영구자석 동기전동기 구동을 위한 신경회로망 PI 파라미터 자기 동조 시뮬레이터

배은경*, 권중동**, 전기영***, 박춘우§, 오봉환§§, 정춘병§§§, 이훈구†, 한경희*
*명지대학교, **공항철도(주), ***중소기업진흥공단, §(주)아코, §§명지전문대, §§§(주)한국종합엔지니어링, †용인송담대학

Neural network PI parameter Self-tuning Simulator for Permanent Magnet Synchronous Motor operation

Eun-Kyeong Bae*, Jung-Dong Kwon**, Kee-Young Jeon***, Choon-Woo Park§,
Bong-Hwan Oh§§, Choon-Byeong Jeong§§§, Hoon-Goo Lee†, Kyung-Hee Han*
*Myongji Univ., **A'REX, ***SBC, §A-Cho, §§Myongji College, §§§KTE, †Yong-in Songdam College

ABSTRACT

In this paper proposed to neural network PI self-tuning direct controller using Error back propagation algorithm. Proposed controller applies to speed controller and current controller. Also, this built up the interface environment to drive it simply and exactly in any kind of reference, environment fluent and parameter transaction of PMSM. Neural network PI self-tuning simulator using Visual C++ and Matlab Simulation is organized to construct this environment, Built up interface has it's own purpose that even the user who don't have the accurate knowledge of Neural network can embody operation characteristic rapidly and easily.

1. 서 론

일반적으로 영구자석 동기전동기가 적용된 시스템에서는 고정 PI 이득을 채택하고 있다. 제어기가 고정 PI이득을 가질 경우, 부하가 변동되는 제어 환경이나 시스템에 비선형성 외란이 가해지는 조건 등에 적용될 경우 최초의 이득설정만으로는 원하는 제어특성을 기대할 수 없는 단점을 갖고 있다.[1] 본 연구에서는 이러한 문제점들을 해결하기 위해서 사용 운전조건과 용이성을 고려한 신경회로망 PI 자기 동조 제어기를 적용하여 속도오차 및 전류오차를 최소화하는 안정된 제어기를 설계하는데 목적을 두었다. 또한, 종래의 제어시스템은 제어알고리즘의 개발과 알고리즘의 검증을 위한 시뮬레이션 과정이 각각 분리되어 있어서 알고리즘 단계에서 수정을 가할 때, 나머지 단계에서도 일일이 검증과 수정을 해야 하는 비효율적인 구조였다. 그러나 본 논문에서 적용한 시뮬레이터는 제어알고리즘과 검증을 위한 시뮬레이션이 자동으로 생성되기 때문에 일부분의 알고리즘 수정이나 시뮬레이션 수정으로 인한 시간과 노력이 줄어들도록 구현하였으며, 알고리즘의 타당성을 실험을 통하여 확인하였다.

2. 영구자석 동기전동기의 수학적 모델링

영구자석 동기전동기를 정확하게 표현하기 위해 수학적 모델이 필수적이다. 영구자석 동기전동기의 고정자 권선은 전기적으로 120 간격으로 배치되어 있고 3상 Y결선으로 이루어져 있으며, 회전자는 영구자석이다. 그림 1은 영구자석 동기전동기의 등가모델을 나타낸 것이다.[2]

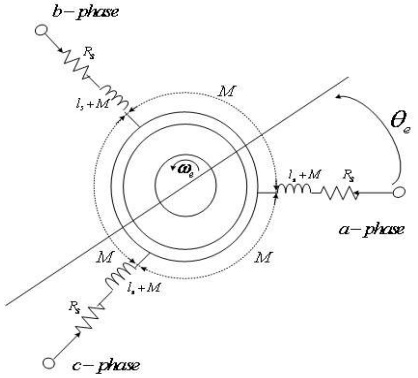


그림 1 영구자석 동기전동기의 등가모델
Fig. 1 Model of permanent magnet synchronous motor

3. 신경회로망 제어기 구조

3.1 PI제어기의 파라미터 자기 동조 알고리즘

일반적인 디지털 PI 제어기의 형태는 식 (1)과 같다.

$$u(t) = u(t-1) + K_p[e(t)-e(t-1)] + K_I e(t) T_s \quad (1)$$

여기서 Ts는 샘플링 시간을 나타내며 각 파라미터들의 조절은 제어 시스템의 e(t)에 영향을 준다. 그림 2의 오차신호는 오류역전파 알고리즘(EBPA : Error back propagation algorithm)과 최급하강법을 이용하여 신경회로망의 연결가중치를 조정으로 목적함수 E가 최소가 되도록 조절한다. 이때 목적함수 E는 식 (2)와 같고, 그림 2은 오류역전파 알고리즘을 사용한 속도제어기 및 전류제어기 학습블록선도를 나타낸다.

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

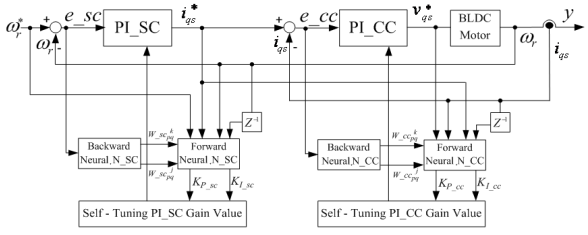


그림 2 EBPA를 이용한 자기 동조 PI제어기 학습 블록선도
Fig. 2 Block diagram of the learning scheme self-tuning PI coneroller using Error Back Propagation algorithm

본 논문에서는 신경회로망 제어기 그림 3와 같이 3개의 층으로 구성된다.

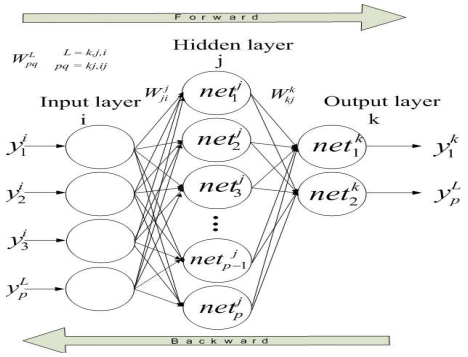


그림 3 신경회로망의 세부구조

Fig. 3 Detailed structure of Neural Networks

각 출력의 뉴런들은 연결가중치 $\triangle W^L_{pq}$의 조정으로 값이 결정된다. 입출력 뉴런들의 관계는 식 (3), (4)와 같이 나타낸다.

$$\neq t^L_P = \sum W^L_{pq} \times y^L_q \qquad (3)$$

$$y^L_p = f(\neq t^L_p) \qquad pq:kj,ji \quad L:k,j,i \qquad (4)$$

이 때 함수 $f(x)$는 활성화 함수로 단조증가, 연속, 미분가능의 성질을 만족해야 하므로 일반적으로 시그모이드 함수를 사용한다. 연결가중치의 변화에 따른 목적함수는 식 (5)와 같이 나타낸다.

$$\triangle W^L_{pq} = -\frac{\partial E}{\partial W_{pq}}\alpha \qquad (5)$$

$\partial E/\partial W^L_{pq}$를 연쇄법칙(Chain rule)에 의해 풀면 식 (6)과 같이 나타낸다.

$$\frac{\partial E}{\partial W^L_{pq}} = \frac{\partial E}{\partial y(t)}\frac{\partial y(t)}{\partial u(t-1)}\frac{\partial u(t-1)}{\partial K(t-1)}\frac{\partial K(t-1)}{\partial W^L_{pq}} \qquad (6)$$

플랜트에 대한 정보를 모른다면 미지의 플랜트에 대해 입출력의 변화량과 부호는 유도할 수 있다. 그러므로 $sign$함수로 나타내면 식 (7)과 같이 나타낸다.

$$\frac{\partial y}{\partial u(t-1)} = sign\left(\frac{\partial y}{\partial u(t-1)}\right)\left|\frac{\partial y}{\partial u(t-1)}\right| \qquad (7)$$

$K(t-1)$은 PI제어기의 파라미터이므로 $K = K_P, K_I$가 된다. 그러므로 $\partial K(t-1)/\partial W_{pq}$는 EBP에 의해 식 (8), (9)와 같이 나타낸다.

$$\left(\frac{\partial K_P}{\partial W^k_{pq}}\ \frac{\partial K_I}{\partial W^k_{pq}}\right) = \left(\frac{\partial y_{1k}}{\partial W^k_{1q}}\ \frac{\partial y_{2k}}{\partial W^k_{2q}}\right) \qquad (8)$$

$$= (F^k_1 y^j_q F^k_2 y^j_q)$$

$$\left(\frac{\partial K_P}{\partial W^j_{pq}}\ \frac{\partial K_I}{\partial W^j_{pq}}\right) = \left(\frac{\partial y_{1k}}{\partial W^j_{1q}}\ \frac{\partial y_{2k}}{\partial W^j_{2q}}\right) \qquad (9)$$

$$= \{F^j_q y^j_q (F^k_q W^k_{pq}) F^j_q y^j_q (F^k_q W^k_{pq})\}$$

$\partial u(t-1)/\partial K(t-1)$는 디지털 PI제어기인 식 (1)과 같이 되고 관계식은 다음 식 (10), (11)과 같이 나타낸다.

$$\frac{\partial u(t-1)}{\partial K(t-1)} = e(t-1) - e(t-2) \qquad K = K_P \qquad (10)$$

$$\frac{\partial u(t-1)}{\partial K(t-1)} = e(t-1) \qquad K = K_I \qquad (11)$$

그러므로 연결가중치의 변화에 따른 목적함수의 변화는 식 (7)~(11)을 식 (5)에 의해 식 (12)와 같이 나타낸다.

여기서 $F^L_V = f_V'(net_L)$ $L=k, j, i$ 층을 나타낸다.

$$\triangle W^L_{pq} = -\frac{\partial E}{\partial W_{pq}}\alpha = \alpha \times e(t)\frac{\partial y}{\partial u(t-1)}$$

$$\left(\partial u(t-1)\frac{\partial K_p(t-1)}{\partial W^L_{pq}} + \partial u(t-1)\frac{\partial K_I(t-1)}{\partial W^L_{pq}}\right) \qquad (12)$$

출력층과 중간층, 중간층과 입력층에서의 연결가중치 변화량을 나타내면 식 (13)과 같이 나타낸다.

$$\triangle W^L_{pq}(t) = W^L_{pq}(t-1) + \triangle W^L_{pq} + \beta W^L_{pq}(t-1) \qquad (13)$$

여기서, α는 학습율 β는 모멘텀 상수이다.

그림 4은 ATH(Absoluteness Tangent Hyperbolic) 활성화 함수를 적용한 전체적인 시스템 제어 블록선도를 나타낸다.

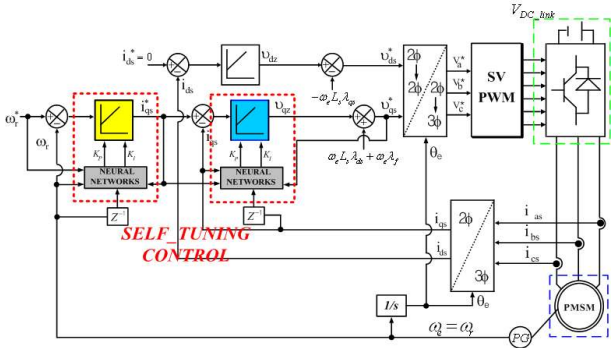


그림 4 전체시스템의 구성 및 전력회로

Fig. 4 Overall system configuration and power circuit

4. 시뮬레이터구성

개발된 시뮬레이터는 Visual C++를 기반으로 제작되었다. 영구자석 동기전동기의 파라미터에 따라 제어 및 신경회로망 파라미터를 설정할 수 있어 다양한 경우의 시뮬레이션 결과를 쉽게 얻을 수 있다. 지령치를 주어 특성을 확인할 수 있고, 신경회로망이 아닌 고정된 파라미터를 가진 제어기 결과도 얻을 수 있어 신경회로망제어기의 우수성도 확인할 수 있다. 그림 5는 영구자석 동기전동기의 구동시뮬레이터를 나타낸다.

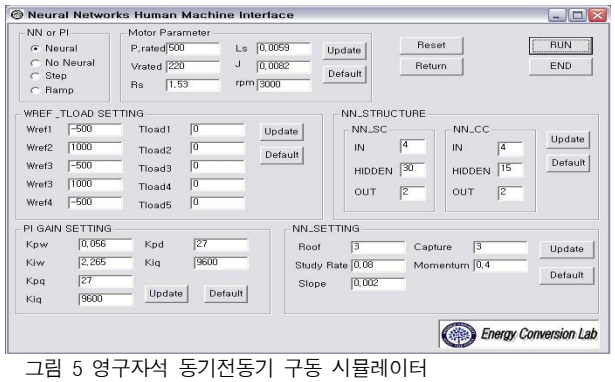


그림 5 영구자석 동기전동기 구동 시뮬레이터

Fig. 5 Permanent magnet synchronous motor operation simulator

제안된 기법을 검증하기 위해 속도제어기 신경회로망은 4개의 입력층과 30개의 은닉층, 2개의 출력층을 가지며 전류제어기 신경회로망은 4개의 입력층과 15개의 은닉층, 2개의 출력층을

갖는 다층퍼셉트론으로 구성하였다. 이때 학습률은 각각 0.08 모멘텀은 0.4이다. 표 1은 시뮬레이션 및 실험에 사용한 영구자석 동기전동기의 정격과 파라미터를 나타낸다.

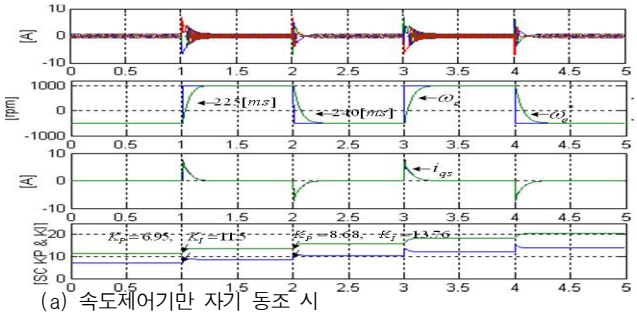
표 1 영구자석 동기전동기의 파라미터

Table 1 Parameters of Permanent magnet synchronous motor

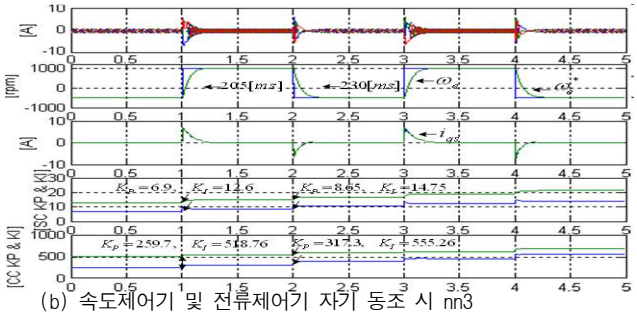
Rated power	500[W]	Rs	1.53[Ω]
Rated voltage	220[V]	Ls	0.0059[H]
Rated speed	3000[rpm]	J	0.0082[kgm^2]

4. 시뮬레이션 및 실험결과

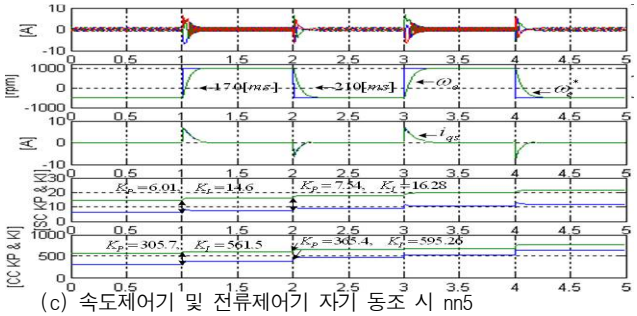
그림 6은 과 그림 7은 신경회로망 자기 동조 제어기가 속도제어기만 적용한 경우와 속도제어기와 전류제어기를 모두 적용한 경우 1[s]마다 속도제령치 −500→1000[rpm]으로 정역 운전할 때, 상전류파형, 속도응답파형, 토크성분전류, PI파라미터를 나타낸 것이다. 1[s]에서 속도제어기만 제어했을 (a) 225[ms]경우보다 속도제어기와 전류제어기 모두 제어했을 (b) 205[ms], (c) 170[ms]의 경우 속도응답특성이 향상됨을 확인할 수 있다.



(a) 속도제어기만 자기 동조 시



(b) 속도제어기 및 전류제어기 자기 동조 시 nn3



(c) 속도제어기 및 전류제어기 자기 동조 시 nn5

그림 6 정역 스텝응답 속도 & 토크성분전류 시뮬레이션 파형

Fig. 6 Step response speed & torque current simulation wave form

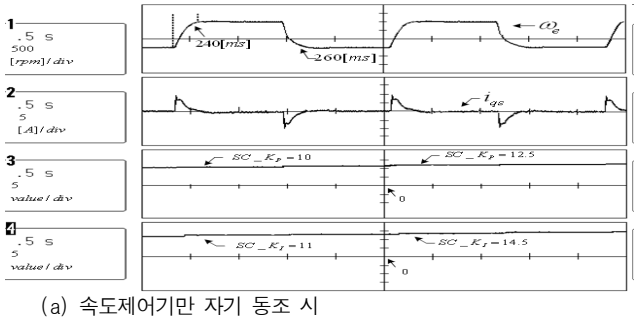
실험 결과파형 그림 7은 −500→1000[rpm]까지의 정역운전 시 파형을 나타낸다. (a)의 240[ms]의 응답속도보다 (b),(c)의 응답시간에 양호함을 확인할 수 있다.

5. 결 론

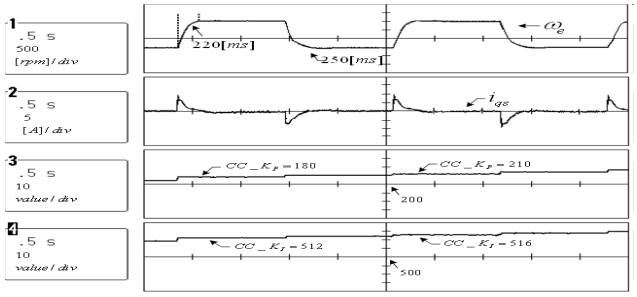
본 논문에서는 영구자석 동기전동기의 정역운전 시 PI제어기 파라미터를 실시간 자기 동조하는 신경회로망 PI제어기를 구현하여 다음과 같은 결론을 얻었다.

1) 영구자석 동기전동기의 정역운전 시 신경회로망 자기 동조 제어기가 속도제어기만 제어했을 경우보다 속도제어기와 전류제어기를 모두 제어했을 경우가 속응성이 향상됨을 시뮬레이션 및 실험을 통하여 확인하였다.

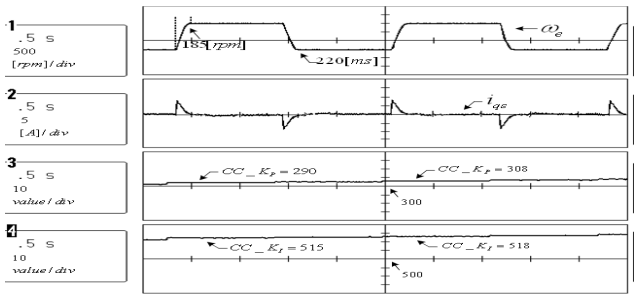
2) 신경회로망을 이용하여 각 제어기의 파라미터를 정함으로써 시스템의 응답특성 향상 및 안정화를 가져 왔으며 파라미터를 자기 동조함으로써 시뮬레이션 시간을 단축시킬 수 있었다.



(a) 속도제어기만 자기 동조 시



(b) 속도제어기 및 전류제어기 자기 동조 시 nn3



(c) 속도제어기 및 전류제어기 자기 동조 시nn5

그림 7 정역 스텝응답 속도 & 토크성분전류 시뮬레이션 파형

Fig. 7 Step response speed & torque current simulation wave form

참 고 문 헌

[1] 권중동, 전기영, 김은기, 이승환, 오봉환, 이훈구, 서영수, 한경희 "신경회로망 제어기를 이용한 PID 파라미터 추정에 관한 연구", 전력전자학회 학술지, 2005. 7

[2] 서울대학교 기초전력공학 공동연구소 하계강좌, "전동기 구동 시스템의 설계 및 실습

[3] 오원현, 강호진, 김종선, 신은철, 유지윤, 김성환, 박태식, "MATLAB Simulink를 이용한 유도전동기 구동 시뮬레이터 개발", 전력전자학술대회 논문집, pp. 217 - 221, 2002

[4] 박철효, 노민식, 박승엽,, "PMSM의 벡터 제어를 위한 디지털 시뮬레이터의 구현", 전력전자학술대회 논문집, pp. 265-268, 2004

[5] 설승기, "전기기기 제어론", 브레인 코리아, 서울대학교 전력전자 연구실, 2002