도전동기 속도 제어 시스템을 제안하여 아래의 결과를 얻었다.

- (1) 제어기와 에플레이터의 입력을 기준 속도에 대한 오차와 그 오차의 변화를 이용함으로써 신경회로망의 구조를 간단하게 하였다.
- (2) 신경회로망을 이용한 유도전동기 속도제어 시스템에서 지도학습을 통하여 학습을 할 경우 신경회로망 제어기 출력단에서의 오차항을 구하는 문제를 신경회로망 에플

레이터를 도입함으로써 해결하였다.

- (3) 컴퓨터 시뮬레이션과 실험결과 신경회로망 제어기는 5회 정도의 학습만으로 기준 속도를 잘 추종하였고, DSP를 이용한 실시간 온라인 학습으로 부하의 변동에도 강인성을 가짐을 알 수 있다. 신경회로망 에플레이터 또한 부하의 변동에 관계없이 유도전동기의 입출력 관계를 잘 모의 하였다.

향후 신경회로망 제어기와 에플레이터의 구성에 있어서 뉴런의 갯수와 은닉층의 갯수가 제어의 성능에 미치는 영향과 최적의 신경망 제어기의 구조 선정 문제에 관한 연구가 필요하다.

참 고 문 헌

- [1] K. Kenzo, O. Tsutomu, and S. Taskashi, "Application Trends in AC Motor Drives", IEEE IECON'92, pp.31-36, 1992.
- [2] 강문호, 김남정, 유지윤, 박귀태, 민경일, "부하변동에 대한 유도전동기의 속도제어 특성에 관한 연구", 대한전기학회 논문지, 제43권, 7호, pp.1075 - 1083, 1994. 7.
- [3] Hau-Chuen Chan, K.T.Chau, and C.C.Chan, "A Neural Network Controller for Switching Power Converter", IEEE PESC'93 , pp.887 - 892, 1993.
- [4] John Bates, Maik E.Elbuluk, Donald S.Zinger, "Neural Network Control of a Chopper-Fed DC Motor", IEEE PESC '93 Record, pp.893-899, 1993.
- [5] 김세찬, 김학성, 류홍제, 원충연, "신경회로망을 이용한 유도 전동기 속도제어에 관한 연구", 대한 전기학회 전력전자 연구회 춘계학술대회, pp.11-15, 1995. 5.
- [6] K. J. Hunt, D.Sbarbaro, R.Zbikowski, and P.J. Gawthrop, "Neural Networks for Control System-A Survey", Automatica, vol.28, No.6, pp.1083-1112, 1992.
- [7] D. E. Rumelhart, J. L. McClelland, and The PDP Research Group, Parallel Distributed Processing. Vol.1-2, MIT Press, 1986.
- [8] Texas Instrument, TMS320C30x user's guide, 1990.

저 자 소 개



김 세 찬 (金世爜)

1963년 7월 7일생. 1989년 성균관대 공대 전기공학과 졸업. 1992년 성균관대 대학원 전기공학과 졸업(석사). 1995년 성균관대 대학원 전기공학과 박사과정 수료. 현재 성균관대 공대 전기공학과 시간강사



원 충 연 (元忠淵)

1955년 5월 10일생. 1978년 성균관대 공대 전기공학과 졸업. 1987년 서울대 대학원 전기공학과 졸업(공학박사). 1991~1992년 미국 University of Tennessee 전기공학과 객원교수. 현재 성균관대 공대 전기공학과 부교수