2007년도 대한전기학회 하계학술대회 논문집 2007. 7. 18 - 20

LabVIEW의 모델기반 제어기 설계와 Compact RIO를 이용한 직류전동기 구동 시스템

송의섭, 이희준, 이용석, 지준근
순천향대학교 전기통신시스템공학과

DC Motor Drive System Using Model Based Cotroller Design of LabVIEW and Compact RIO

Yui-Sub Song, Hui-Jun Lee, Yong-Suk Lee, Jun-Keun Ji
Department of Electrical Communication Syste m Engineering, Soonchunhyang University

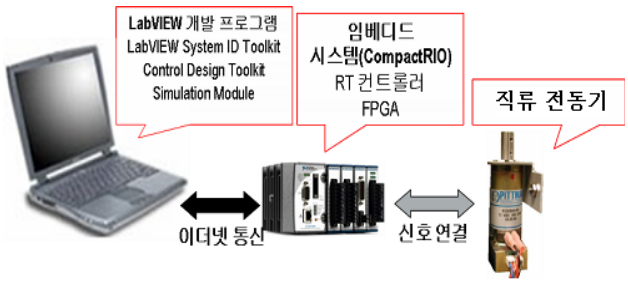
Abstract - 본 논문에서는 모델기반의 제어기 설계 프로그램인 National Instruments(NI)사의 System Identification Toolkit과 Control Design Toolkit, Simulation module을 사용하여 기존의 제어기 설계방식보다 쉽고 편리하게 제어기를 설계할 수 있었다. 직류전동기의 속도 제어시스템을 구현하기 위해서 하드웨어는 NI사에서 제공하는 실시간 제어기(Real-Time Controller:RT) CompactRIO를 사용하였다. 먼저는, 테스트 입력 신호를 전동기에 인가하고 얻은 출력신호를 통해 제어대상 플랜트인 직류전동기 구동시스템의 전달함수를 구할 수 있었다. 다음으로는 원하는 제어응답성능을 갖는 극점, 영점 제어기를 설계한 후, 모의실험을 통해 속도제어응답을 확인할 수 있었고, 실시간프로그램으로 다운로드하여 실제 전동기 구동시스템의 실험을 통해서 설계된 속도제어기의 응답 결과를 모의실험과 비교하여 검증하였다.

1. 서 론

오늘날 제어공학의 발전으로 많은 제어기술이 선보여지고 있다. 이러한 제어기술은 오늘날의 복잡한 시스템을 제어가능하게 만들었고 제어기의 성능을 향상시키었다. 그렇지만 이러한 제어기술의 향상에도 불구하고 아직도 복잡한 시스템을 제어하기 위해선 제어 대상을 해석적으로 표현할 필요가 있다. 이러한 해석적 표현은 시스템이 복잡하고 불연속적일수록 표현이 불가능해진다. 그리하여 복잡한 시스템의 동작 확인과 최상의 제어기 설계를 하는 것은 엔지니어에게 꼭 필요하다. 이러한 요구조건을 만족시키는 제어시스템 설계도구로서 그래픽기반 프로그램인 LabVIEW가 있고 실시간 제어가능한 시스템으로서 NI사가 제공하는 CompactRIO(이하 cRIO)가 있다. 본 연구에서는 cRIO의 실시간 제어시스템과 LabVIEW 프로그램의 System Identification Toolkit과 Control Design Toolkit, Simulation Module을 사용하여 쉽고 빠르게 직류전동기 구동 시스템의 속도제어기 설계 및 실시간 실험을 하였다.

2. 전체 시스템 구성

그림 1은 LabVIEW의 모델기반 제어기 설계 프로그램들과 임베디드 시스템 구현을 위한 cRIO 및 직류전동기로 구성되는 전체 시스템 구성을 보여주고 있다.



<그림 1> 전체 시스템 구성도

2.1 LabVIEW 모델기반 제어기 설계 프로그램

LabVIEW는 그래픽 기반의 프로그램으로써 그래픽한 아이콘을 사용하여 프로그램을 쉽게 작성하고 디버깅할 수 있다. 그리고 그래픽 사용자 인터페이스(Graphic User Interface:GUI)에서 설정값을 입력하고 버튼을 이용해 실제적인 액추에이터를 구동시킴으로서 자동제어나 계측을 쉽게 원하는 대로 할 수 있다. 여기서 사용한 LabVIEW 개발프로그램에는 System Identification Toolkit과 Control Design Toolkit, Simulation Module이 있다.

2.2 CompactRIO 임베디드 시스템

cRIO는 전력 소모가 적은 실시간 임베디드 프로세서를 고성능의 FPGA 칩셋과 결합한 하드웨어다. cRIO는 LabVIEW FPGA 요소 I/O 함수를 사용하는 각 I/O 모듈의 I/O 회로로 하드웨어를 집적 연결한다. cRIO 임베디드 시스템은 LabVIEW Real-Time의 어플리케이션을 안정적이고 높은 신뢰성을 갖게 한다.

2.3 직류전동기

여기서 사용한 전동기는 직류 서보 전동기로서, 고정자로 영구자석을 사용하고 회전자(전기자)로 코일을 사용하여 구성한 것으로 전기자에 흐르는 전류에 의해서 회전력을 생성시키는 전동기이다. 직류 서보 전동기는 제어용 전동기로서 매우 우수한 특성을 가지고 있다. 예를 들면 급격한 가속성, 큰 시동 토크, 리니어한 회전 특성 등 대체로 제어용 전동기에 요구되는 모든 성능을 겸비한 우수한 전동기이다. 이러한 제어시스템에 많이 사용되는 직류 서보 전동기는 토크와 전류가 비례하여 선형제어시스템을 구성하는 것이 가능하므로 비교적 간단하게 안정된 제어기 설계가 가능하다.

3. 시스템의 하드웨어 구성

3.1 전동기 구동부

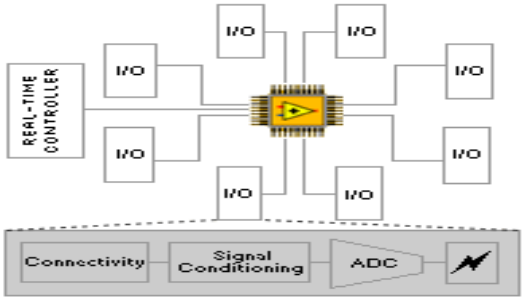
직류전동기를 구동하기 위해서 전동기에 입력되는 전력을 제어하는 전자 회로부를 "구동회로" 또는 "서보 증폭기"라고 한다. 서보 증폭기는 내장된 반도체 소자를 구동하는 방식에 따라 크게 2가지로 구분할 수 있다. 하나는 "선형 서보 증폭기"로서 바이폴라 트랜지스터를 선형 동작 영역에서 구동하는 방식이고, 다른 하나는 "PWM 서보 증폭기"로서 바이폴라 트랜지스터나 MOSFET를 스위칭 모드에서 구동하는 방식을 말한다. 전압 제어 방식의 PWM 서보 증폭기는 전동기에 인가되는 전압을 제어함으로써 전동기를 제어하게 된다. 본 연구에서는 직류전동기 구동회로에 L298이라는 H-브리지 서보 증폭기를 사용하였다.

3.2 FPGA 프로그램

FPGA는 실리콘칩으로서 독립적인 로직게이트로 되어 있다. FPGA의 기능은 규정되어진 소프트웨어의 FPGA 게이트를 구성할 수 있다는 점이다. 이점은 어플리케이션에 따라 필드를 재구성할 수 있다. 본 연구에서는 NI사에서 제공해 주는 cRIO의 FPGA를 사용하였다.

그림 2는 cRIO의 LabVIEW FPGA 요소를 I/O 함수로 나타낸 것으로 사용하는 각 I/O 모듈의 I/O 회로로서 하드웨어를 직접 연결한다. 각 I/O 모듈에는 내장 연결성, 신호 컨디셔닝, 변환 회로(ADC 또는 DAC) 및 절연 기능이 있다.

cRIO의 NI-9401은 8채널 TTL 디지털 입력/출력 모듈로서 FPGA 프로그래밍을 하여 PWM 신호를 출력하고 직류전동기의 속도 측정을 위해 엔코더 펄스를 입력 받을 수 있다.



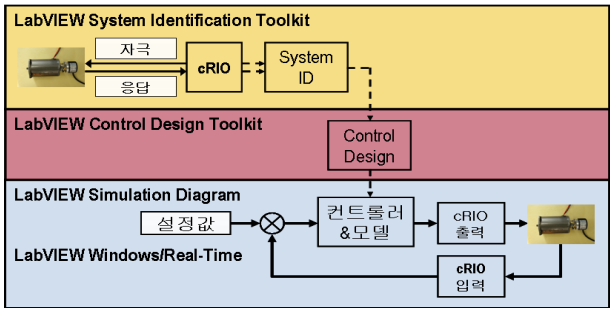
<그림 2> CompactRIO의 FPGA 구성

3.3 실시간(Real-Time) 시스템

cRIO는 Real-Time 컨트롤러가 내장되어 있다. cRIO 임베디드 시스템은 LabVIEW Real-Time 어플리케이션을 안정적이고 결정적으로 실행하는 산업용 200MHz 펜티엄 클래스 프로세서를 포함하고, 내장된 수천 개의 LabVIEW 함수에서 선택하여, 실시간 제어, 분석, 데이터 로깅 및 통신용 멀티스레드 임베디드 시스템을 구축할 수 있게 하였다. 이러한 Real-Time시스템은 높은 신뢰성을 가지고 어플리케이션의 완성도를 높인다. 본 연구에서는 실시간 시스템을 사용하여 직류 전동기를 실시간으로 제어 하였고, 실시간으로 모의실험과 비교할 수 있었다.

4. LabVIEW 프로그래밍

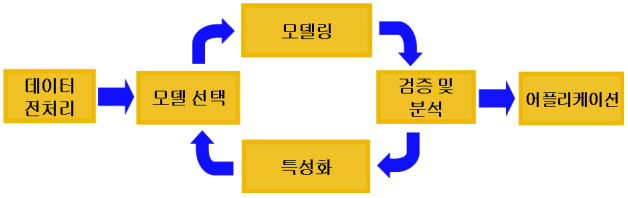
그림 3은 LabVIEW 프로그램의 개요를 나타낸 것이다. 본 연구에서는 직류 전동기에 자극을 주고 그에 대한 응답을 받아 시스템 식별(System Identification; 이하 ID)을 하였고 System ID로 추정하여 찾은 모델을 Control Design Toolkit를 사용하여 제어기를 설계한다. 설계된 제어기로 Simulation Module을 사용하여 시뮬레이션한 결과와 실제 직류 전동기의 실험결과를 비교 검증한다.



〈그림 3〉 LabVIEW 프로그램의 개요

4.1 시스템 식별(System Identification)

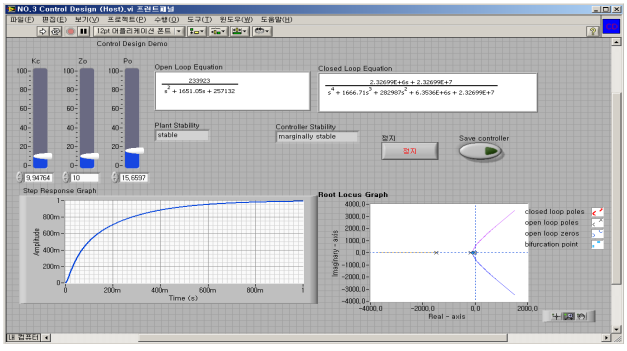
그림 4는 System ID의 과정을 보여주고 있다. System ID를 하기 위해 자극을 주고 받는 응답이 있어야 한다. 이 응답으로 추정한 모델의 모델식을 만들고 만들어진 모델식은 시뮬레이션을 통해 검증과 분석을 한다.



〈그림 4〉 시스템 ID의 단계

4.2 제어기 설계(Control Design)

시스템 식별에서 모델의 전달함수를 구함으로써 상승시간(rise time), 정착시간(Setting time), 오버슈트(Overshoot)에 대한 요구조건에 맞는 제어기를 설계할 수 있게 되었다. 그림 5에서 보여지는 Control Design 프로그램의 프런트 패널은 종속제어기를 플랜트 앞에 놓는 방식에 따라서 폐루프시스템의 동작이 어떻게 결정되는 지를 보여준다.

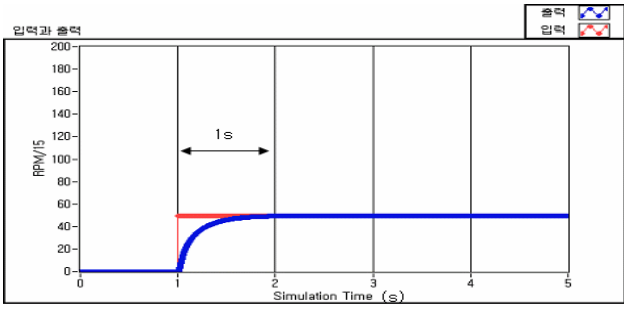


〈그림 5〉 Control Design 프로그램의 프런트 패널

그림 5에서 비례 이득, 영점, 극점을 조절하여 제어기를 설계한다. 이 같은 방법을 극배치법(pole placement)이라 한다. 극배치법이란 극점의 위치와 시스템 성능과의 관계를 고려하여 폐루프 시스템의 극점들의 위치를 적절히 지정하고 이 위치에 폐루프 시스템의 극점이 놓이도록 제어기를 설계함으로써 원하는 성능목표를 이루는 제어기 설계방식을 말한다. 본 논문에서는 적절한 극점의 위치를 찾기 위해 비례 이득, 영점, 극점의 조절 게이지로 그림 5에서 보여지는 폐루프 시스템에 대한 단위 스텝응답의 상승시간, 정착시간, 오버슈트를 조절하였다.

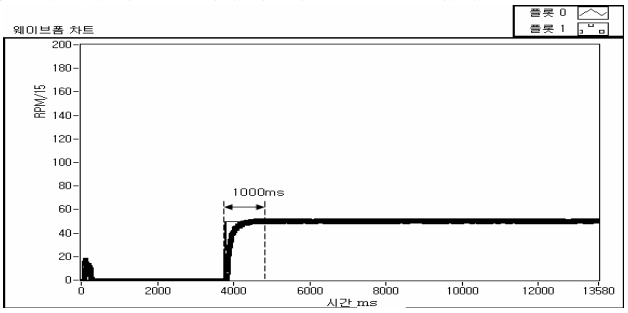
5. 시뮬레이션 및 실험결과

System ID를 사용하여 구해진 전동기의 전달함수와 Control Design을 사용하여 구해진 제어기를 가지고 시뮬레이션을 하여 보았다. 그림 6은 시뮬레이션의 결과이다. 결과를 보면 설정값 50[RPM/15]에 따라 응답이 따라가는 것을 볼 수 있다. 이는 오버슈트가 없고 정착시간까지 약 1초가 걸린다. 이는 위에서 설계한 제어기의 응답 결과와 같다.



〈그림 6〉 시뮬레이션 결과 그래프

그림 7은 전동기를 실제로 구동했을 때의 실험 결과이다. 시뮬레이션에서 처럼 설정값 50[RPM/15]으로 주었을 때 오버슈트 없이 약 1초의 시간에서 정착하는 것을 볼 수 있다. 이는 시뮬레이션 결과가 실제 실험했을 때 거의 비슷한 결과가 나온다는 것을 알 수 있다.



〈그림 7〉 실제 실험 결과 그래프

6. 결 론

본 연구에서는 직류전동기 구동시스템의 전달함수를 구하기 위해 System Identification Toolkit을 사용하였고, 속도제어기 설계를 위해 Control Design Toolkit을 사용하였다. System Identification Toolkit은 직류전동기의 정수와 시스템 파라미터를 몰라도 전동기 구동 시스템의 전달함수를 구할 수 있었고, Control Design Toolkit으로는 보다 쉽게 속도제어기를 설계할 수 있었다. 설계한 속도제어기를 Simulation module을 이용하여 시뮬레이션을 한 결과를 확인한 후, cRIO를 이용하여 실제 직류전동기를 구동한 실험결과와 일치함을 알 수 있었다. LabVIEW 프로그램과 NI사에서 제공해 주는 하드웨어인 cRIO 및 제어설계 소프트웨어인 Contril Design을 이용할 경우 기존의 제어시스템 개발시보다 제어기의 설계 및 구현이 매우 쉽고 편리하며, 응답결과도 확인이 편리하여 제어시스템 해석 및 설계를 쉽게 이해하고 빠른 시간에 제어 시스템을 설계할 수 있었다.

[참 고 문 헌]

[1] 곽두영, “컴퓨터 기반의 제어와 계측 LabVIEW”, Ohm사, 2006.
[2] 전범수, 신현근, “MMI(Man Machine Interface)에 의한 DC 서보모터 속도제어”, 순천향대학교 정보기술공학부 학사학위논문, 2001.
[3] 이용석, 성인채, “dSPACE시스템을 이용한 직류전동기 구동 시스템”, 순천향대학교 정보기술공학부 학사학위논문, 2005.
[4] National Instrument, FPGA Module User Manual, 2004.
[5] National Instrument, LabVIEW Control Design Toolkit User Manual, 2006.
[6] National Instrument, LabVIEW System Identification Toolkit User Manual, 2006.
[7] National Instrument, LabVIEW 8.0 Simulation Module Help, 2006.