

산업용 컴퓨터를 이용한 BLDC 모터 속도 제어에 대한 연구

*박 민중, 이 광호, 문 형섭, 홍 선기
호서대학교 정보제어공학과

A study for BLDCM speed control using industrial computer

*Min-Jong Park, Kang-Ho Lee, Hyoung-Seop Moon, Sun-Ki Hong
Department of information Control Engineering Hoseo university

Abstract - 본 연구에서는 그래픽 언어기반 소프트웨어인 LabVIEW와 Real-time Module 및 FPGA 모듈을 포함한 산업용 컴퓨터를 이용하여 BLDC 속도제어를 하려고 한다. 이 프로그램은 GUI기반으로 수집, 계측, 제어부분에 탁월한 성능을 가지고 있는 그래픽 언어이다.

BLDC 속도제어를 하기 위해서 BD80-220N100의 BLDC를 사용하였으며, 드라이버를 연결하여 BLDC 속도제어 GUI 프로그램을 구현하였다.

BLDC 속도제어를 하기 위해서는 먼저 속도 측정하여 모터의 속도에 대한 안정도를 판별하고 사용자가 원하는 응답에 안정하면서 빠르게 도달할 수 있도록 PI 속도제어를 하였다.

이를 바탕으로 그래픽 언어에 대한 이해 및 프로그램 코딩 습득을 하여 BLDC 모터 속도제어에 전반적인 가치를 두려고 한다. 본 연구를 통하여 하드웨어를 구성하거나 일반적으로 쓰는 텍스트 언어를 사용할 경우와 비교하여 수월하게 알고리즘을 변경시키며 유연성 있는 프로그램을 구현하였다.

1. 서 론

현재 제일 많이 사용되고 있는 언어는 C 언어이다. 그러나 C 언어를 대체하여 쓸 수 있는 그래픽 언어 기반 소프트웨어인 LabVIEW와 Real-time Module 및 FPGA 모듈을 포함한 산업용 컴퓨터를 이용하여 BLDC 속도제어를 하려고 한다. 이 프로그램은 GUI기반으로 수집, 계측, 제어부분에 탁월한 성능을 가지고 있는 그래픽언어이다. BLDC 속도 제어를 하기 위해서 BD80-N220100의 BLDC를 사용하였으며 드라이버를 연결하여 BLDC 모터 속도제어 GUI(Graphical user interface)프로그램을 구현하여 GUI언어의 활용성과 속도 제어에 관한 이해를 확립하였다.

첫째로 BLDC 속도제어를 하기 위해서는 먼저 속도 측정을 하여야 하는데 BLDC는 홀센서 신호를 드라이버에서 받아 펄스로 출력하게 된다. 이 펄스를 이용하여 모터에 대한 속도 측정을 한다. 이러한 속도 측정을 하기 위한 방법으로 M method, T method, M/T method를 적용하여 모터 속도 측정을 할 수가 있다.

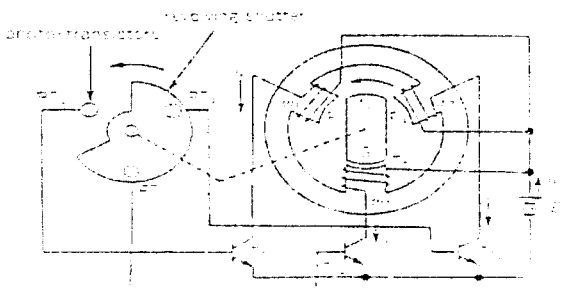
둘째로 속도 측정이 끝나면 모터 제어를 위해 시스템을 폐루프화 하여 모터의 속도에 대한 안정도를 판별하고 사용자가 원하는 응답에 안정하면서 빠르게 도달하는 것과 외란 및 부하에 대해 PI 속도 제어를 하려고 한다.

2. 본 론

2.1 BLDC 모터

BLDC(brushless DC) 전동기는 브러시 전동기가 뒤집힌 모양으로서 전기자가 고정자이며 자석이 회전자로 되어 있다. 회전자가 영구자석으로 있고 자기계어 모드로 동작(회전자 위치 센서를 사용하며 고정자 권선에 전류를 제어하기 위해 인버터를 사용한다.)하게 되어 있다.

회전자석계자의 회전위치를 검출하여 그것에 응답해서 적절한 전기자전류를 공급하여 2상 또는 3상 전기자에 의해서 회전자속을 항상 수직이 되도록 회전자자력을 생기게 하는 것이다. 일반 직류 모터와 거의 구조가 유사하여 플레밍의 법칙에 의한 회전력을 발생됨으로 부하에 상응하는 토크를 발생하게 되어 전동기를 안정된 상태로 동작하게 된다.



<그림 1> 3상 유니폴라 BLDC

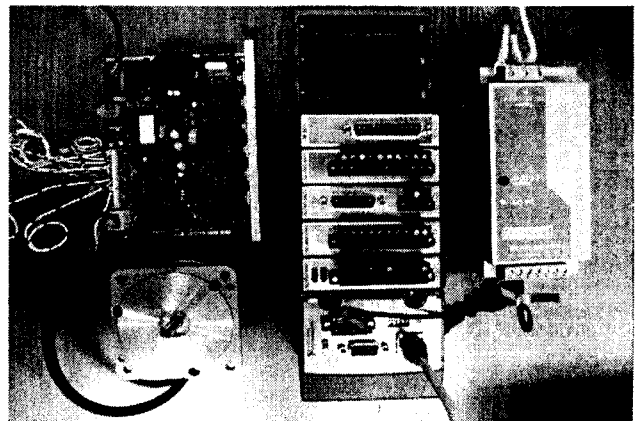
2.2 시스템의 구성

PC로부터 일반 마이컴에서 하는 바와 같이 GUI로 프로그램을 만들고 그 프로그램을 PXI의 산업용 컴퓨터를 거쳐 Compact - RIO device로 다운로드되어 Real-time Module 및 각각의 FPGA Module을 통해 그래픽 방식의 프로그램에 기반을 둔 VHDL 코드를 생성한 다음 자동으로 최적화, 합성 및 다운로드를 실행함으로써 사용자가 원하는 application을 신속하게 구성 및 개발 할 수가 있게 된다.

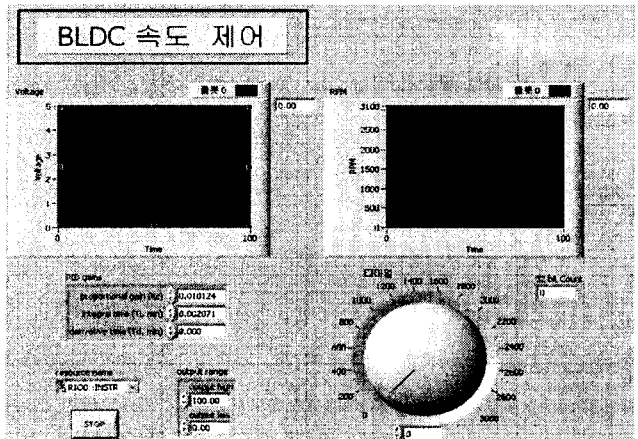
FPGA의 여러 가지 모듈 중 디지털 in/output 및 아날로그 in/output 을 사용하여 BLDC에서 나오는 3상의 홀센서 신호를 드라이버로 카운터 하여 신호 변환한 1개의 펄스로 출력시키는 페루프 형식으로 5V에서 구동하는 모터의 전압을 입력 Rpm에 맞게 가변하여 속도 측정 및 제어를 할 수가 있게 된다.



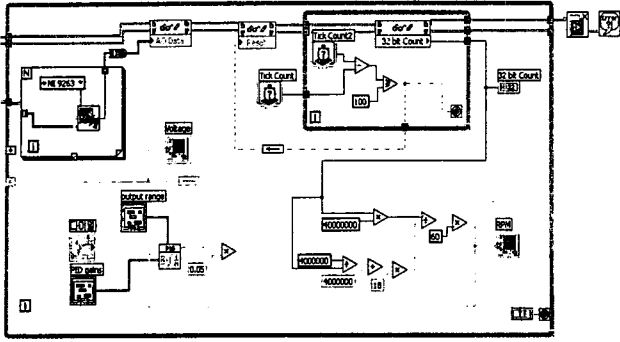
<그림 2> BLDC 속도 제어를 위한 시스템 구성도



<그림 3> 실제 시스템 구성



<그림 4> Display화 한 프로그램 시스템



〈그림 5〉 속도측정 및 PI제어 구동 GUI프로그램

2.3 BLDC 속도 측정

엔코더로부터 나오는 일정한 펄스를 가지고 모터에 대한 속도를 계산하는 3가지 방법이 있다.

첫 번째로 일정한 샘플링시간 T_c 시간마다 발생하는 엔코더 펄스의 개수를 세어 BLDC의 회전 속도를 측정하는 M 방식의 속도 측정 방법으로

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

이며 고속에서 많이 쓰이며 저속에서는 측정되는 속도가 각 샘플링 주기에서 순간적으로 매우 큰 차이를 보이게 되어 속도 제어특성에 있어서 나빠진다.

두 번째로 엔코더의 펄스와 펄스사이의 시간 간격을 측정하여 f_c 의 클럭 펄스(Clock pulse)가 m 개 들어갔을 때의 속도를 측정하는 T 방식의 속도 측정 방법으로

$$N_f = \frac{60f_c}{P_{PR}m} \quad (r/min) \quad (2)$$

이고 저속으로 정밀한 속도를 측정할 수 있으며 측정에 따른 시간지연이 적다는 장점이 있다. 그에 반해 속도 측정시간이 속도에 따라 가변되는 문제점이 있다.

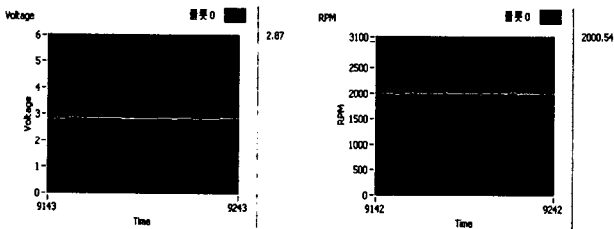
마지막 방법으로 고속에서 좋은 특성을 보이는 M방식과 저속에서 유리한 T방식을 결합한 방식으로 두 가지의 측정방법에 대한 오차를 줄이는 M/T 방식의 속도 측정방법으로

$$N_f = \frac{60X}{2\pi T_d} = \frac{60X}{2\pi(T_c + \Delta T)} \quad (r/min) \quad (3)$$

측정시간 T_d 동안에 m_1 의 펄스가 발생하고 $T_d = T_c + \Delta T$ 는 주파수 f_c 의 클럭 펄스로 이산화 되는데 그 값이 m_2 라 하면 측정 속도 N_f 는 다음과 같아진다.

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

위의 식과 같이 측정시간 T_d 때문에 일반적인 속도 범위에서 좋은 특성을 갖는다. 이 논문에서는 세 번째 방법을 사용하였고 위의 〈그림 5.2 GUI 프로그램-2〉에서 속도 측정방법을 100ms마다 측정할 수 있도록 구성하였다. 그 데이터들을 수집하여 그래프아이콘에 대입하여 아래와 같은 출력을 얻었다. 아래 〈그림 6 M/T 방식의 속도 측정〉은 P_{PR}이 18 펄스인 모터에 2000 Rpm의 입력치를 주었을 때 출력되는 약 60개의 펄스로 전압 및 BLDC의 속도그래프이다.



〈그림 6〉 M/T 방식의 속도 측정결과

2.4 BLDC PI 속도 제어

PI제어를 사용하면 어떠한 일정한 기준입력 및 외란입력에 대하여 빠른 응답상태 및 정상상태오차가 발생하지 않는다는 것을 알 수 있다. 특히 비례제어기(K_p) P계인은 정상상태에 빠르게 도달하게 하는 반면 정상상태 오차를 만든다.

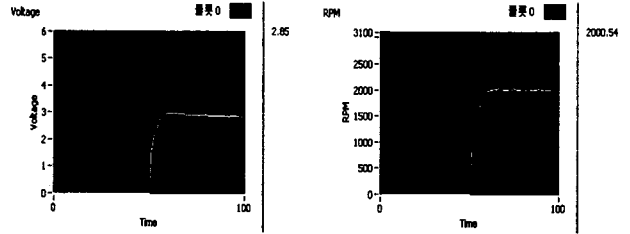
이러한 정상상태 오차를 적분제어기(K_i) I계인을 이용하여 정상상태오차가 발생하지 않도록 하는 효과를 가지고 있다. 그러나 적분제어기 한가지만을 사용하였을 경우 시스템의 안정도를 저하시키는 경향이 있다. 그러므로 제어기에 적분제어 요소만을 사용하는 경우는 극히 드물고, 제어시스템을 안정하게 하며 일정한 입력에 대하여 정상상태오차가 발생하지 않도록

할 수 있는 비례 - 적분 제어기를 사용하였다. 제어기에 대한 전달함수 형태는 다음과 같다.

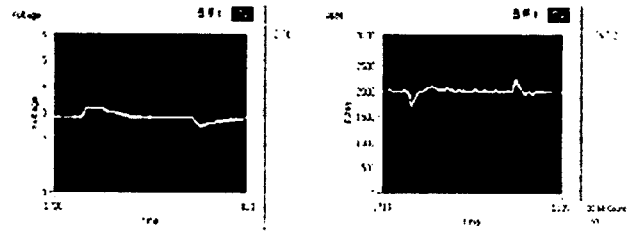
$$K(s) = K_p + \frac{K_i}{s} \quad (5)$$

LabVIEW 프로그램은 〈그림 5 GUI 프로그램-2〉에서와 같이 한가지의 아이콘으로 이러한 프로그램 코딩을 간단히 끝낼 수 있었다. 이러한 기능을 텍스트 언어로 구성하게 되었을 때 걸리는 많은 시간투자를 짧게 끝낼 수 있었으며 C언어를 모르는 사람이라도 빠른 이해를 할 수 있을 것이다.

아래 그림들은 2000Rpm으로 모터를 구동하기 위한 화면으로 BLDC가 얼마만큼의 시간에 정상상태에 도달하여 안정화를 나타내고 있는가를 나타내고 있는 그림과 BLDC에 일정한 부하를 걸어 그 외란에 따른 오차 값을 PI제어함으로써 응답상태에 따른 정상상태를 유지하는 것을 나타내고 있는 그림을 보여주어 이론적인 내용과 일치함을 볼 수가 있었다. BLDC는 P_{PR}이 18 펄스로 속도제어 결과 100ms당 60펄스의 출력으로 ±33RPM의 오차를 보였다.



(a) 시동 응답



(b) 외란 및 부하 응답

〈그림 7〉 PI 제어 응답

3. 결 론

위의 같이 LabVIEW의 그래픽 언어를 사용하여, 보다 편리하고 좀 더 체계적인 프로그램 완성에 한발자국 나아갈 수가 있었다. 일반적으로 모터 제어를 하려면 Micro Controller를 사용하여 C로 프로그램을 만들어야 하며 수학적인 알고리즘도 거기에 따라 구현해야 한다. 하지만 C로 프로그램을 짜면서 시행오차 및 프로그램 다운로드, 실행 등 많은 불편함이 따른다.

그러한 단점을 해결해 주며 계측 및 제어, 측정에 성능 및 편의성이 우수한 프로그램을 이용하여 BLDC 제어를 하였다. 특히 LabVIEW는 데이터 수집 및 분석하고 결과를 표시하기에 적합한 솔루션이라고 할 수 있다.

본 연구를 통하여 그래픽 언어에 대한 이해 및 프로그램 코딩 습득을 하여 BLDC 속도 제어에 전반적인 가치를 두려고 한다. 마지막으로 연구를 통하여 하드웨어를 구성하거나 일반적으로 쓰는 텍스트 언어를 사용한 경우와 비교하여 수월하게 알고리즘을 변경시키며 유연성 있는 프로그램을 구현을 하였다.

감사의 글

본 연구는 산업자원부 지방기술혁신사업(RTI04-01-02) 지원으로 수행되었음

〔참 고 문 헌〕

[1] 설승기, "전기기기 제어론", 브레인코리아, 2002
 [2] 광두영, "LabVIEW Express", Ohm사, 2003
 [3] 김중식의 3명, "제어 시스템 설계", 정문각, 2003
 [4] Paresh C. Sen, "Principles of Electric Machines and power Electronics", SciTech media, 1996