Journal of the Korean Institute of Illuminating and Electrical Installation Engineers Vol. 13, No.4, pp. 109~119 November 1999 [논문 : 13-4-16]

퍼지-신경망 제어기를 이용한 스위치드 리럭턴스 전동기의 속도제어

(A Speed Control of Switched Reluctance Motor using Fuzzy-Neural Network Controller)

박지호* · 김연충** · 원충연*** · 김창림° · 최경호°°

(Ji-Ho Park · Yuen-Chung Kim · Chung-Yuen Won · Chang-Lim Kim · Kyung-Ho Choi)

요 약

스위치드 리럭턴스 전동기(SRM)는 상대적으로 낮은 가격, 간단하고 견고한 구조, 제어의 용이성과 고효율을 가지기 때문에 가변속 구동에서 점점 응용범위가 확대되고 있다.

본 논문에서 신경망이론은 퍼지-신경망 제어기의 소속함수와 퍼지규칙을 결정하는데 사용하였으며, 신경망 에뮬레이터는 SRM의 전방향 동특성을 모사하는데 사용하였다. 에뮬레이터의 역전파 오차는 퍼지-신경망 제어기의 소속함수와 퍼지규칙을 개선하는 경로를 제공한다.

32비트 DSP(TNS329C31)는 고속연산과 퍼지-신경망 제어 알고리즘을 실현하는데 사용하였다.

시뮬레이션과 실험결과는 부하변화의 경우 제안된 제어방법이 속도응답에서 종래의 방법보다 우수하였다.

Abstract

Switched Reluctance Motor(SRM) have been expanding gradually their applications in the variable speed drives due to their relatively low cost, simple and robust structure, controllability and high efficiency.

In this paper neural network theory is used to determine fuzzy-neural network controller's membership functions and fuzzy rules. In addition neural network emulator is used to emulate forward dynamics of SRM and to get error signal at fuzzy-neural controller output layer. Error signal is backpropagated through neural network emulator. The backpropagated error of emulator offers the path which reforms the fuzzy-neural network controller's membership functions and fuzzy rules.

32bit Digital Signal Processor(TMS320C31) was used to achieve the high speed control and to realize the fuzzy-neural control algorithm.

Simulation and experimental results show that in the case of load variation the proposed control method was superior to a conventional method in the respect of speed response.

---

* 정회원 : 국방과학연구소 연구원
** 정회원 : 성균관대학교 전기전자 및 컴퓨터공학부 박사과정
*** 정회원 : 성균관대학교 전기전자 및 컴퓨터공학부 교수
o 정회원 : (주)효성 전동기부 부장
oo 정회원 : (주)효성 연구원
접수일자 : 1999년 8월 29일

1. 서 론

일반적으로 가변속 운전 분야에서는 직류 전동기가 속도제어가 쉽다는 이유로 많이 사용되어 왔다. 그러나 직류 전동기의 경우 정류자편과 브러시를 사용함에 따라 구조가 복잡해 제작단가 및 마모에 따른 유지비가 많이 들며, 보수가 어려운 단점이 있다. 전자기적인 구조가 간단하고 견고하며 고효율, 고토오크/관성비, 넓은 범위의 가변속 운전 등의 장점을 가진 전동기구로써 여러 산업분야 및 가전분야에 그 적용범위가 증가되고 있다[1-2].

SRM의 가변속 제어에 가장 많이 사용되고 있는 PI 제어기의 경우 알고리즘이 간단하여 아날로그 제어기로도 쉽게 구현이 가능하고, 제어 이득이 적절히 조정될 경우 좋은 운전 특성을 지니는 장점을 지니고 있으나, 운전점이 달라지거나 시스템의 파라미터가 변하는 경우 제어 이득을 적절히 조정해 주어야 하는 문제가 있다. 특히 SRM의 경우 비선형 인덕턴스 특성을 지니는 이유로 수학적 모델링이 불가능하고, 정상상태에서의 등가회로가 존재하지 않기 때문에 적절한 제어 이득을 얻는데 있어 실험적인 방법에 의존하고 있다. 이에 따라 1990년대부터 신경망과 퍼지 이론에 의한 지능형 제어방식이나 가변구조제어, 적응제어 이론과 같은 현대 제어방식으로 SRM의 제어성능을 개선하고자 하는 연구가 많이 이루어지고 있다[3-4].

퍼지 논리는 퍼지 집합론을 기준으로 한 제어 이론으로 제어 대상 플랜트를 수학적으로 모델링하는 대신 입력과 출력의 관계만을 통하여 제어 규칙을 생성하므로 플랜트의 수학적 모델링이 필요없고, 외란에 강인한 특성을 가지고 있다. 그러나 퍼지 제어기는 제어 규칙, 소속함수, 입출력 이득을 생성하기 위한 일반적 규칙이 존재하지 않고 전문가의 지식에 의존하며, 설계된 범위 밖의 입력값에 대해서는 적절한 제어량을 만들어 내지 못하는 단점을 지니고 있다[5,6].

한편 신경망 제어기는 PDP그룹에 의해 1980년대 중반에 은닉층을 가진 신경회로망 모델과 오차 역전파 학습 알고리즘이 발표된 이후, 구조 설계 문제의 어려움 등에도 불구하고 간단한 프로그래밍, 정보의 분산처리, 환경 변화에 대한 적응성, 외란에 대한 강인성 등의 특성으로 인하여 전력전자 분야에서 활발히 연구·응용되고 있다. 그러나 대부분의 경우 오프라인으로 학습되거나 학습에 많은 시간이 걸리는 문제점이 있다. 이러한 퍼지 및 신경망의 문제점으로 인해 1980년대 말부터 퍼지이론과 신경회로망 이론을 결합하여 상호 단점을 보완하고자 하는 연구가 진행되어왔다[7,8].

본 논문에서는 SRM의 속도제어를 위해 신경망 에뮬레이터를 이용하여 퍼지 제어기의 소속함수와 제어규칙을 실시간으로 학습하여 실시간 플랜트의 동정이 가능한 퍼지-신경망 제어기를 사용하였으며, 현재 SRM의 실용화에 문제점으로 지적되고 있는 토오크 리플을 저감하기 위해 히스테리시스 전류제어 방식을 사용하였다. 또한 경제성을 가진 구동 시스템을 설계하기 위해서 고가의 절대 엔코더에 비해 저렴한 증분형 엔코더를 사용하였으며, 초기 기동시 회전자의 정확한 위치를 판별하기 위한 방법을 제안하였다. 시스템의 구현은 연산 속도가 빠르며 실시간 부동 소수점 연산이 가능한 32bit DSP TMS320C31을 사용하였으며 기존의 제어방법과 제시된 제어방법의 특성을 시뮬레이션과 실험을 통해 비교 분석하여 제안된 제어 방법의 우수성을 확인하였다.

2. SRM의 동작원리

그림 1은 본 논문에서 사용한 고정자 극수가 6개, 회전자 극수가 4개인 3상 6/4극 SRM의 단면도이다. SRM은 고정자와 회전자가 모두 돌극형 구조로, 고정자에는 권선이 감겨져 있으나 회전자에는 권선이나 영구자석이 없는 규소강판을 적층한 구조로 되어 있다.

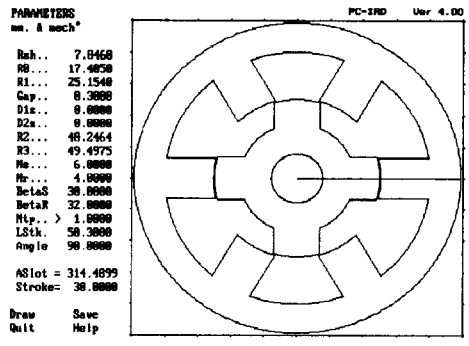


그림 1. 6/4극 SRM의 고정자 및 회전자 구조
Fig. 1. The structure of the stator and rotor of 6/4 pole SRM

그림 2에서와 같이 인덕턴스가 증가하는 구간에서 고정자상에 전류를 흘려주면 회전자를 정렬위치로 회전시킬 수 있고, 6/4극 전동기의 경우 각 상의 인덕턴스 곡선은 30°씩 위상차를 가지고 변화하므로 전동기를 지속적으로 회전시키기 위해서는 순차적으로 스위칭하여 각 상의 인덕턴스가 증가하는 구간에만 전류를 흘려주는 것이 필요하다. 만약 전류가 회전방향에 대해 인덕턴스가 감소하는 구간에서 흐르게 되면 역토오크가 발생될 수 있으며, 이 경우 전동기의 회전을 정지시키거나 발전기 모드로 동작시키게 된다. SRM의 순시토오크 T 는 식 (1)로 표현된다. 여기서, L 은 회전자의 현재 위치에서의 인덕턴스이며, i 는 순시 전류이다[2],[9].

$$T = \frac{1}{2} i^2 \frac{dL}{d\theta} \tag{1}$$

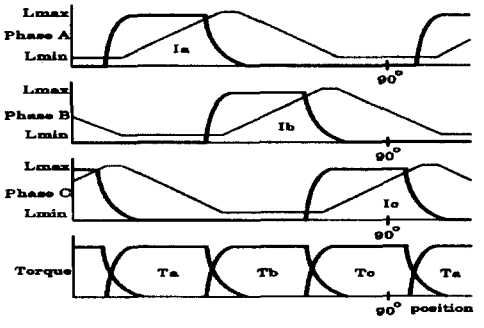


그림 2. 6/4극 SRM의 회전토오크
Fig. 2. The rotation torque of 6/4 pole SRM

3. SRM 구동을 위한 퍼지-신경망 제어기 설계

3.1. 신경망 에뮬레이터의 구성과 학습

신경망은 신호 전달이 한 방향으로 진행하고 있는 구조이고, 입력층, 은닉층, 출력층을 포함하고 있다. 신경망 학습을 위해서 역전파 알고리즘을 사용하였다. 이 역전파 학습 알고리즘의 기본 원리는 출력값과 기대값을 비교하여 그 차이를 줄여가는 방향으로 접속강도(weight)를 조정하고, 상위층에서 역전파하여 하위층에서 이를 근거로 다시 자기층의 접속강도를 조정해 나가게 된다.

신경망의 제어계통을 구성하는 방법으로 직접적인 제어방법과 간접적인 제어방법으로 분류할 수 있다. 직접적인 제어방법은 시스템의 쟈코비안을 알아야 하지만, 간접적인 제어방법에서는 시스템의 쟈코비안을 알 필요가 없는 대신에 비선형 시스템을 동정하는 부가적인 에뮬레이터를 구성하여야 한다. SRM을 제어하는 경우에는 제어대상에 대한 파라미터를 알지 못하고 제어대상의 입출력에 대한 정보만 얻을 수 있으므로 본 논문에서는 간접적인 제어방법을 사용하였다[10].

그림 3에서 신경망 에뮬레이터의 입력으로 기준값 $r(k)$ 와 플랜트 실제 출력 $y(k)$ 와의 오차항 $se(k)$, 그 오차의 변화분 $sce(k)$, 그리고 제어기 출력 $u(k)$ 를 입력으로 이용한다. 여기서 신경망 에뮬레이터의 출력으로 나오는 값은 실제 플랜트의 출력을 모사한다. 즉, 신경망 에뮬레이터의 기능은 입출력 패턴을 이용하여 실제 플랜트의 동특성을 모의하는 것으로 에뮬레이터는 플랜트의 전달함수와 동일한 역할을 수행한다[13].

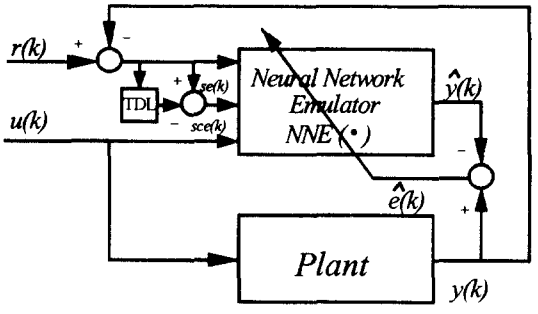


그림 3. 신경망 에뮬레이터의 개략도
Fig. 3. The block diagram of neural network emulator

학습된 신경망 에뮬레이터는 식 (2)와 같이 플랜트의 전방향 동특성을 모의할 수 있다.

$$\hat{y}(k) = NNE(se(k), sce(k), u(k)) \tag{2}$$

여기서 신경망 에뮬레이터 $NNE(\cdot)$는 비선형 매핑을 나타내며, $\hat{y}(k)$는 신경망 에뮬레이터 출력으로서 실제 플랜트 출력값의 추정치가 된다.

학습이 완료되면

$$\hat{e}(k) = y(k) - \hat{y}(k) \simeq 0 \tag{3}$$

가 된다.

신경망 에뮬레이터의 또 다른 기능은 퍼지-신경망 제어기 출력층에서의 오차항을 계산하기 위하여 플랜트 출력단에서의 오차항을 역전파시키기 위한 오차 역전파 경로를 제공하는 것이다. 다시 말해서 퍼지-신경망 제어기를 학습시키기 위한 지도 입력에 대한 지도 출력이 없으므로 퍼지-신경망 제어기 출력층에서의 오차항은 플랜트 출력단에서 기준 출력값과 실제 플랜트 출력값과의 오차를 이 에뮬레이터를 통하여 역전파시켜 제어기 출력층에서의 오차항을 계산하는 것이다.

그림 4는 SRM의 동특성을 모의하기 위하여 사용한 신경망 에뮬레이터로써 3개의 입력층 뉴런, 5개의 은닉층 뉴런, 1개의 출력층 뉴런으로 구성하였다.

신경망 에뮬레이터를 학습시키기 위한 학습 과정을 살펴보면, 먼저 오차항 $\hat{e}(k)$는 다음과 같다.

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

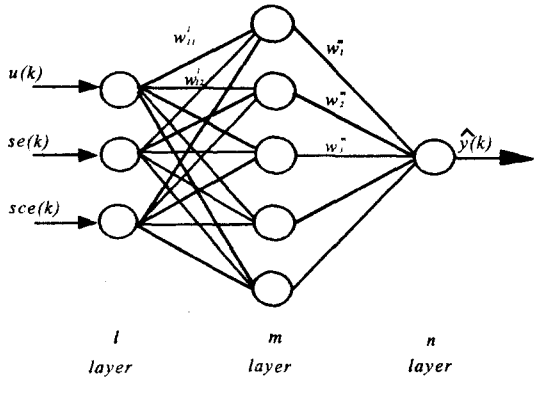


그림 4. 신경망 에뮬레이터의 구조
Fig. 4. The structure of neural network emulator

여기서 $y(k)$는 플랜트의 출력, $\hat{y}(k)$는 에뮬레이터 출력이다. 학습은 식 (4)를 최소화하도록 일반화된 델타 규칙에 의해 이루어진다.

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

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

$$\Delta W_j^m = \eta \delta^n out_j^m \tag{6}$$

여기서, η 는 학습률이다.

δ^n는 n번째 층의 뉴런의 오차항이며 다음과 같이 구해진다.

$$\delta^n = \frac{\partial \hat{e}(k)}{\partial net^n} = \frac{\partial \hat{e}(k)}{\partial out^n}\frac{\partial out^n}{\partial net^n} \tag{7}$$

위 식에서 우변의 두 번째 항은 다음과 같다.

$$\frac{\partial out^n}{\partial net^n} = f'(net^n) \tag{8}$$

결국 식 (11)을 이용하여 출력층 뉴런에서의 오차항은 다음과 같이 표현된다.

$$\delta^n = -f'(net^n)(y(k) - \hat{y}(k)) \tag{9}$$

은닉층의 경우에는 지도 출력값을 알 수 없으므로 이미 알고 있는 출력층에서의 오차항을 이용하면 다음과 같이 구할 수 있다.

$$\begin{aligned}\frac{\partial \hat{e}(k)}{\partial out_i^m} &= \frac{\partial \hat{e}(k)}{\partial net^n}\frac{\partial net^n}{\partial out_i^m} \\ &= \frac{\partial \hat{e}(k)}{\partial net^n}\frac{\partial}{\partial out_i^m}\sum_i out_i^m W_i^m \\ &= \frac{\partial \hat{e}(k)}{\partial net^n} W_i^m = \delta^n W_i^m \end{aligned} \tag{10}$$

위의 결과를 식 (13)에 대입하면 은닉층의 경우 오차항은 다음과 같이 표시된다.

$$\delta_i^m = f'(net_i^m)\,\delta^n W_i^m \tag{11}$$

활성함수로서 $out_h^k = \tanh(net_j^k)$ 를 이용하면 출력층의 경우

$$\begin{aligned}\delta^n &= -f'(net^n)(y(k) - \hat{y}(k)) \\ &= -(1-(\hat{y}(k))^2)(y(k) - \hat{y}(k))\end{aligned} \tag{12}$$

은닉층의 경우

$$\begin{aligned}\delta_i^m &= f'(net_i^m)(\delta^n W_i^m) \\ &= (1-(out_i^m)^2)(\delta^n W_i^m)\end{aligned} \tag{13}$$

가 된다.

일반적으로 보다 빠른 학습 속도를 위해 식 (5)에 모멘텀(momentum)항이 추가되는데 이 경우 가중치 조절은 다음과 같이 표현된다.

$$\Delta W_j^m(k+1) = \eta \delta^n out_j^m + \alpha \Delta W_j^m(k) \quad (14)$$

$$W_j^m(k+1) = W_j^m(k) + \Delta W_j^m(k+1) \quad (15)$$

여기서 k 는 학습 반복 횟수, η ($0<\eta<1$)는 학습률, α ($0<\alpha<1$)는 관성항이다.

3.2 퍼지-신경망 제어기의 구성과 학습

본 논문에서 사용한 퍼지-신경망 제어기는 퍼지 제어기의 제어규칙을 자동생성하고 소속함수를 자동 조정한다.

그림 5는 퍼지-신경망 제어기를 나타낸 것으로 퍼지 제어기의 퍼지화에 해당하는 입력층, 제어규칙을 구성하는 은닉층, 비퍼지화에 해당하는 출력층 등의 3층 구조를 갖고 있다.

퍼지-신경망 제어기의 입력 $x_1(k)$, $x_2(k)$ 로는 기준속도 $r(k)$ 와 실제속도 $y(k)$ 의 오차

$$x_1(k) = se(k) = r(k) - y(k) \quad (16)$$

그리고 오차의 변화율

$$x_2(k) = sce(k) = se(k) - se(k-1) \quad (17)$$

을 선택하였다. 그리고 각 층의 뉴런들은 상호 연결되어 있으며 연결강도는 가중치 W_i 로 표시하였다.

입력층에서는 각 입력 변수에서 각 언어변수의 소속함수값에 해당하는 $A_i(x_1)$와 $B_i(x_2)$를 출력한다. 여기서 사용된 각 입력에 대한 퍼지 변수의 소속함수값은

$$A_i(x_1) = \frac{1}{1+[(\frac{x_1-c_{i1}}{2a_{i1}})^2]^{b_{i1}}} \quad (18)$$

$$B_i(x_2) = \frac{1}{1+[(\frac{x_2-c_{i2}}{2a_{i2}})^2]^{b_{i2}}} \quad (19)$$

에 의해 계산된다.

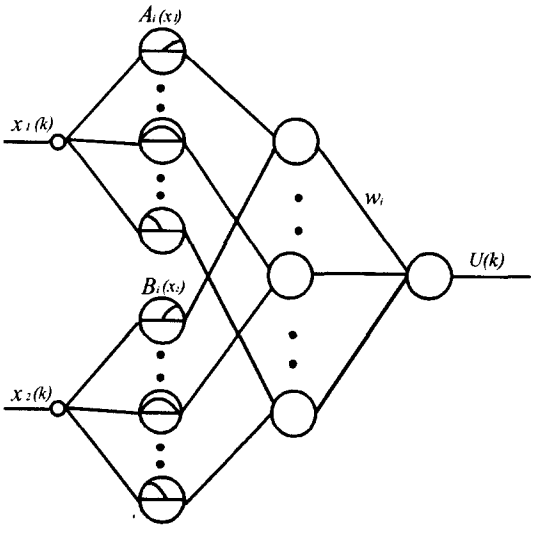


그림 5. 퍼지-신경망 제어기
Fig. 5. Fuzzy-neural network controller

이 값들은 은닉층으로 전달되며 은닉층에서는 이 두 값을 곱한 값

$$\mu_i = A_i(x_1) \times B_i(x_2) \quad (20)$$

를 출력하며 가중치 W_i 와 곱해져 출력층으로 전달한다.

$$y = \frac{\sum_{i=0}^{n} \mu_i W_i}{\sum \mu_i} \quad (21)$$

출력층은 비퍼지화 단계에 해당하므로 식 (21)와 같이 무게중심법에 의해 비퍼지화를 실행한다.

신경망 에뮬레이터가 학습된 후 제어기 출력단에서의 오차항을 구하기 위하여 플랜트 출력단에서의 오차항 $e(k)$를 구하면

$$e(k) = \frac{1}{2}(r(k) - y(k))^2 \quad (22)$$

여기서 $y(k)$는 SRM 실제 출력, $r(k)$는 SRM 기준 속도이다.

이 오차항은 오차 역전파 알고리즘에 의해 신경회로망 에뮬레이터를 통하여 역전파된다.

은닉층에서의 오차항은

$$\delta_i^m = f'(net_i^m)(\delta^n W_j^m) = (1-(out_i^m)^2)(\delta^n W_i^m) \quad (23)$$

$$\delta^n = f'(net^n)(r(k)-y(k)) = (1-(y(k))^2)(r(k)-y(k)) \quad (24)$$

이 되고 신경회로망 에뮬레이터 입력층에서 퍼지-신경망 제어기의 출력 $u(k)$와 연결된 노드에서의 오차항은 은닉층의 각 노드에서 역전파되어 온 오차항의 합이 된다.

$$\delta_j' = \sum_i \delta_i^m \quad (25)$$

여기서 구해진 오차항 δ_j'은 다시 퍼지-신경망 제어기를 학습시키기 위한 오차항으로 사용된다. 한편 퍼지-신경망 제어기에서 학습될 파라미터는 식 (18)의 a_{i1}, b_{i1}, c_{i1} 과 식 (19)의 a_{i2}, b_{i2}, c_{i2} 그리고 출력층에서의 가중치 W_i가 된다. 각 파라미터는 오차 역전파 알고리즘에 의해 최적의 제어량을 출력하도록 실시간 학습된다. 각 층에서의 학습과정은 다음과 같다.

먼저 출력층에서의 가중치 변화량은

$$\Delta W_i \propto -\frac{\partial e(k)}{\partial W_i(k)} \quad (26)$$

이 된다.

체인 룰(chain rule)에 의해 식 (30)은

$$\Delta W_i = -\frac{\partial e(k)}{\partial W_i(k)} = -\frac{\partial e(k)}{\partial u(k)} \times \frac{\partial u(k)}{\partial W_i(k)} \quad (27)$$

로 표현될 수 있고, 여기서

$$\frac{\partial u(k)}{\partial W_i(k)} = \frac{\partial}{\partial W_i(k)}\left(\frac{\sum \mu_i(k)\, W_i(k)}{\sum \mu_i(k)}\right) = \frac{\mu_i(k)}{\sum \mu_i(k)} \quad (28)$$

이 된다.

그러나 $\frac{\partial e(k)}{\partial u(k)}$는 $e(k)$를 구할 수 없으므로 계산이 불가능하다. 따라서 $\frac{\partial e(k)}{\partial u(k)}$를 계산하는 대신 이미 알고 있는 에뮬레이터를 통하여 역전파된 오차값을 이용하면,

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

$$\delta^{k+1} = \delta_j' = \sum_i \delta_i^m \quad (30)$$

이 되며 이것을 사용하면 제어기 출력층에서의 가중치 변화량은

$$\Delta W_i(k+1) = \eta\, \delta^{k+1}\left(\frac{\mu_i(k)}{\sum \mu_i(k)}\right) \quad (31)$$

이 된다. 그러므로 출력층의 가중치 $W_i(k+1)$는

$$W_i(k+1) = W_i(k) + \Delta W_i(k+1) \quad (32)$$

에 의해 조정된다.

3.3 퍼지-신경망 제어기를 이용한 속도 제어 시스템의 구성

퍼지-신경망 제어기와 신경망 에뮬레이터를 사용한 SRM 전체적인 속도 제어 시스템은 그림 6과 같다.

구성된 SRM의 속도 제어 시스템 동작 순서를 살펴보면 다음과 같다. 먼저 전체 시스템을 초기화한다. 그리고 퍼지-신경망 제어기의 출력으로서 임의의 값을 인가하여 매 샘플링 시간마다 신경망 에뮬레이터의 출력과 플랜트의 출력과의 오차를 최소화하는 방향으로 에뮬레이터를 학습시킨다. 예비학습 단계를 거치면 기준 속도 $w^*(k)$와 SRM의 실제 속도 $w(k)$와의 오차 x_1과 오차의 변화율 x_2가 퍼지-신경망 제어기 입력으로 들어가고 출력으로 전류 기준값인 $u(k)$를 내보낸다. 그리고 제어기 입력값인 x_1과 x_2가 $u(k)$와 함께 다시 에뮬레이터 입력으로 사용되고 에뮬레이터에서 계산된 출력값과 실제 전동기 속도와의 오차를 이용하여 신경망 에뮬레이터를 학습시킨다. 에뮬레이터 학습 단계가 완료되면 퍼지-신경망 제어기의 학습을 위하여 기준 속도와 SRM의 실제 속도와의 오차를 신경망 에뮬레이터를 통하여 역전파시켜 퍼지-신경망 제어기 출력단에서의 오차항을 계산한다. 그리고 마지막으로 이 오차항을 이용하여 퍼지-신경망 제어기를 학습시키게 된다.

본 논문에 쓰인 에뮬레이터는 종래의 간접형 신경회로망에서 미리 얻어진 학습패턴을 이용하여 오프라인으로 수십, 수백 회 예비 학습하는 단계가 필요없다. 단지 온라인으로 몇 개의 제어 입력 샘플을 임

의의 시간동안 제시하는 온라인 예비 학습 단계를 거친 후 신경망 에뮬레이터를 학습시킨 것과 유사한 방식으로 퍼지-신경망 제어기를 학습시킨다.

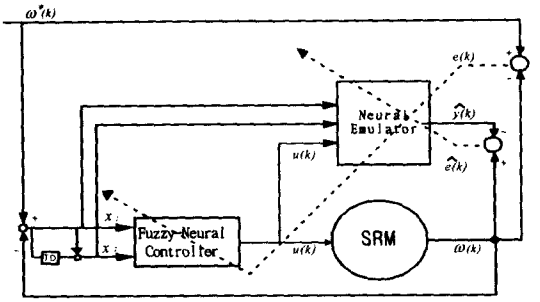


그림 6. SRM 속도 제어 시스템
Fig. 6. The speed control system of SRM

4. 시뮬레이션

본 논문에서는 비선형 인덕턴스 특성을 갖는 SRM의 시뮬레이션을 위하여 입력측 DC링크 전압의 맥동을 무시하고, SRM의 인덕턴스는 자기포화가 없다고 가정하였다. 그리고, 비대칭 브리지 컨버터의 스위치 및 다이오드의 손실은 무시하였다.

시뮬레이션은 C언어를 사용하였으며, 미분 방정식의 해를 구하기 위해 Runge-Kutta 방법을 사용하였으며 제어루프의 샘플링 시간은 2[ms]로 하였다.

그림 7은 PI제어기를 사용하였을 경우, 속도 지령 500[rpm]에서 1500[rpm]으로의 속도 응답 파형이다. 오버슈트없이 0.2초 이내에 지령속도에 도달한다.

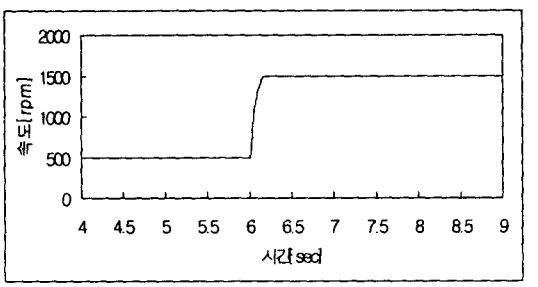


그림 7. PI 제어기를 사용했을 경우의 속도 응답(무부하)
Fig. 7. The speed response of SRM using the PI controller with no load

그림 8, 9는 퍼지-신경망 제어기를 사용하였을 때의 속도 응답 파형이다. 1회 학습 때에는 상당히 큰 오버슈트와 정상상태에 도달하는데 많은 시간이 걸리고 있지만 학습과정을 통하여 점차 빠른 응답을 나타내고 있다. 그림 12에서 4회 학습한 경우, 속도 지령치를 0.1초만에 오버슈트없이 지령 속도를 잘 추종하고 있음을 볼 수 있다.

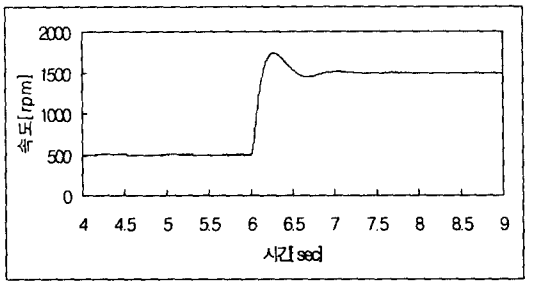


그림 8. 퍼지-신경망 제어기를 사용했을 경우의 속도 응답 (1회 학습, 무부하)
Fig. 8. The speed response of SRM using fuzzy-neural network controller (after 1 training, no load)

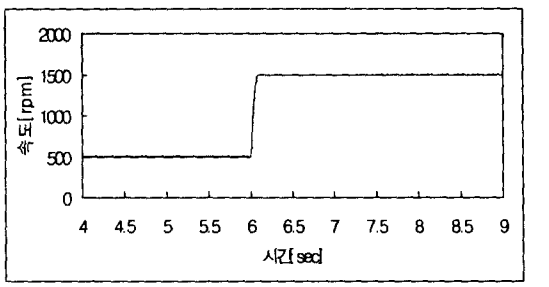


그림 9. 퍼지-신경망 제어기를 사용했을 경우의 속도 응답 (4회 학습, 무부하)
Fig. 9. The speed response of SRM using fuzzy-neural network controller (after 4 training, no load)

그림 10은 1000[rpm]에서 부하를 인가하였을 경우의 파형이다. 정정시간이 상당이 큼을 확인할 수 있다. 퍼지-신경망 제어기를 7회 학습한 후의 부하 응답이 그림 11에 나타나 있다. 학습을 통하여 부하에도 강인한 특성을 보여주고 있다.

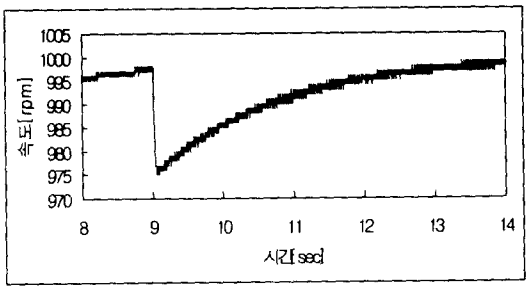


그림 10. 부하 인가시 PI 제어기를 사용했을 경우의 속도 응답
Fig. 10. The speed response of SRM using the PI controller with load

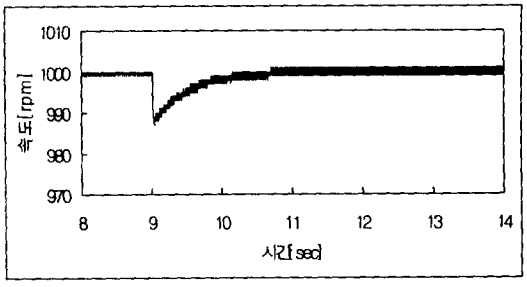


그림 11. 부하 인가시 퍼지-신경망 제어기를 사용했을 경우의 속도 응답
Fig. 11. The speed response of SRM using fuzzy-neural network controller with load

5. 실험 결과 및 고찰

그림 12는 실험에 사용된 전체 하드웨어 구성도를 나타낸다.

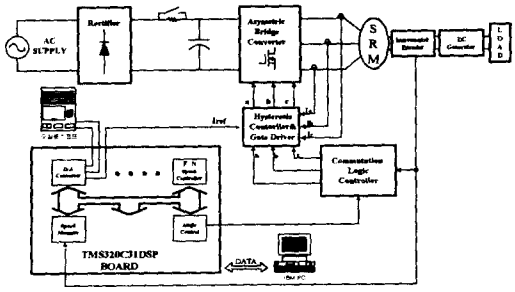
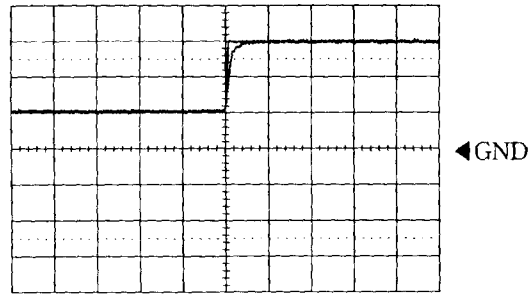


그림 12. 실험장치 구성도
Fig. 12. The configuration of experimental setup

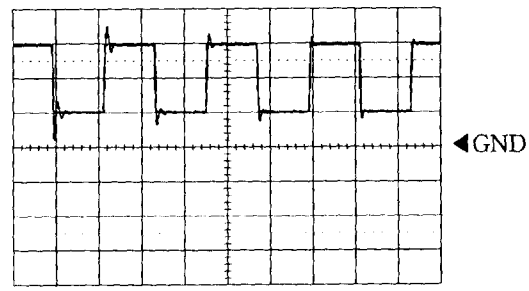


(X=0.5s/div, Y=500rpm/div)

그림 13. 무부하시 PI 제어기를 사용했을 경우의 속도 응답
Fig. 13. The speed response of SRM using the PI controller with no load

SRM의 속도 및 전류제어 시스템의 하드웨어는 고성능 DSP TMS320C31보드와 전류제어형 비대칭 브리지 컨버터로 구성하였으며 속도측정 회로, 기동 시퀀스 및 시스템 보호회로등 대부분의 제어회로를 시스템의 소형화가 가능하도록 EPLD내부에 구성하였다. 본 논문에서는 SRM의 속도제어를 위하여 실시간 학습이 가능한 퍼지-신경망 제어기를 구성하여, 기존의 PI제어기만을 사용한 속도제어 시스템과 비교하였다[14].

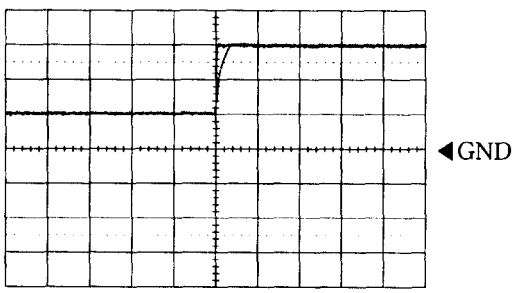
그림 13은 무부하 상태에서 기준속도를 500[rpm]에서 1500[rpm]으로 변화시켰을 때의 PI 제어기 적용시 속도응답 파형으로, 오버슈트없이 약 0.3초만에 지령속도에 도달하는 것을 볼 수 있다.



(X=5s/div, Y=500rpm/div)

그림 14. 퍼지-신경망 제어기의 학습 과정의 속도 응답
Fig. 14. The speed response of training processof fuzzy-neural network controller during 4 cycle

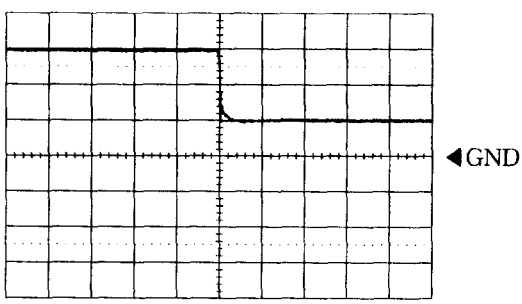
그림 14는 온라인 학습이 가능한 퍼지-신경망 제어기를 사용하였을 경우 지령속도 500[rpm]에서 1500[rpm]으로의 속도응답 파형이다. 신경망 에뮬레이터의 학습기능을 이용하여 시스템의 파라미터를 전혀 모르는 상태에서도 빠른 시간 내에 퍼지-신경망 제어기의 소속함수와 제어규칙을 찾아가고 있음을 보여준다. 그림 15는 그림 14와 같은 과정으로 7회 학습한 후의 속도 응답 파형으로, 0.2초만에 오버슈트없이 지령속도에 도달하는 것을 볼 수 있다.



(X=0.5s/div, Y=500rpm/div)

그림 15. 퍼지-신경망 제어기의 속도 응답 (7회 학습, 무부하)
Fig. 15. The speed response of fuzzy-neural network controller(after 7 training, no load)

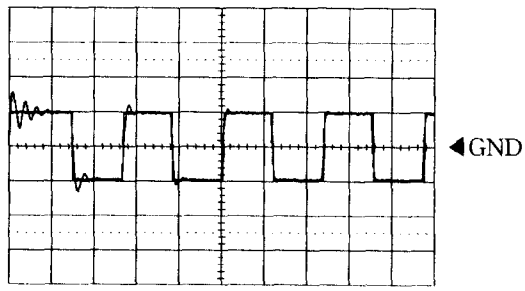
그림 16은 속도 지령치를 1500[rpm]에서 500[rpm]으로 감속하는 경우에 대한 속도 응답 파형이다.



(X=0.5s/div, Y=500rpm/div)

그림 16. 감속시 퍼지-신경망 제어기의 속도 응답
Fig. 16. The speed response of fuzzy- neural network controller during deceleration

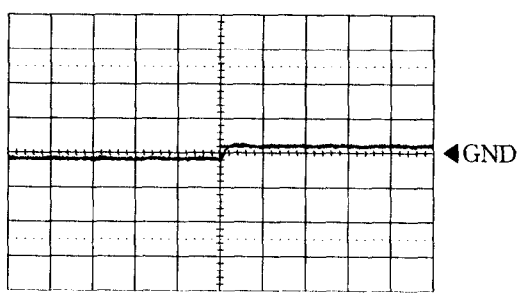
그림 17은 SRM을 500[rpm]에서 -500[rpm]으로 정역 운전하였을 경우 1회 학습과정에서부터 4회 학습과정까지의 속도 응답 파형이다. 1회 학습시에는 상당히 큰 오버슈트와 정상상태에 도달하는데 많은 시간이 걸리지만 2회, 3회 학습을 거치면서 오버슈트와 정상상태 도달 시간이 감소하고 있음을 볼 수 있다. 그림 18은 지령속도를 100[rpm]에서 -100[rpm]으로 인가했을 때 속도응답 파형이다. 저속이기 때문에 약간의 토오크 리플이 존재하지만 지령속도를 잘 추종함을 볼 수 있다.



(X=5s/div, Y=500rpm/div)

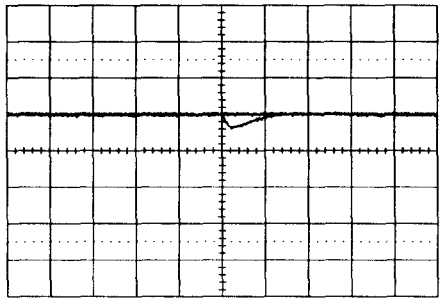
그림 17. 정 · 역 운전시 퍼지-신경망 학습과정
Fig. 17. The training process of fuzzy-neural network controller at the forward and reverse rotation

그림 19, 20은 전동기의 부하를 증가시켰을 경우 PI제어기와 퍼지-신경망 제어기에 대한 속도변동을 나타낸 파형이다. PI제어기를 사용한 경우보다 퍼지-신경망 제어기를 사용한 경우가 속도의 변동 및 정정시간이 감소함을 확인할 수 있다.



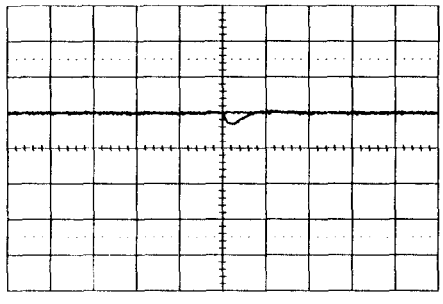
(X=0.5s/div, Y=500rpm/div)

그림 18. 저속 운전시 퍼지-신경망 제어기의 속도응답
Fig. 18. The speed response of the fuzzy -neural network controller at low speed



(X=0.5s/div, Y=1000rpm/div)

그림 19. 부하 인가시 PI 제어기 속도 응답 특성
Fig. 19. PI controller speed response characteristics with load



(X=0.5s/div, Y=1000rpm/div)

그림 20. 부하 인가시 퍼지-신경망 제어기 속도 응답 특성
Fig. 20. Fuzzy-neural network controller speed response characteristics with load

6. 결론

본 논문에서는 비선형 인덕턴스 특성을 갖는 SRM의 속도제어를 위하여 실시간 연산이 가능한 TMS320C31 32bit 마이크로프로세서를 사용하였으며 시뮬레이션과 실험을 통하여 다음과 같은 결과를 얻었다.

(1) SRM의 속도제어를 위해 일반적으로 사용되고 있는 PI제어기의 경우 최적의 제어이득을 선정하기가 어려우므로 수학적 모델링을 요구하지 않는 퍼지-신경망 제어기를 구성하여 제어성능의 향상을 확인하였다.

(2) 신경망의 학습기능을 이용하여 부하의 변동이나 외란에 실시간으로 적응할 수 있는 퍼지-신경망 제어기는 7회 정도의 학습만으로도 실제 속도가 기준 속도를 정확히 추종하였으며 외란에도 강인한 특성을 나타냄을 확인할 수 있었다. 또한 저속·고속 운전 및 정·역 운전 그리고 감속운전 등 다양한 속도제어에서도 우수한 성능을 나타내었다.

(3) 경제성 및 신뢰성 있는 구동 시스템을 구현하기 위해 위치센서로는 고가의 절대 엔코더 대신 증분형 엔코더를 사용하였으며 초기위치 확립을 위해 초기 기동 회로를 제안하여 SRM의 안정적인 구동을 가능하게 하였다.

본 연구는 산업자원부에서 시행하는 에너지 절약기술개발사업의 지원에 의해 수행되었음.

참 고 문 헌

[1] 임근희, "Switched reluctance motor의 개발 및 응용", 전기공업 9월호, pp.29~43, 1993.
[2] T. J. E Miller, "Switched Reluctance Motors and Their Control", Oxford University press, 1993.
[3] D. S. Reay, T. C. Green and B. W. Williams, "Application of Associative Memory Neural Networks to the Control of a Switched Reluctance Motor", proc. of IECON'93, pp. 200-206, 1993.
[4] S. Bolognani, M. Zigliotto, "Fuzzy Logic Control of Switched Reluctance Motor Drive", IEEE IAS Annu. Meet. pp. 2049-2054, 1993.
[5] S.K. Panda, X.M. Zhu and P. K. Dash, "Fuzzy Gain Scheduled PI Speed Controller for Switched Reluctance Motor Drive", IECON'97.
[6] Mahmood Nagrial, M. T. Qureshi, Mostafa A. Mohamed, "Fuzzy Logic Applications to High Performance Drives with Special Reference to Reluctance Motor", Proc. of PEDS'97, vol. 1, pp. 193-198, 1997.
[7] Y. dote, Y. Fujino and A. Suyitno, "Neuro-Fuzzy Robust Controllers for AC Drives System", Proc. 19th IEEE Int. Conf. on Industrial Electronics, Controls and Instrumentations, vol.3, pp.2311-2316, 1993.
[8] Philip D. Wasserman, "Advanced Methods in Neural Computing", Van Mostrand Reinhold, 1993.
[9] Texas Instrument, "Application Note DSP Solutions for Switched Reluctance Motors", July, 1997.
[10] 김대수, "신경망 이론과 응용", 1995, 하이테크 정보.

[11] Jyh-Shing R. Jang, "Self-Learning Fuzzy Controllers Based on Temporal Back Propagation", IEEE Trans. on Neural Net-works, vol. 3, No. 5, pp.714-723, Sep. 1992.
[12] Marzuki Khalid and Rubiyah Yusof, "Neuro-Control and its Application", Springer, 1996.
[13] S. Omatu, M. Khalid and R. Yusof, "Neuro-Control and its Applications", Springer-Verlag London Limited, 1996.
[14] Texas Instrument, TMS320Cx User's Guide, 1994.

◇저자소개◇

박 지 호(朴志浩)
1997년 성균관대학교 전기공학과 학사. 1999년 성균관대학교 전기전자 및 컴퓨터공학부 석사. 1999년 8월 ~ 현재, 국방과학연구소 연구원.

김 연 충(金淵忠)
1995년 성균관대학교 전기공학과 학사. 1997년 성균관대학교 전기공학과 석사. 1997년 2월 ~ 현재, 성균관대학교 전기전자 및 컴퓨터공학부 박사과정.

원 충 연(元忠淵)
1978년 성균관대학교 전기공학과 학사. 1980년 서울대학교 전기공학과 석사. 1987년 서울대학교 전기공학과 박사. 1998년 3월 ~ 현재, 성균관대학교 전기전자 및 컴퓨터공학부 교수.

김 창 림(金昌林)
1974년 인하대학교 전기공학과 학사. 1977년 ~ 현재, 주식회사 효성 전동기부 부장.

최 경 호(崔景好)
1991년 2월 영남대학교 전기공학과 학사. 1995년 영남대학교 전기공학과 석사. 1995년 ~ 현재, 주식회사 효성 전동기 설계 Part 근무.