

신경회로망 PI자기동조를 이용한 BLDC 모터제어

배은경* · 권중동* · 전기영** · 함년근[§] · 이승환^{§§} · 이훈구† · 정춘병†† · 한경희*
 명지대학교* · 중소기업진흥공단** · (주)인텍 에프에이[§] · 대덕대학^{§§} · 용인송담대학† · (주)한국종합기계엔지니어링††

BLDC Motor Control using Neural Network PI Self tuning

E.K. Bae* · J.D. Kwon* · K.Y. Jeon** · N.G. Hahm[§] · S.H. Lee^{§§} · H.G. Lee† · C.B. Chung†† · K.H. Han*
 Myongji Univ.* · SBC** · in Tech-FA[§] · Daedok Col.^{§§} · Yong-In Songdam Col.† · KTME††

Abstract - The conventional self-tuning methods have the speed control problem of nonlinear BLDC motor which can't adapt against any kinds of noise or operation circumstances. In this paper, supposed to solve these problem to PI parameters controller algorithm using ANN. In the proposed algorithm, the parameters of the controller were adjusted to reduce by on-line system the error of the speed of BLDC motor. In this process, EBPA NN was constituted to an output error value of a BLDC motor and conspired an input and output. The performance of the self-tuning controller is compared with that of the PI controller tuned by conventional method(Z&N). The effectiveness of the proposed control method is verified thought the Matlab Simulink.

1. 서 론

BLDC 모터(Brushless DC Motor)는 유도기에 비하여 전원의 안정성과 넓은 속도 범위에 대한 높은 효율을 유지하며, 수명이 길고 제어기 설계가 용이하여 가변속 운전이 요구되는 응용개소에 최적의 전동기로 각광받고 있다.^{[1][2]} 그러나 BLDC 모터의 PID 제어기를 적용하는데 있어서의 문제점은 제어기의 파라미터를 최적화하는 것이 어렵다.^{[3][4]} 또한, 일반적으로 제어기 파라미터가 고정인 경우 일정한 부하운전시에는 양호한 제어특성을 보이지만, 부하가 변동되거나 제어환경이나 시스템에 외란 및 잡음이 가해지면 원하는 제어를 수행 할 수 없다.^{[5][6]}

본 논문에서는 속도제어기의 PI 게인값을 추정하기 위하여 신경회로망 오류역전파(EBP : Error Back Propagation)알고리즘을 적용한 BLDC 모터를 제어하는 방법에 관하여 기술하였다. 신경회로망에 의한 오류역전파 알고리즘은 부하변동시에도 속도 제어기 PI 게인값을 자기동조하므로써 외란 및 잡음에도 시스템이 강인함을 확인하였다.

2. 신경회로망 제어기 구조

2.1 PI제어기의 파라미터 동조 알고리즘

일반적인 속도형 PI제어기의 일반형태는 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 제어시스템의 $e(t)$ 에 영향을 준다. 그림1의 오차신호는 EBP학습알고리즘과 최급하강법을 이용하여 신경회로망의 연결가중치를 조정으로 목적함수 E 가 최소가 되도록 조절한다. 이 때 목적함수 E 는 식(2)와 같이 나타낸다.

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

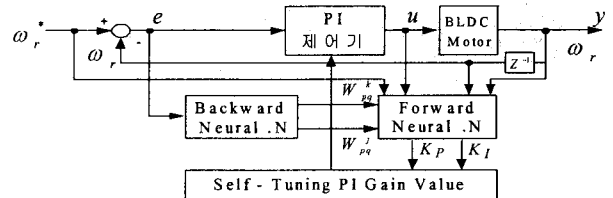


그림 1. EBP알고리즘을 이용한 자기동조 PI제어기 학습 블록선도

Fig. 1. Block diagram of the learning scheme self-tuning PI controller using Error Back-Propagation

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

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

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

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

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

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

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

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

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

$K(t-1)$ 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.

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

$$= (F_{1q}^k y_q^k F_{2q}^k y_q^k) \quad (9)$$

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

$$= (F_{1q}^k y_q^k (F_{1q}^k W_{pq}^k) F_{2q}^k y_q^k (F_{2q}^k W_{pq}^k))$$

$\partial u(t-1) / \partial K(t-1)$ 는 디지털 PI제어기 중 속도형 PI 제어기인 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 $F_v^L = f_v(\text{net}_L)$, $L=k, j, i$ 층을 나타낸다.

$$\Delta W_{jk} = -\frac{\partial E}{\partial W_{jk}} \alpha = \alpha \times e(t) \frac{\partial y}{\partial u(t-1)}$$

$$\left(\frac{\partial u_1(t-1)}{\partial W_{jk}} \frac{\partial K_p(t-1)}{\partial W_{jk}} + \frac{\partial u_2(t-1)}{\partial W_{jk}} \frac{\partial K_I(t-1)}{\partial W_{jk}} \right) \quad (12)$$

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

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

여기서 α 는 학습율 β 는 모멘텀 상수이다.

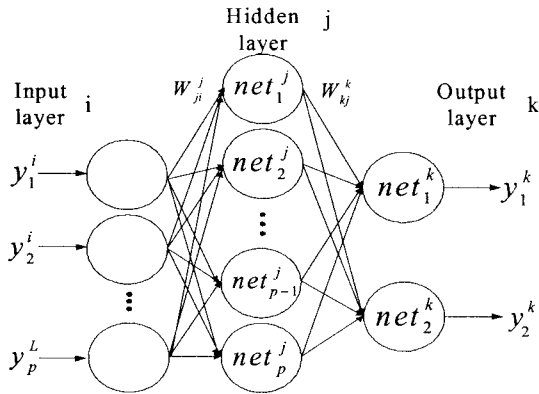


그림 2. 신경회로망의 세부구조
Fig. 2. Detailed structure of Neural Network

3. BLDC 모터 속도제어 적용

BLDC 모터 방정식을 정리하면 식(14)과 같은 $d-q$ 방정식으로 나타낼 수 있다. 여기서 $e_{\alpha} = \omega_r \lambda_m$ 이다.

$$\frac{d}{dt} \begin{bmatrix} i_{d^*} \\ i_{q^*} \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_s} & \omega_r \\ -\omega_r & -\frac{R_s}{L_s} \end{bmatrix} \begin{bmatrix} i_{d^*} \\ i_{q^*} \end{bmatrix} + \frac{1}{L_s} \begin{bmatrix} v_{d^*} \\ v_{q^*} \end{bmatrix} - \frac{1}{L_s} \begin{bmatrix} 0 \\ e_{\alpha} \end{bmatrix} \quad (14)$$

식(14)은 $d-q$ 축 전기자 전압인 v_{d^*} 와 v_{q^*} 로 $d-q$ 축 전기자전류인 i_{d^*} 와 i_{q^*} 를 제어할 수 있음을 나타낸다. 토오크식을 나타내면 식(15)와 같다.

$$T_e = -\frac{d}{dt} \lambda_m [i_{q^*} \cos \theta - i_{d^*} \sin \theta] = -\frac{d}{dt} \lambda_m i_{q^*} \quad (15)$$

운동방정식은 점성 마찰계수를 무시한 모터 출력축의 회전각속도 w_r 은 극수를 P 로 하면 w_r/P 이므로 식(16)으로 표현될 수 있다.

$$\frac{d}{dt} J \left(\frac{w_r}{P} \right) = \frac{d}{dt} J w_m = T_e - T_L \quad (16)$$

여기서, T_e 는 모터의 부하토크, J 는 모터와 부하의 관성모멘트 ($J_L + J_M$)이다.

그림 3은 앞의 관계식들로부터 구할 수 있는 블록선도이다.

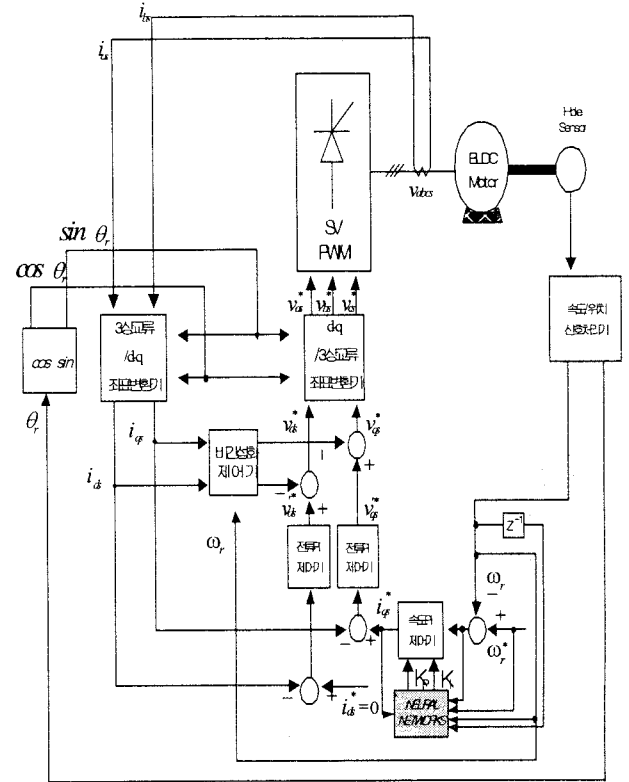


그림 3. 전체 제어기의 구성
Fig. 3. Configuration of overall controller

4. 시뮬레이션 결과

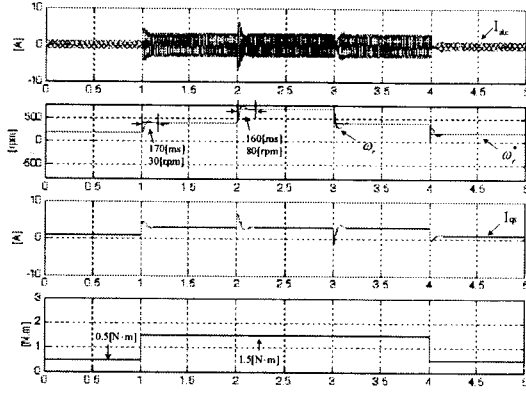
제안된 기법과 기존의 기법과의 성능 비교 검증을 위하여 신경회로망은 4개의 입력층과 20개의 은닉층, 2개의 출력층을 갖는 다층 퍼셉트론으로 구성하였다. 이때 학습율은 0.1 모멘텀은 0.4이다. 표 1은 시뮬레이션에 사용한 BLDC 모터의 정격과 파라미터를 나타낸다.

표 1. BLDC 모터의 파라미터
Table 1. Parameter of BLDC motor

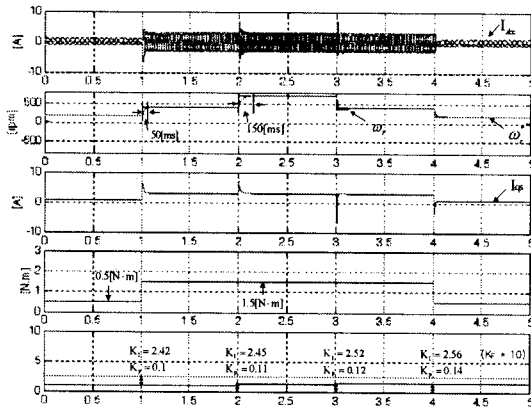
T_{α} (최대연속토크)	3.4[N·m]
K_T (토크상수)	1.13[N·m/A]
L_d (상간인덕턴스)	18[mH]
R_d (상간저항)	6.4[Ω]
J (관성)	0.32[kg·m ² × 10 ⁻³]
I_{α} (최대연속정격전류)	3.4[A]

그림 4는 1[s]마다 속도 지령치를 200->400->700->400->200[rpm]으로 스텝변환 하고 1[s]와 4[s]에서는 부하토크를 0.5->1.5->0.5[N·m]로 가변 하는 경우 기존 PI 제어기 적용 시 (a)와 신경회로망 PI제어기 적용시 (b)의 상전류 파형, 속도 응답, 토크성분 전류, 부하토크 및 PI 게인값을 나타낸다.

기존 PI제어기를 사용한 경우 (30[rpm],170[ms]), (80[rpm],160[ms]), (100[rpm],150[ms]), (35[rpm],140[ms])의 오버슈트와 응답시간을 갖지만 신경회로망 PI제어기를 사용한 경우에는 기존 PI제어기 비하여 응답시간이 50[ms], 150[ms], 100[ms], 50[ms]로 향상됨을 확인할 수 있다. 특히 오버슈트는 완전 제거되어 시스템의 안정성을 갖는다.



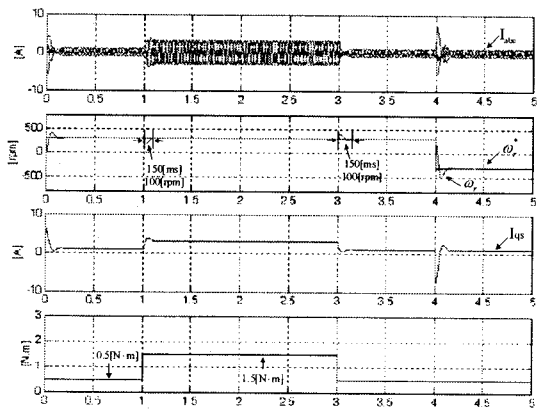
(a)The Conventional PI Controller



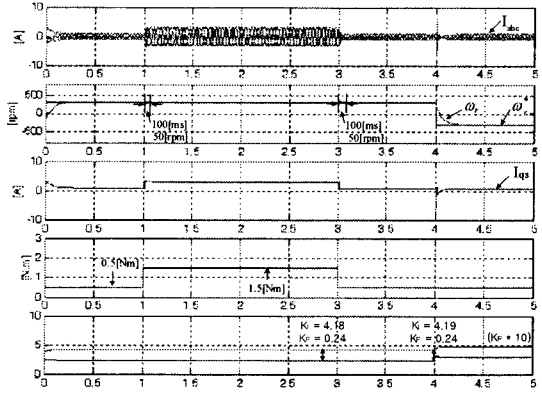
(b)NN Self tuning PI Controller After 5 training
 그림 4. 스텝응답 속도 및 토크파형
 Fig. 4. Speed & Torque Waveform of the Step response

그림 5는 1[s]와 3[s]에서 부하토크를 0.5->1.5->0.5 [N·m]로 인가하고 이때 속도 지령치를 4[s]에서 300->-300[rpm]으로 가변 하는 경우 기존 PI 제어기 적용시 (a)와 신경회로망 PI제어기 적용시 (b)의 상전류 파형, 속도 응답, 토크성분 전류, 부하토크 및 PI 게인값을 나타낸다.

기존 PI 제어기를 사용한 경우 (100[rpm],150[ms]), (100[rpm],150[ms]), (180[rpm],200[ms])의 오버슈트와 응답시간 갖지만, 신경회로망 PI제어기를 사용한 경우에는 부하인가시 50[rpm], 50[rpm]의 오버슈트로 양호해 졌으며 속도가변시에는 오버슈트가 완전제거 되었다. 응답시간도 100[ms], 100[ms], 100[ms]로 기존 PI 제어기보다 신경회로망 PI제어기가 강인함을 확인하였다.



(a)The Conventional PI Controller



(b)NN Self tuning PI Controller After 5 training

그림 5. 정역 스텝 응답 속도 및 토크 파형
 Fig. 5. Speed & Torque Waveform of the F & B Step response

5. 결론

본 논문에서는 BLDC 모터의 부하변동 및 속도가변 시 속도형 PI제어기 파라미터를 실시간 자기동조하는 신경회로망 PI제어기를 사용하여 다음과 같은 결론을 얻었다.

- 1) BLDC 모터의 속도가변시 및 정역운전시의 속응성이 기존 PI제어기보다 신경회로망 PI제어기가 각각 60[%], 58[%] 정도 향상됨을 확인하였다.
- 2) BLDC 모터의 오버슈트가 기존 PI제어기보다 신경회로망 PI제어기가 속도가변시에는 제거되고, 정역운전시에는 40[%]정도 출력특성이 양호함을 확인하였다.

[참고 문헌]

- [1] P. Pillay, R. Krishnan., "Application Characteristics of Permanent Magnet Synchronous and Brushless DC Motor for Servo Drives", Conf. Rec. IEEE/IAS Annual Meeting, pp. 30-390, 1987
- [2] 여형기, 김태형, 박정배, 이광운, 유지윤, "BLDC 전동기용 센서리스 드라이브 토크 리플 저감을 위한 전류방법", 전력전자 학술대회 논문집, pp 337-341, 1998
- [3] Toshio Fukuda, "Theory and Applications of Neural Network for Industrial Control System", IEEE Trans. Vol. 39, No. 6, Dec., 1992
- [4] C.A. Smith and A.B. Corripip, "Principle and Practice of Automatic process Control", Hohn Wiely and Sons, 1985
- [5] "30 years of adaptive neural networks: perceptron, Madaline, and backpropagation", Widrow, B.; Proceedings of the IEEE Volume 78, Issue 9, Sept. 1990 Page(s):1415 - 1442
- [6] K.J, Astrom, T.Hagglund, C.C. hang and W.K. Ho, "Automatic Tuning and Adaption for PID Controllers - A Survey", Control Eng. Practice, Vol. 1, No.4, pp699-714, 1993