

그래픽 언어를 이용한 BLDC모터 digital PI 제어에 관한 연구

*이 동 현, 홍 선 기
호서 대학교 서보기기 및 제어연구실

A study of BLDC motor digital PI control using a graphic language tool

*Dong-Hyun Lee, Sun-Ki Hong

Department of Information & Control Eng Hose University

Abstract - It is very difficult to control a motor and apply a mathematical algorism using regular micro controller. However we can embody a program using the graphic language faster and easier than using another language and conveniently compose the plate based on a PC through the DAQ board included AD and DA converter. In addition, The controllible system is made due to GUI program. I verified appropriateness PI control and determining a speed of BLDC motor which is nomally used through this program.

1. 서 론

보통의 MICRO CONTROLLER를 이용한 모터 제어는 구현하기 상당히 까다로울 뿐만 아니라 수학적인 알고리즘을 넣는 것이 어렵다. 하지만 그래픽 언어를 이용하여 다른 언어보다 쉽고 빠르게 프로그램을 구현하고 AD, DA컨버터가 내장 되어 있는 DAQ보드를 통해 보다 편하게 PC기반 플랫폼을 구성 할 수 있다. 또한 GUI 프로그램이기 때문에 사용자가 보다 편하게 제어 할 수 있는 환경을 만들 수 있다. 이 프로그램으로 현재 많이 쓰이고 있는 BLDC모터의 속도를 MT method를 이용하여 측정을 하고 PI제어를 함으로서 타당성을 보였다.

2. 본 론

2.1 시스템 구성

PC에서 NI사에서 제공하는 랩뷰라는 언어로 입출력 프로그램을 코딩하면 DAQ board에서 AD,DA를 통해 원하는 타겟 디바이스에 원하는 출력과 디바이스에서 나오는 입력을 PC로 읽을 수 있다. 현재 구성하고 있는 시스템에서는 전압의 세기를 이용하여 BLDC motor의 속도를 가변하고 속도에 따라 나오는 홀센서의 펄스파형을 통해 모터의 속도를 측정 하고 제어 할 수 있도록 하였다. 그림 1에서 시스템의 구성도를 나타내었다.



그림 1. 시스템 구성도

PC로 BLDC모터를 제어하기 위해 DAQ보드와 모터 드라이버가 필요하다.

2.2 BLDC속도 측정

BLDC모터 드라이버의 DC전압 컨트롤 단자에 0~5V전압을 인가해서 속도를 제어 할 수 있다. 여기에 인가하는 전압은 프로그램 상에서 다이얼로 가변 시킴으로서 DAQ보드에서 출력되는 전압이다. 속도측정은 일반적으로 널리 쓰이는 MT method를 구현 하였다.

2.3 MT method구현

측정시간 T_d 는 주어진 샘플링 시간 T_c 이후에 처음으로 발생되는 홀센서 펄스에 동기 되어 결정된다. 측정 속도 N_f 는 측정 시간과 그 동안에 회전자가 움직인 X(rad)에 의해 식 (1)과 같이 주어 진다.

$$N_f = \frac{60X}{2\pi T_d} = \frac{60X}{2\pi(T_c + \Delta T)} \quad (r/min) \quad (1)$$

1회전 당 펄스 수 P_{PR} 의 홀센서 펄스가 T_d 시간 동안에 m_1 의 펄스를 발생한다면 각 변위 X(rad)는 식(2)과 같이 구해진다.

$$X = \frac{2\pi m_1}{P_{PR}} \quad (2)$$

또한 측정 시간 $T_d = T_c + \Delta T$ 는 주파수 f_c 의 클럭펄스로 이산화 되는데, 그 값이 m_2 라 하면 T_d 는 식(3)과 같이 구해진다.

$$T_d = \frac{m_2}{f_c} \quad (3)$$

이상의 식에서 측정 속도 N_f 는 식(4)를 이용하여 구할 수 있다.

$$N_f = \frac{60 f_c m_1}{P_{PR} m_2} \quad (4)$$

클럭 펄스의 주파수는 20MHz이고 1회전 당 홀센서 펄스의 수는 18개 이므로 식(10)과 같이 구할 수 있다.

$$N_f = \frac{60 \cdot 2 \cdot 10^7 \cdot m_1}{18 \cdot m_2} \quad (5)$$

위의 수식을 그림 2처럼 코딩하였다.

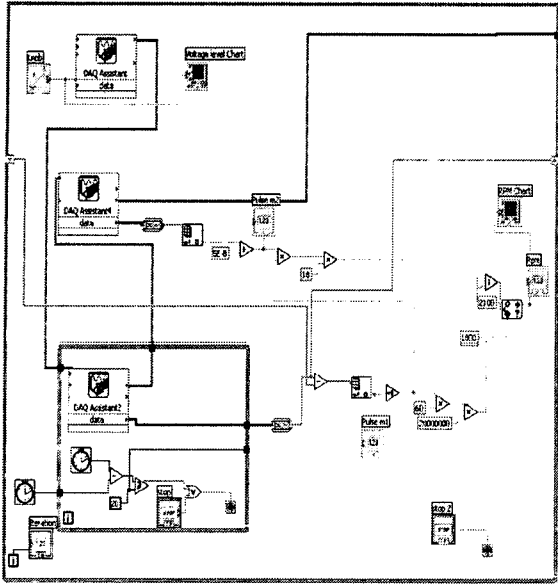


그림 2. MT method block

2.4 DC motor modeling

정류가 완벽하고 전류파형이 정확하며 변환기가 이상적인 직류 전압 V 를 공급해 준다고 가정하면 직류기는 식(6)과 같이 모델링 할 수 있다.

$$V_a = R_a i_a + L \frac{di_a}{dt} + e \quad (6)$$

부하 토크에 관한 운동방정식을 표현하면 식(7)과 같이 나타낼 수 있다.

$$T_e = J \frac{d\omega_m}{dt} + B\omega_m + T_L \quad (7)$$

계자 자속은 일정하다고 가정하면 자속을 상수 K 에 포함시킬 수 있고 식(6)과 (7)을 라플라스 변환하면 각각 식(8)과 (9)로 표현할 수 있다.

$$V_a = (R_a + sL_a)I_a + K\omega_m \quad (8)$$

$$T_e = (sJ + B)\omega_m + T_L \quad (9)$$

식(8)과 (9)를 기반으로 매트랩을 구현 하면 그림 3과 같이 표현 할 수 있다. 이 과정으로 전압에 따른 속도의 값을 시뮬레이션 할 수 있다.

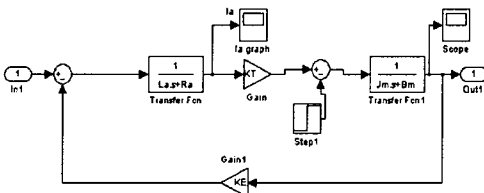


그림 3. DC motor Matlab simulink

2.5 Digital PI control 구성

비례적분 제어란 오차 신호를 적분하여 제어신호를 만들어 내는 적분제어를 비례제어에 병렬로 연결하여 사용하는 제어기법으로서 입력 신호는 식(10)과 같이 나타낼 수 있다.

$$Y(t) = K_p \times e(t) + K_i \times \int_0^t e(t) dt \quad (10)$$

식(10)을 PI 디지털 제어로 바꾸면 식(11)과 같이 표현할 수 있고 이를 Labview로 코딩하기 위한 5가지 단계로 표현 할 수 있는데 우선 측정시간 단위 dt 를 설정하고 오차 연산을 한 후 P게인과 오차를 곱해 비례제어값을 만들고 에러값과 dt , I게인의 곱과 이전까지의 에러 값의 합으로 적분제어값을 만든다. 마지막으로 내보내는 값은 P값과 I값의 합이 되는데 이 과정을 그림 4~8에 나타내었다.

$$Y(k) = K_p \cdot e(k) + \{u(k-1) + K_i \cdot e(k)\} \quad (11)$$

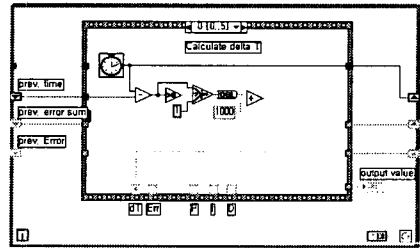


그림 4. I제어 dt설정

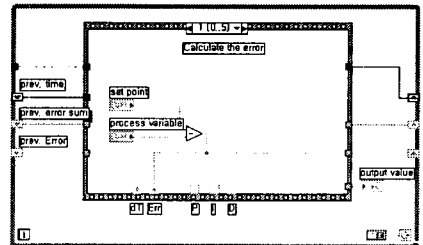


그림 5. 오차 연산

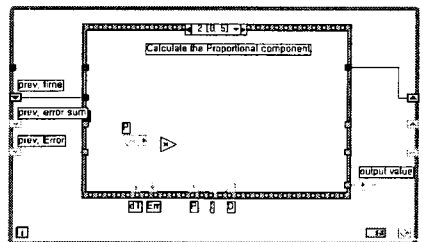


그림 6. P값 계산

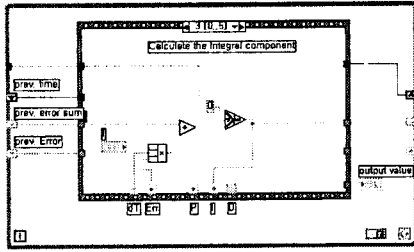


그림 7. I값 계산

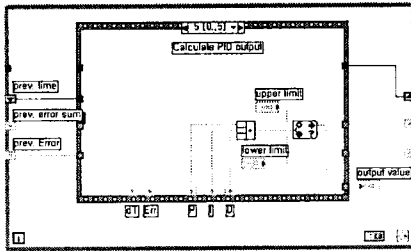


그림 8. 출력 값 결정

이 과정을 sub VI **simple PID.vi**로 만들어서 BLDC 모터 PI 컨트롤 프로그램을 그림9와 같이 구성하였고 모터의 모델링 부분을 마찬가지로 sub VI **GC Model**로 만들어서 모터의 전달함수를 구현 하였다.

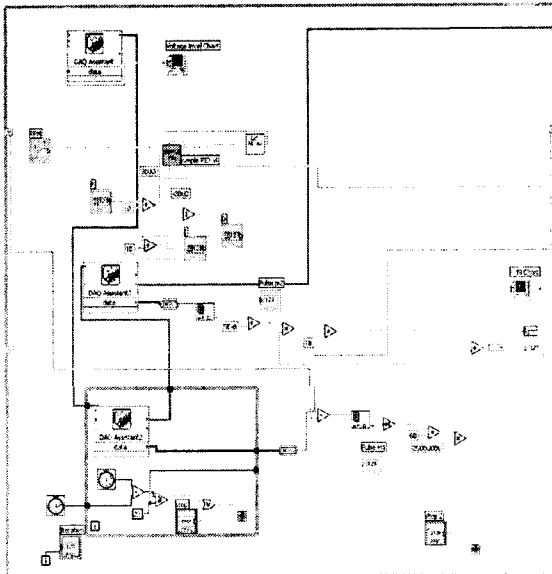


그림 9. PI control block

3. 실험 및 결과

그림 10은 오픈 루프로 돌렸을 때 MT method를 이용하여 속도측정을 하였을 때의 속도 그래프이다. 속도가

균일하게 나오지 않는 문제를 PI제어기를 연결 하였을 때 전압의 조정(그림11)으로서 그림 12와 같이 균일한 속도가 나오도록 제어하였다.

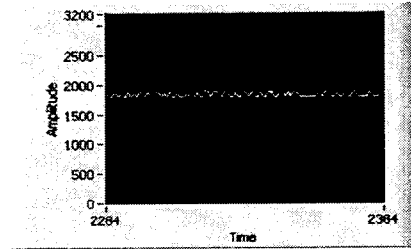


그림 10. MT METHOD SPEED CHART

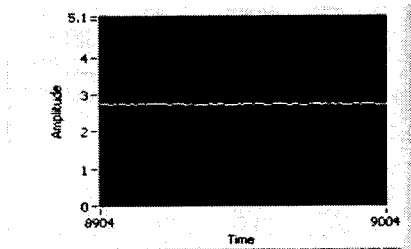


그림 11. PI CONTROL VOLTAGE GRAPH

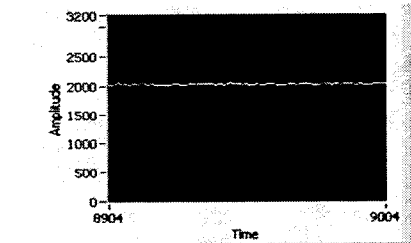


그림 12. PI control RPM graph

4. 결 론

본 연구에서는 그래픽 언어와 DAQ보드를 이용하여 모터의 속도측정 과 제어 알고리즘을 좀 더 쉽게 구현하였다. 속도측정과 제어를 위한 하드웨어를 구성하거나 일반적으로 쓰이는 텍스트 언어를 사용한 경우에 비교하여 대단히 수월하게 알고리즘을 변경 시킬 수 있으므로 유연성 있는 제어 구현이 가능하였다. 이로부터 알고리즘이 검증 되면 시행착오를 줄이고 시스템을 구성 할 수 있게 되었다.

[참 고 문 헌]

- [1] 설승기, "전기기기제어론", 도서출판 브레인코리아, pp.46-60, 84~108, 2002.
- [2] 박두영, "LabVIEW 컴퓨터 기반의 제어와 계측 solution", Ohm사, pp.50~140, 2002
- [3] Dae-Kyong Kim, Kwang-Woon Lee, Byung-Il Kwon, "Torque Ripple Reduction Method in a Sensorless Drive for the BLDC Motor", KIEE International Transactions on EMECS, vol. 4, pp.196-200, 12 2004
- [4] S.Y. han, S.H.Baek, Y.Kim, J.P.Kim, I.R.Jung, "A Study On BLDC Motor Control by Using I-P Algorithm", 학술대회 논문집, pp.988-990, 2000
- [5] Gyu-Man Cho, Yong Kim, Soo-Ilyun Baek, "Digital Controller of BLDC Motor Using DSP", 학술대회논문집, pp.988-990, 2000