

LabVIEW와 cRIO를 이용한 AC 서보시스템의 속도제어

윤기현, 지준근  
 순천향대학교 전기통신시스템공학과

Speed Control of AC servo system using LabVIEW and cRIO

Ki-Hyeon Yun, Jun-Keun Ji  
 Dept. of Electrical Communication System Engineering, Soonchunhyang University

**Abstract** - This paper presents a speed control of AC servo system using LabVIEW program and cRIO (Compact RIO) hardware which is a real-time controller made in National Instruments company. LabVIEW is a GUI programming language easy to implement control system and cRIO is a reconfigurable hardware platform which is very simple. Therefore LabVIEW and cRIO will be excellent tools to design and implement control system.

1. 서 론

본 논문에서는 GUI 프로그래밍언어인 LabVIEW와 NI사의 cRIO(CompactRIO)를 이용하여 AC 서보시스템의 속도제어를 설계하였다. 제어기술은 여러 분야의 기반 기술로 큰 역할을 맡고 있다. 그러나 이처럼 중요한 제어기술은 복잡한 수식의 표현 등으로 어렵다는 말을 자주 듣게 되며, 이론뿐만 아니라 설계 후 적용이라는 실험과정을 수반하여야 제어기 설계에 대한 이해와 설계능력을 갖추게 되는데, 그런 과정이 거의 수반되지 않고 있는 실정이다. 그러나 GUI기반 프로그래밍의 용이한 접근성과 쉬운 프로그래밍, 그리고 cRIO라는 LabVIEW 기반의 하드웨어로 이런 복잡하고 어려운 과정 없이 실험을 진행할 수 있어 좀 더 쉽게 AC 서보시스템의 접근이 가능하다.

2. 본 론

2.1 PID 속도제어기

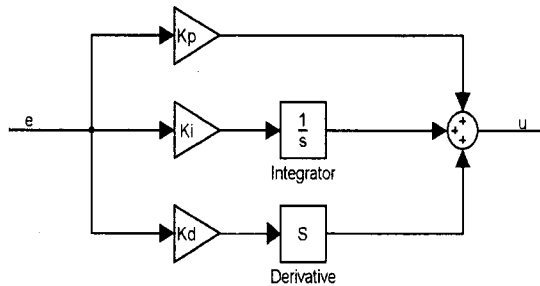


그림1. PID 제어기의 블록도

본 논문에서는 가장 널리 쓰이는 PID 제어기를 이용하여 속도제어를 설계하였으며, 다음과 같은 식으로 나타낼 수 있다.

$$u = K_p e + K_I \int e dt + K_D \frac{de}{dt}$$

각각의 제어기는 고유의 응답특성을 가지는데, 비례이득  $K_P$ 의 값은 플랜트응답의 상승시간을 줄이는 효과와 정상상태오차를 줄이는 효과가 있으나 정상상태오차를 없애지는 못한다. 그리고 적분이득  $K_I$ 의 값은 정상상태 오차를 제거하는 효과를 가지고 있지만 과도응답특성을 좋지 않게 만들 수 있고, 미분이득  $K_D$ 의 값은 시스템의 안정도를 향상시키는 효과를 가지고 있어서 오버슈트를 줄이고 과도응답 특성을 향상시킬 수 있지만 잡음에 약하여 대부분의 제어시스템에서는 비례이득과 적분이득 두 가지를 이용하여 제어기를 설계한다.

PID 제어기 각각의 이득값 변화에 대한 특성을 표1로 정리하였다.

표1. PID 이득값 변화에 대한 응답특성

응답특성 \ 이득값	상승시간	오버슈트	정착시간	정상상태 오차
$K_P$	감소	증가	약간 변화	감소
$K_I$	감소	증가	증가	제거
$K_D$	약간변화	증가	감소	약간변화

2.2 cRIO와 AC 서보시스템의 구성

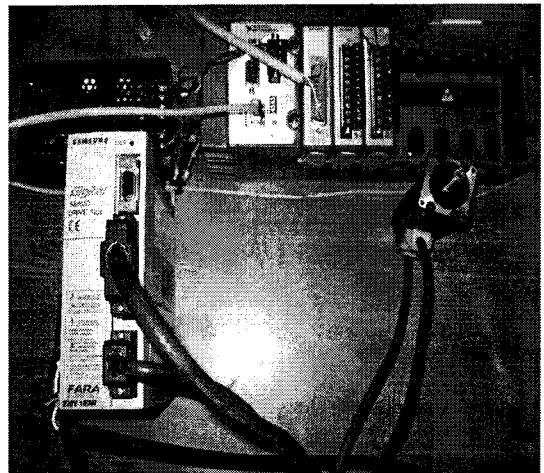


그림2. 전체시스템 구성

그림2는 전체 시스템의 실제 모습으로 cRIO, 전원을 공급하는 전원공급장치, 전동기 및 서보 드라이버로 구성 되어있다.

## 2.2.1 cRIO(CompactRIO)의 구성 및 사양

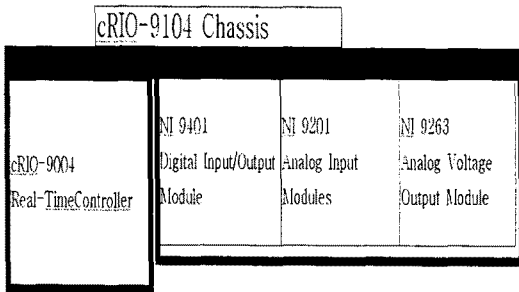


그림3. cRIO(CompactRIO)의 구성도

그림3과 같이 cRIO는 3부분으로 구성되어 있다. 본체라고 할 수 있는 Realtime Controller인 cRIO-9004와 FPGA가 내장되어있고, 각종 I/O모듈을 장착할 수 있는 cRIO-9104 채시 부분이 있으며, cRIO-9104 채시에 장착하여 사용하는 각종 I/O모듈이 있다.

본 논문에서 사용하는 I/O모듈은 3가지로 엔코더 펄스를 받는 NI 9401모듈과 속도 모니터링 신호를 받는 NI 9201모듈, 그리고 토크지령 출력을 내보내는 NI 9263모듈이다. cRIO-9104 채시는 기본적으로 40MHz의 클럭으로 동작을 한다. 아래의 표2,3,4는 실험에서 사용하는 I/O모듈의 사양을 나타낸다.

표2. NI9401 I/O 모듈의 사양

8-Channel TTL Digital Input/Output Module	
Input type .....	TTL, single-ended
Output type .....	TTL, single-ended
Maximum input-signal switching frequency by number of input channels	
- 8 input channels	9MHz
- 4 input channels	16MHz
- 2 input channels	30MHz

표3. NI9201 I/O 모듈의 사양

8-Channel, Analog Input Module	
ADC resolution .....	12bit
Input range .....	±10V
Settling time .....	2μs

표4. NI9263 I/O 모듈의 사양

4-Channel, Analog Voltage Output Module	
DAC resolution .....	16bit
Output range .....	±10V
Operating voltage	
Nominal .....	±10.7V
Minimum .....	±10.3V
Maximum .....	±11V

## 2.2.2 AC 서보시스템의 구성

AC 서보시스템은 100[W] 영구자석 동기전동기와 전동기를 구동하는 서보드라이버 및 각종 신호커넥터로 이루어져 있다.

표5. AC 서보시스템 구동용 신호

신호종류	신호에 대한 설명
모터입력신호 (+)	모터의 토크 지령
모터입력신호 (GND)	모터의 토크 지령 GND
엔코더 A	모터의 엔코더 출력 A상
엔코더 $\bar{A}$	모터의 엔코더 출력 $\bar{A}$ 상
엔코더 B	모터의 엔코더 출력 B상
엔코더 $\bar{B}$	모터의 엔코더 출력 $\bar{B}$ 상

## 2.3 AC 서보시스템의 속도제어

### 2.3.1 LabVIEW 프로그램

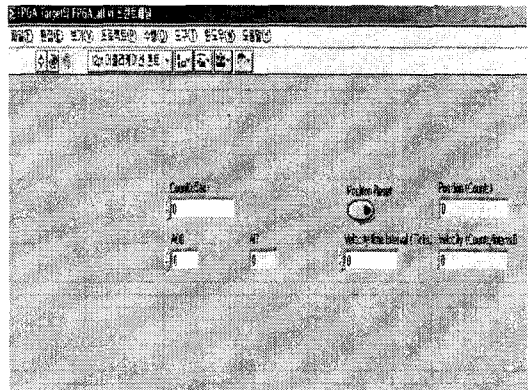


그림4. FPGA 프로그램(프론트패널)



그림5. FPGA 프로그램(블록다이어그램)

그림4와 그림5는 LabVIEW로 설계한 FPGA 프로그램의 프론트 패널과 블록다이어그램이다. 서보드라이버에서 출력되는 속도 모니터링 신호를 측정하기 위한 Analog input 신호, 토크 지령을 주기위한 Analog output 신호, 그리고 엔코더 펄스를 카운트하기 위한 Digital input 신호를 받거나 내보내기위한 LabVIEW 프로그램을 컴파일하여 cRIO-9104내부의 FPGA에 다운로드 한다.

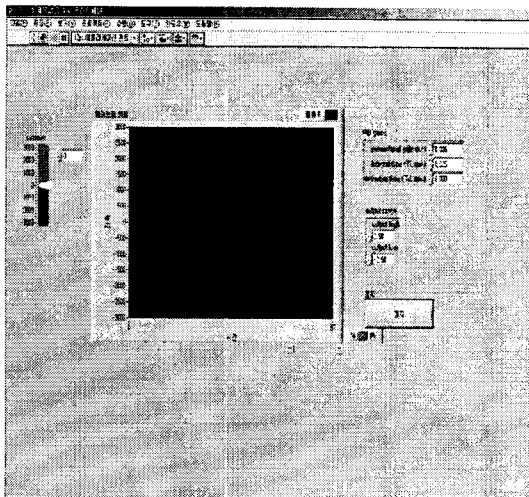


그림6. LabVIEW 프로그램의 프론트 패널

그림6은 LabVIEW 프로그램의 프론트 패널이다. 속도 지령을 인가하는 컨트롤 팔레트 및 속도 지령과 응답을 비교하여 볼 수 있는 그래프, 속도제어기 출력에서의 Limit값 설정 및 PID 제어기의 계인을 설정 할 수 있는 부분으로 이루어져 있다.

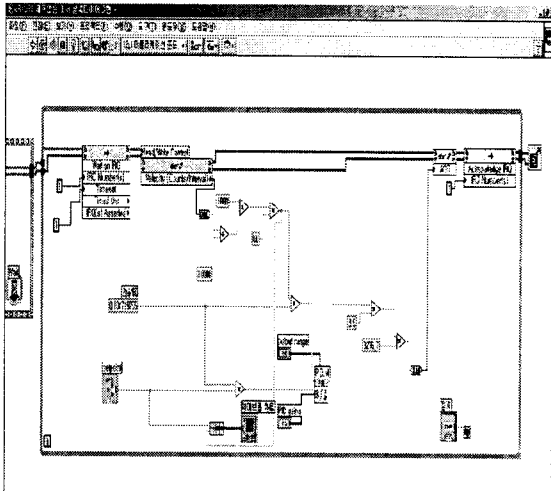


그림7. LabVIEW 프로그램의 블록다이어그램

그림7은 LabVIEW 프로그램의 블록다이어그램 화면을 나타내고 있다. FPGA 프로그램의 설정과 데이터를 받기 위한 블록들이 존재하고, 속도신호와 Analog output 신호인 토크지령을 스케일링하기 위한 부분이 있으며, SubVI 형태의 PID 제어기가 있다.

### 2.3.2 실험 결과

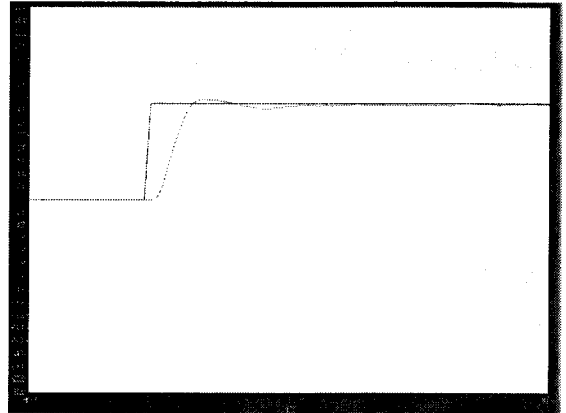


그림8. 속도응답[1500rpm속도지령시]

그림8은 속도지령을 1500(rpm)으로 주었을 때의 프론트 패널에서 AC 서보시스템의 속도지령 응답과형이다. 계인의 실시간 변경으로 속도제어기의 과도응답 및 정상상태응답을 제어목적에 맞게 얻을 수 있다.

### 3. 결 론

본 논문에서는 LabVIEW 프로그램과 cRIO(Compact RIO) 하드웨어를 이용하여 AC 서보시스템의 속도제어를 하였다. LabVIEW는 GUI방식의 프로그래밍 언어로 사용자가 시스템을 구현하기 용이하고, cRIO의 구성은 간단하여 제어시스템을 이해하고 구현하는데 용이하여 제어기술을 습득하는데 좋은 도구가 될 것이고, 실제 시스템을 설계할 때에도 개발시간이 단축될 것으로 기대된다.

### [참 고 문 헌]

- [1] 곽두영, "LabVIEW Express", Ohm사, p384-390, 2003
- [2] National Instruments, "LabVIEW FPGA Module User Manual", 2004
- [3] 이병욱, "LabVIEW를 이용한 기초과학실험", Ohm사, p179-201, 2004
- [4] 삼성, "CSDJ PLUS SERIES Servo Drive 사용자 매뉴얼", 1999
- [5] 삼성, "Smartjog 사용자 매뉴얼", 2004