

BLDC 모터 구동을 위한 신경회로망 PI파라미터 자기 동조 시뮬레이터

배은경*, 권중동**, 김태우*, 김대균***, 전지용§, 이승환§§, 이훈구§§§, 김용주†, 한경희*

*명지대학교, **FR-TECH, ***경기인력개발원, §아주자동차대학, §§대덕대학, §§§용인송담대학, †동양공업전문대학

Neural Network PI Parameters Self-tuning Simulator for BLDC Motor operation

E. K. Bae*, T. W. Kim*, J. D. Kwon**, D. K. Kim***, J. Y. Chun§, S. H. Lee§§, H. G. Lee§§§, Y. J. Kim†, K. H. Han*

*Myongji Univ., **FR-TECH, ***GHRDI, §Ajou motor Col., §§Daedok Col., §§§Yong-in Songdam Col., †Dongyang Technical Col.

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 BLDC motor. Neural network PI self-tuning simulator using Visual C++ and Matlab Simulation is organized to construct this environment. Built-u-p 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

최근 BLDC 모터는 높은 안정성, 넓은 속도 범위에 대한 고효율 유지, 제어기 설계 용이, 가·변속 운전 등의 장점으로 각 산업분야에 수요가 점차 증가하는 추세이다. 일반적으로 BLDC 모터가 적용된 시스템에서는 고정 PI이득을 채택하고 있다. 제어가 고정 PI 이득을 가질 경우, 부하가 변동되는 제어 환경이나 시스템에 비선형성 외란이 가해지는 조건 등에 적용될 경우 최초의 이득 설정만으로는 원하는 제어특성을 기대할 수 없는 단점을 가지고 있다.^[1] 본 연구에서는 이러한 문제점들을 해결하기 위하여 사용 운전조건과 용이성을 고려한 신경회로망 PI 자기 동조 제어기를 적용하여 속도오차 및 전류오차를 최소화함과 동시에 오버슈트가 저감되는 안정된 제어를 설계하는데 목적을 두었다.

또한, 설계한 제어시스템에 소요되는 시간을 줄이고, BLDC 모터의 파라미터 변동에도 간단하고 정확하게 운전할 수 있도록 제어시스템 설계, 시뮬레이션 구현까지의 전 과정을 최소한의 프로그램 코딩 및 자동 실행되는 실시간 시뮬레이터를 개발하고자 한다. 종래의 제어시스템은 제어알고리즘의 개발과 알고리즘의 검증에 위한 시뮬레이션 과정이 각각 분리되어 있어서 알고리즘 단계에서 수정을 가할 때, 나머지 단계에서도 일일이 검증과 수정을 해야 하는 비효율적인 구조였다. 그러나 본 논문에서 개발한 시뮬레이터는 제어알고리즘과 검증을 위한 시뮬레이션이 자동으로 생성되기 때문에 일부분의 알고리즘 수정이나 시뮬레이션 수정으로 인한 시간과 노력이 줄어들도록 Visual C++과 Matlab Simulink를 통해 자동실행 되도록 구현하였다.

2. 신경회로망 제어기 구조

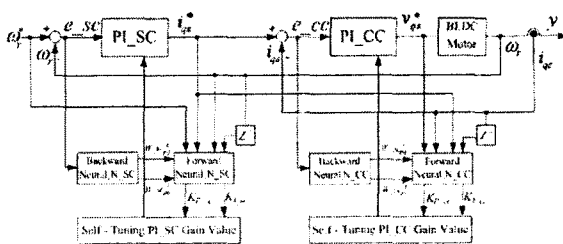
2.1 PI제어기의 파라미터 동조 알고리즘

일반적인 디지털 PI제어기의 일반 형태는 식(1)과 같다.

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

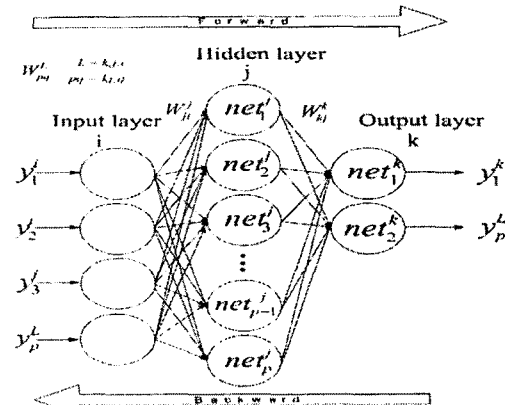
여기서 T_s 는 샘플링 시간을 나타내며 각 파라미터들의 조절은 제어 시스템의 $e(t)$ 에 영향을 준다. 그림1의 오차신호는 오류역전파 알고리즘(EBPA : Error back propagation algorithm)과 최급하강법을 이용하여 신경회로망의 연결가중치를 조정으로 목적함수 E 가 최소가 되도록 조절한다. 이 때 목적함수 E 는 식(2)와 같고, 그림 1은 오류역전파 알고리즘을 사용한 속도제어기 및 전류제어기 학습 블록선도를 나타낸다.

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



〈그림 1〉 EBPA를 이용한 자기 동조 PI제어기 학습 블록선도

본 논문에서는 신경회로망 제어기를 그림 2와 같이 3개의 층으로 구성된다.



〈그림 2〉 신경회로망 제어기의 상세 구조도

각 출력의 뉴런들은 연결가중치 ΔW_{pq}^L 의 조정으로 값이 결정된다. 일출력 뉴런들의 관계는 식(3), (4)와 같이 나타낸다.

$$\neq t_p^L = \sum W_{pq}^L x_q^L \quad (3)$$

$$y_p^L = f(\neq t_p^L) \quad pq: kj, ji \quad L: k, j, i \quad (4)$$

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.

$$\Delta W_{pq}^L = - \frac{\partial E}{\partial W_{pq}^L} \alpha \quad (5)$$

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

$$\frac{\partial E}{\partial W_{pq}^L} = \frac{\partial E}{\partial y(t)} \frac{\partial y(t)}{\partial u(t-1)} \frac{\partial u(t-1)}{\partial K(t-1)} \frac{\partial K(t-1)}{\partial W_{pq}^L} \quad (6)$$

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

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

$K(t-1)$ 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.

$$\left(\frac{\partial K_p}{\partial W_{pq}^k} \frac{\partial K_I}{\partial W_{pq}^k} \right) = \left(\frac{\partial y_{1k}}{\partial W_{1a}^k} \frac{\partial y_{2k}}{\partial W_{2a}^k} \right) \quad (8)$$

$$= (F_{1a}^k y_{1a}^j F_{2a}^k y_{2a}^j)$$

$$\left(\frac{\partial K_p}{\partial W_{pq}^j} \frac{\partial K_I}{\partial W_{pq}^j} \right) = \left(\frac{\partial y_{1k}}{\partial W_{1a}^j} \frac{\partial y_{2k}}{\partial W_{2a}^j} \right) \quad (9)$$

$$= \{ F_{1a}^j y_{1a}^i (F_{1a}^k W_{pq}^k) F_{2a}^j y_{2a}^i (F_{2a}^k W_{pq}^k) \}$$

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

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

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

그러므로 연결가중치의 변화에 따른 목적함수의 변화는 식(7), (8), (9)와 식(10), (11)을 식(5)에 의해 식(12)와 같이 나타낸다.

여기서 $F_i^L = f'(net_i)$ $L=k, j, i$ 층을 나타낸다.

$$\Delta W_{pq}^L = -\frac{\partial E}{\partial W_{pq}^L} \alpha = \alpha \times e(t) \frac{\partial y}{\partial u(t-1)} \quad (12)$$

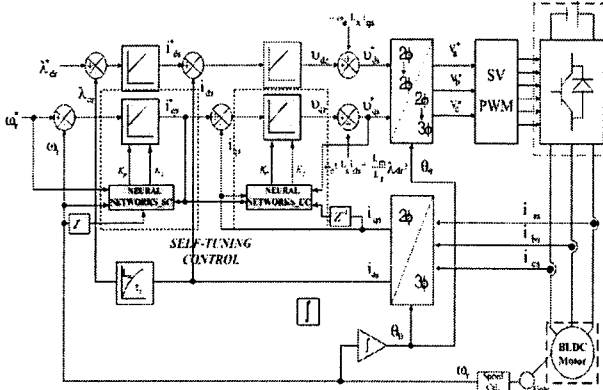
$$\left(\frac{\partial u(t-1)}{\partial W_{pq}^L} \frac{\partial K_b(t-1)}{\partial W_{pq}^L} + \frac{\partial u(t-1)}{\partial W_{pq}^L} \frac{\partial K_f(t-1)}{\partial W_{pq}^L} \right)$$

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

$$\Delta W_{pq}^L(t) = W_{pq}^L(t-1) + \Delta W_{pq}^L + \beta W_{pq}^L(t-1) \quad (13)$$

여기서 α 는 학습률 β 는 모멘텀 상수이다.

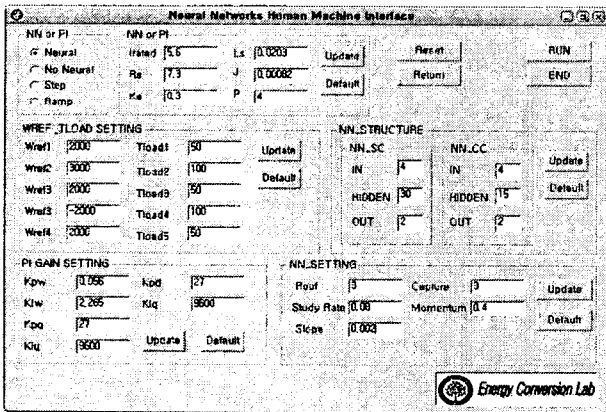
그림 3은 ATH(Absoluteness Tangent Hyperbolic) 활성화 함수를 적용 한 전체적인 시스템 제어 블록선도를 나타낸다.



〈그림 3〉 시스템 제어 블록선도

3. 시뮬레이터 구성 및 시뮬레이션 결과

개발된 시뮬레이터는 Visual C++를 기반으로 제작되었다. BLDC 모터의 파라미터에 따라 제어파라미터 및 신경회로망 파라미터를 설정할 수 있어 사용자가 다양한 경우의 시뮬레이션 결과를 쉽게 얻을 수 있다. 지령치 등을 총 5단계로 나누어 STEP, RAMP 지령을 주어 특성을 확인 할 수 있다. 또한 신경회로망이 아닌 기존 고정된 파라미터를 가진 제어기의 사용 결과도 얻을 수 있어 신경회로망 제어기의 우수성도 확인 할 수 있다. 그림 4는 BLDC 구동 시뮬레이터를 나타낸다.



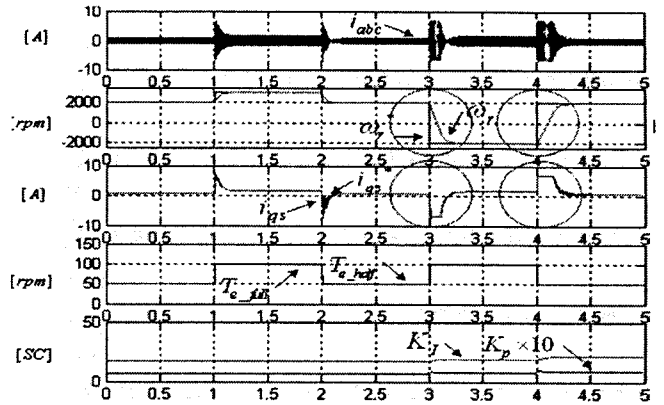
〈그림 4〉 BLDC 구동 시뮬레이터

제안된 기법을 검증하기 위해 속도제어기 신경회로망은 4개의 입력층과 30개의 은닉층, 2개의 출력층을 가지며 전류제어기 신경회로망은 4개의 입력층과 15개의 은닉층, 2개의 출력층을 갖는 다층 퍼셉트론으로 구성하였다. 이 때 학습률은 각각 0.08 모멘텀은 0.4이다. 표 1은 시뮬레이션에 사용한 BLDC 모터의 정격과 파라미터를 나타낸다.

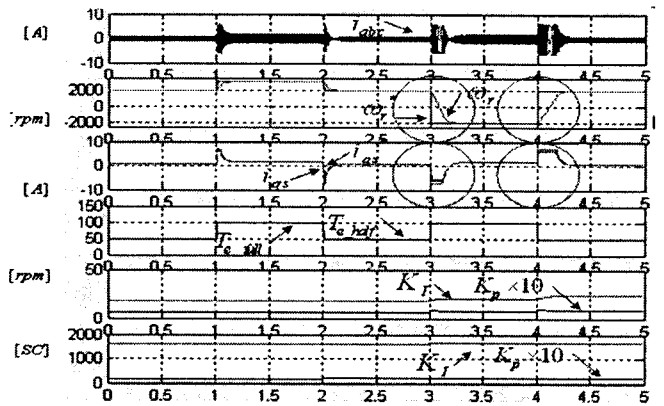
〈표 1〉 BLDC 전동기의 파라미터

정격 전압 [V]	310 [v]	인덕턴스 [L _s]	20.3 [mH]
정격 토크 [T _e]	2 [Nm]	역기전력 상수 [k _e]	0.3 [v/(rad/sec)]
저항 [R _s]	7.3 [Ω]	극 [P]	4

그림 5와 그림 6은 속도제어기만 적용한 경우와 속도제어기와 전류제어기를 모두 적용한 경우이다. 1[s]마다 속도 지령치를 2000→3000→2000→-2000→2000[rpm]으로 스텝변환하고 반부하→전부하→반부하→전부하로 가변 하는 경우 상전류 파형, 속도 응답파형, 토크 성분 전류, 부하토크 및 PI파라미터를 나타낸다.



〈그림 5〉 정역 스텝 응답 속도 및 토크 전류 파형(NN_SC)



〈그림 6〉 정역 스텝 응답 속도 및 토크 전류 파형(INN_SC & NN_CC)

3[s], 4[s]시 속도가변과 동시에 부하가변을 주었는데, 속도제어기만 적용했을 경우는 응답시간이 각각 524[ms], 536[ms]였지만, 속도제어기와 전류제어기를 모두 제어했을 경우는 응답시간이 각각 489[ms], 439[ms]로 응답특성이 향상됨을 확인할 수 있었다.

4. 결 론

본 논문에서는 BLDC 모터의 속도가변 및 부하변동 시 PI제어기 파라미터를 실시간 자기 동조 하는 신경회로망 PI제어기의 시뮬레이터를 사용하여 다음과 같은 결론을 얻었다.

- 1) BLDC 모터의 속도가변 및 부하변동 시 속도 제어기만 제어했을 경우보다 속도제어기와 전류제어기 모두 제어했을 경우가 속응성이 각각 6%, 20% 정도 향상됨을 확인하였다.
- 2) 각 제어기들의 파라미터를 신경회로망을 사용하여 시스템의 응답특성 향상 및 안정화를 가져 왔으며, 특히 파라미터를 자기 동조하여 시뮬레이션 시간을 단축시킬 수 있었다.
- 3) 시뮬레이터 제작으로 알고리즘에 대한 기초지식만으로도 쉽게 시뮬레이션을 가능하게 할 수 있었다. 또한 모터 파라미터 및 제어기 파라미터를 변동 시 시뮬레이션의 번거로움을 감소시킬 수 있었다.

〈참 고 문 헌〉

- [1] 권중동, 전기영, 김은기, 이승환, 오봉환, 이훈구, 서영수, 한경희 "신경회로망 제어기를 이용한 PID 파라미터 추정"에 관한 연구, 전력전자학회 학술지, 2005. 7
- [2] 오원현, 강호진, 김중선, 신은철, 유지운, 김성환, 박태식, "MATLAB Simulink를 이용한 유도전동기 구동 시뮬레이터 개발", 전력전자학회대회 논문집, pp. 217 - 221, 2002
- [3] 박철호, 노민석, 박승엽, "PMSM의 벡터 제어를 위한 디지털 시뮬레이터의 구현", 전력전자학회대회 논문집, pp. 265-268, 2004
- [4] 설승기, "전기기기 제어론", 브레인 코리아, 서울대학교 전력전자 연구실, 2002
- [5] 김태성, 안성찬, 현동석 "BLDC 전동기의 토크리플 저감을 위한 새로운 전류제어 알고리즘에 대한 연구", 전력전자학회대회 논문집, pp. 25-29, 2001