

# LabVIEW의 모델기반 제어기 설계와 Compact RIO를 이용한 직류전동기 구동 시스템

지준근<sup>1\*</sup>

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

Jun-Keun Ji<sup>1\*</sup>

**요 약** 본 논문에서는 모델기반의 제어기 설계 프로그램인 National Instruments(NI)사의 System Identification Toolkit 과 Control Design Toolkit, Simulation module을 사용하여 기존의 제어기 설계방식보다 쉽고 편리하게 제어기를 설계 할 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 출력신호 를 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 다운로드 하여 실제 전동기 구동시스템의 실험을 통해서 설계된 속도제어기의 응답 결과를 모의실험과 비교하여 검증하였다.

**Abstract** This paper presents a controller implementation using model based controller design programs-System Identification Toolkit, Control Design Toolkit, Simulation module. This method is easier and simpler than conventional controller design method. To implement speed control system of DC motor, a CompactRIO, Real-Time(RT) cntroller provided by NI(National Instruments), is used as hardware equipment. Firstly transfer function of DC motor drive system, which was a control target plant, can be acquired through System Identification Toolkit by using test input signal applied to motor and output signal from motor. And designing of pole-zero compensator satisfying desired control response performance through Control Design Toolkit, designed speed control response can be tested through Simulation Module. Finally LabVIEW program is converted to real-time program and downloaded to CompactRIO real-time controller. Through experimental results to real DC motor drive system, designed speed control response is compared to simulation results.

**Key Words** : LabVIEW and Compact RIO, Model Based Cotroller Design, DC Motor Drive System

### 1. 서론

오늘날 제어공학의 발전으로 많은 제어기술이 선보여 지고 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 불연속적 일수록 표현이 불가능해진다. 그리하여 복잡한 시스템의 동작 확인과 최상의 제어기 설계를

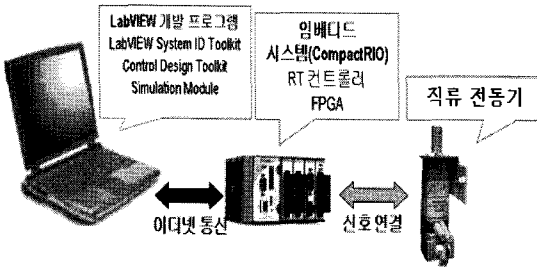
하는 것은 엔지니어에게 꼭 필요하다. 이러한 요구조건을 만족시키는 제어시스템 설계도구로서 그래픽기반 프로그램인 LabVIEW가 있고[1], 실시간 제어가능한 시스템으로서 NI사가 제공하는 CompactRIO(이하 cRIO)가 있다. 본 연구에서는 cRIO의 실시간 제어시스템과 LabVIEW 프로그램의 System Identification Toolkit과 Control Design Toolkit, Simulation Module을 사용하여 쉽고 빠르게 직류전동기 구동 시스템의 속도제어기 설계 및 실시간 실험을 하였다.

<sup>1</sup>순천향대학교 전기통신공학과

\*교신저자 : 지준근(jkji@sch.ac.kr)

## 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 구동시킴으로써 자동제어나 계측을 쉽게 원하는 대로 할 수 있다. 여기서 사용한 LabVIEW 모델기반 제어기 설계 프로그램에는 System Identification Toolkit과 Control Design Toolkit, Simulation Module이 있다.

### 2.2 CompactRIO 임베디드 시스템

cRIO는 전력 소모가 적은 실시간 임베디드 프로세서로 고성능의 FPGA 칩셋과 결합한 하드웨어이다. cRIO는 LabVIEW FPGA 요소 I/O 함수를 사용하는 각 I/O 모듈의 I/O 회로로 하드웨어를 집적 연결한다. 또한 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 사용되

는 직류 서보 전동기는 토크와 전류가 비례하여 선형제어 시스템을 구성하는 것이 가능하므로 비교적 간단하게 안정된 제어기 설계가 가능하다. 본 연구에서 사용한 직류 서보 전동기의 정격사양은 표 1과 같다.

[표 1] 직류 서보 전동기의 정격사양

직류 서보 전동기	정격출력	15W
	정격전압	12V
	최대속도	3000RPM
엔코더	500Pulses/Revolution	
	Open Collector 방식	
기어박스	기어비 1:20	Panasonic : M6GA20M

## 3. 시스템의 하드웨어 구성

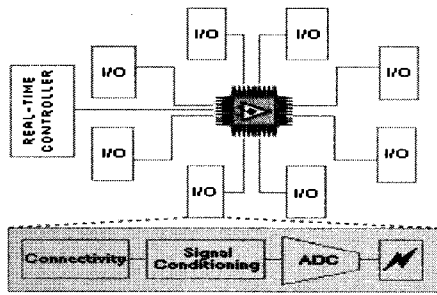
### 3.1 전동기 구동부

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

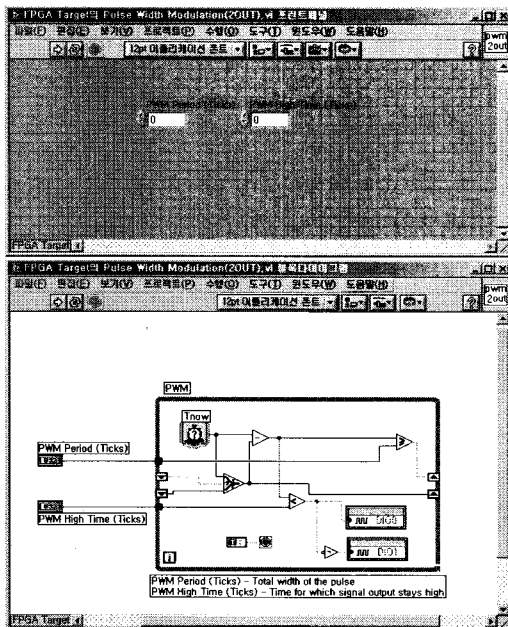
FPGA는 실리콘칩으로서 독립적인 로직게이트로 되어 있다. FPGA의 기능은 규정되어진 소프트웨어의 FPGA 게이트를 구성할 수 있다는 점이다. 이점은 어플리케이션에 따라 필드를 재구성할 수 있다. 본 연구에서는 NI사에서 제공해 주는 cRIO의 FPGA를 사용하였다.[2]

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



[그림 2] CompactRIO의 FPGA 구성

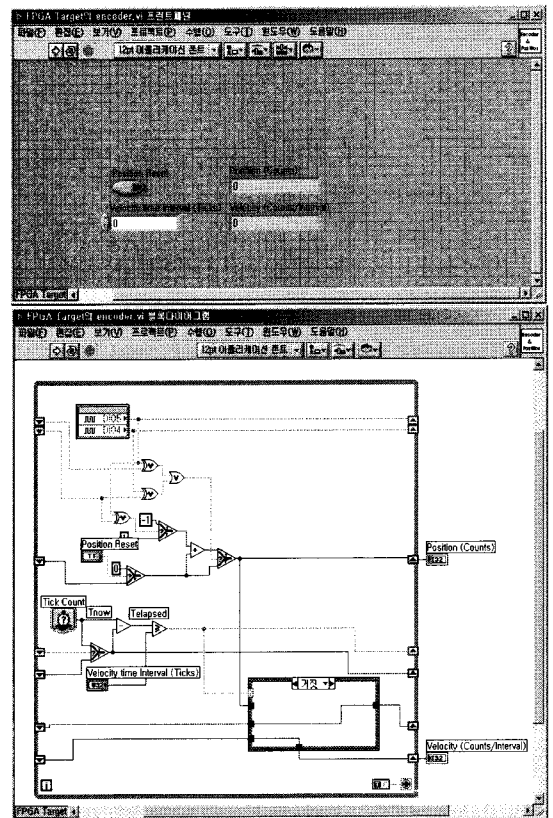
cRIO는 전력 소모가 적은 실시간 임베디드 프로세서를 고성능의 FPGA 칩셋과 결합한 하드웨어이다. 각 CompactRIO의 각 I/O 모듈은 내장형 신호 컨디셔닝 및 스크류 터미널, BNC 또는 D-Sub 커넥터를 포함하여 다음과 같은 다양한 I/O 유형이 사용 가능하다. (± 80mV 열전대 입력, ±10 V 동시-샘플링 아날로그 I/O, 최대 1A 전류 드라이브의 24 V 산업 디지털 I/O, 엔코더용 5 V 조절 공급 출력의 차동/TTL 디지털 입력 및 250 V(rms) 유니버설 디지털 출력 등이 있다.) 모듈은 확장 전압 범위 또는 산업 신호 유형을 위한 내장형 신호 컨디셔닝을 포함하므로 CompactRIO 모듈에서 센서 및 액추에이터로 직접 와이어를 연결할 수 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 3] PWM 프로그램

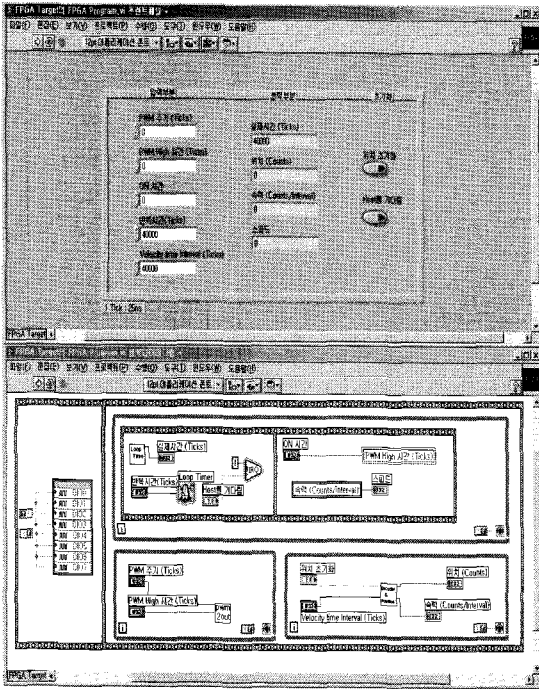
그림 3은 FPGA에 PWM을 프로그램한 것으로 NI-9401 모듈의 2채널을 사용하여 PWM을 프로그램하였다. 입력으로 상승구간과 PWM의 주기를 받았고 출력으로 2개의 상반된 PWM 신호를 출력하였다. 그림 4는 직류 전동기의 속도검출을 위해 엔코더의 펄스를 받아 위치값과 속도값을 구하는 FPGA 프로그램이다. 엔코더의 속도 검출방식은 펄스 카운트 방식을 사용하였고 A상, B상을 받아 4채배를 한 후에 1ms동안 카운트를 하였다.

$$\frac{\text{카운트펄스수}}{\text{단위시간}} \times \frac{1}{1\text{회전당 엔코더 출력펄스수}} = \text{회전속도}$$



[그림 4] Encoder 프로그램

그림 5는 전체적인 FPGA 프로그램으로서 LabVIEW로 프로그램된 FPGA를 볼 수 있다. 이러한 프로그램은 VHDL로 컴파일하여 CompactRIO에 다운로드 된다. NI-9401 모듈에 재구성된 FPGA는 LabVIEW 프로그램 및 CompactRIO의 실시간(Real-Time) 시스템과 연동하여 실시간으로 제어를 할 수 있게 한다.



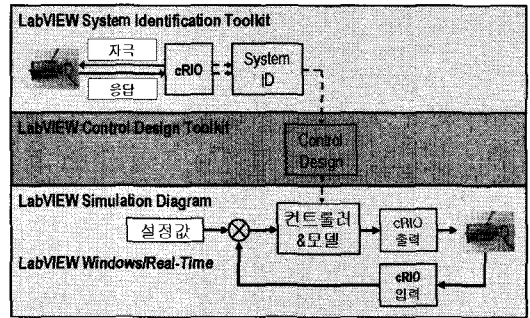
[그림 5] FPGA 프로그램

### 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-Time 시스템은 높은 신뢰성을 가지고 어플리케이션의 완성도를 높인다. 본 연구에서는 실시간 시스템을 사용하여 직류 전동기를 실시간으로 제어 하였고, 실시간으로 모의실험과 비교할 수 있었다.

## 4. LabVIEW 프로그래밍

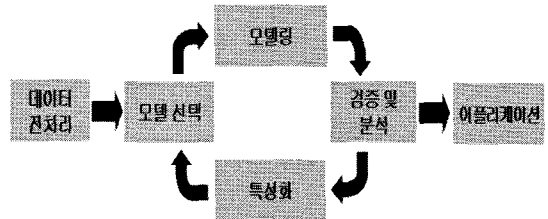
그림 6은 LabVIEW 프로그램의 개요를 나타낸 것이다. 본 연구에서는 직류 전동기에 자극을 주고 그에 대한 응답을 받아 시스템 식별(System Identification; 이하 ID)을 하였고 System ID로 추정하여 찾은 모델을 Control Design Toolkit를 사용하여 제어기를 설계한다. 설계된 제어기를 Simulation Module을 사용하여 시뮬레이션한 결과와 실제 직류 전동기의 실험결과를 비교 검증한다.



[그림 6] LabVIEW 프로그램의 개요

### 4.1 시스템 식별(System Identification)

그림 7은 System ID의 과정을 보여주고 있다. System ID를 하기 위해 자극을 주고 받는 응답이 있어야 한다. 이 응답으로 추정된 모델의 모델식을 만들고 만들어진 모델식은 시뮬레이션을 통해 검증과 분석을 한다.[3]



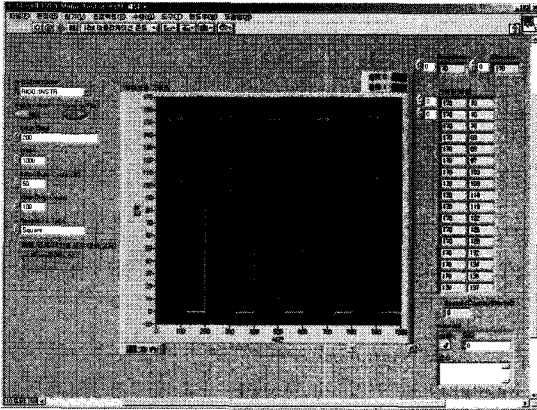
[그림 7] 시스템 ID의 단계

#### 4.1.1 데이터 전처리

전동기를 테스트하여 자극에 대한 응답값을 받기 위해 전동기에 0과 12V의 입력값을 1초의 주기로 200번 반복하여 자극을 주었고 그에 대한 응답값을 보았다. 이 자극에 대한 응답값들은 배열로 sysIDdata.txt의 이름으로 파일로 저장하였다. 저장한 sysIDdata.txt 파일은 뒤에 System ID를 하기 위해 사용된다. 표 2는 자극을 위해서 생성된 신호의 설정을 나타낸다.

[표 2] 생성된 신호의 설정

Step	Setting / Action
Signal Input = Encoder Signal Output = PWM	Signal Type = Square
	Frequency = 500Hz
	Amplitude = 170
	Sample Rate = 500s/s

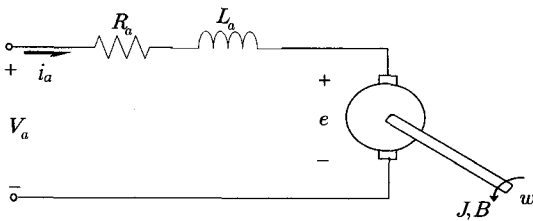


[그림 8] 전동기 테스트 결과

그림 8을 보면 얇은 흰색 실선은 자극값이고, 굵은 적색 점선은 자극에 대한 응답값이다. 오른쪽은 시간에 따른 자극과 응답값을 배열로 저장한 것을 보여 준 것이다. 이 배열을 파일로 저장하여 다음 단계의 시스템 ID를 한다.

4.1.2 모델 선택과 모델링

4.1.1절에서 구한 자극에 대한 응답 값을 가지고 System ID를 하기 위해 우선적으로 모델에 대한 선택이 이루어져야 한다. 모델에 대한 선택을 하기 위해서는 직류 전동기 구동 시스템의 오더(차수)의 범위를 알아야 할 필요가 있다. 계자전압이 일정하다고 가정했을 때의 직류 전동기의 등가회로는 그림 9와 같이 나타낼 수 있고, 전기자 전압을 제어하는 직류전동기의 속도는 전기자 전압  $V_a$ 에 의해서 제어된다.



[그림 9] 직류전동기의 등가 회로

- $V_a$  전기자 단자전압(Terminal Voltage) [V]
- $i_a$  전기자 전류 [A]
- $e$  역기전력 [V]
- $R_a$  전기자 권선의 등가저항 [Ω]
- $L_a$  전기자 권선의 등기인덕턴스 [H]

- $w$  전기자의 회전각속도 [rad/s]
- $K_e$  역기전력상수 [V/(rad/sec)]
- $K_T$  토크상수 [nt-m/A]
- $T_e$  회전력 [nt-m]
- $T_L$  부하토크 [nt-m]
- $J$  관성 [kg-m<sup>2</sup>]
- $B$  마찰계수 [nt-m/(rad/sec)]

그림 9에 나타난 회로에 키르히호프 전압법칙을 적용하면 다음과 같은 미분방정식을 얻을 수 있다.

$$V_a = R_a i_a + L_a \frac{di_a}{dt} + e \tag{1}$$

계자전류가 일정하면 공극자속이 일정하게 되어, 결과적으로 토크는 다음 식과 같이 전기자 전류  $i_a$ 에만 비례하게 된다.

$$T_e = K_T \cdot i_a \tag{2}$$

전기자 전류에 의하여 발생된 토크가 관성과 마찰을 포함하고 있는 회전체 시스템에 가해지면 다음과 같은 수식이 성립된다.

$$T_e = J \frac{dw}{dt} + Bw + T_L \tag{3}$$

전기자가 회전할 때, 자속과 각속도의 곱에 비례하는 전압이 전기자에 유도된다. 자속이 일정한 경우, 유도된 전압  $e$ 는 각속도  $w$ (rad/s)에 비례한다.

$$e = K_e \cdot w \tag{4}$$

직류전동기의 입력변수는  $V_a$  와  $T_L$  이고 상태변수는  $i_a$  와  $w$  이다. 식 (1) 과 (3)를 상태변수로 다시 정리하면 다음과 같은 수식이 성립된다.

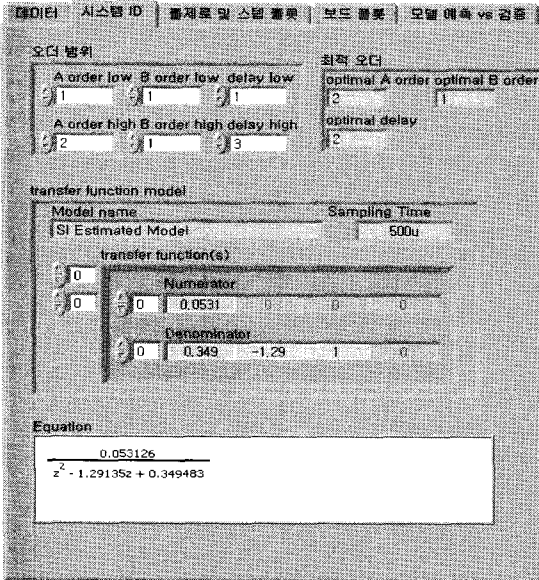
$$\frac{di_a}{dt} = \frac{1}{L_a} (V_a - R_a i_a - e) \tag{5}$$

$$\frac{dw}{dt} = \frac{1}{J} (T_e - Bw - T_L) \tag{6}$$

위의 식들로부터 상태방정식을 구하면 다음과 같다.

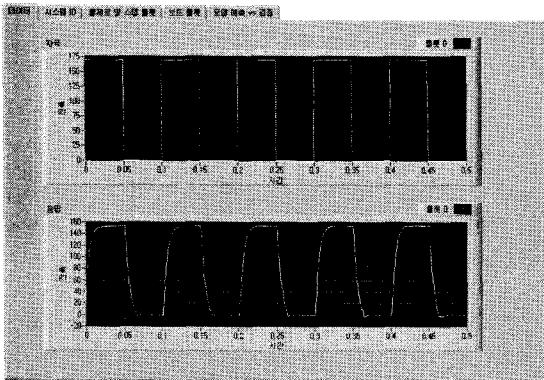
$$\begin{pmatrix} \frac{di_a}{dt} \\ \frac{dw}{dt} \end{pmatrix} = \begin{pmatrix} -\frac{R_a}{L_a} & -\frac{K_e}{L_a} \\ -\frac{B}{J} & \frac{K_T}{J} \end{pmatrix} \begin{pmatrix} i_a \\ w \end{pmatrix} + \begin{pmatrix} \frac{1}{L_a} \\ 0 \end{pmatrix} V_a + \begin{pmatrix} 0 \\ \frac{1}{J} \end{pmatrix} T_L \tag{7}$$

위와 같은 직류 전동기의 모델식으로부터, A 오더(차수)범위가 2차이고 B 오더(차수)범위가 1차라는 것을 알 수 있다. 그림 10을 보면 A, B 오더(차수)의 범위를 정해 주고 그에 따라 모델의 전달함수가 생성되는 것을 볼 수 있다.



[그림 10] 시스템 ID 결과

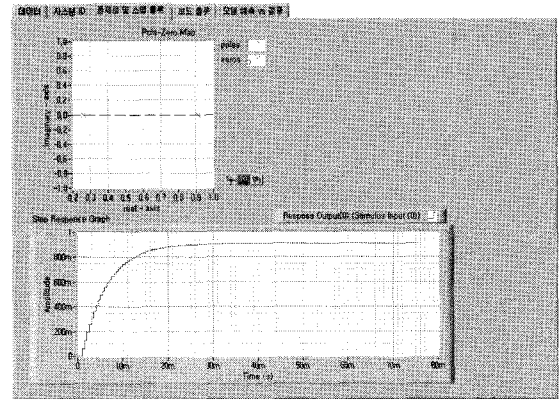
### 4.1.3 모델 검증 및 분석



[그림 11] 직류 전동기에 대한 자극과 응답

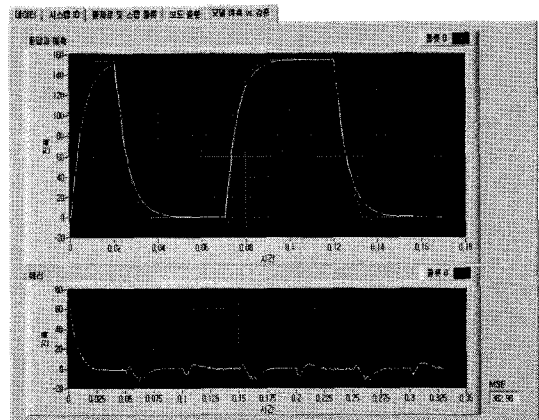
4.1.2절에서 구해진 전달함수가 실제 모델과 같은지 검증하기 위해 자극에 대한 응답을 확인해 본다. 4.1.1절에 구한 데이터를 그림 11에서 그래프로 나타내어 보았다. 위의 결과는 앞의 그림 8의 전동기 테스트 결과를 그래프로 나누어 그려본 것이다.

그림 12는 System ID로 구한 전달 함수에 대한 극점과 영점 그래프와 단위시스템응답 그래프이다. 스텝응답을 보면 기준입력인 1까지 도달하지 못하고 900m정도에 머무는 것을 볼 수 있다.



[그림 12] 직류전동기 전달함수에 대한 극점과 영점 및 단위시스템응답 그래프

그림 13의 직류 전동기의 응답과 전달함수의 검증에서는 System ID로 구한 전달함수와 실제 응답에 대해 비교해 보았다. 실제 응답과 거의 비슷한 결과를 보여주고 있는 것을 볼 수 있다. 이와 같이 System ID로 구한 전달함수가 실제 플랜트인 직류 전동기의 실험결과 같은 결과를 보여 준다는 것을 알 수 있었다. 검증을 마친 직류 전동기의 전달 함수를 sysID.LTI 파일로 저장하였다. 이는 나중에 Simulation Module에서 시뮬레이션을 위한 데이터파일로 쓰인다.

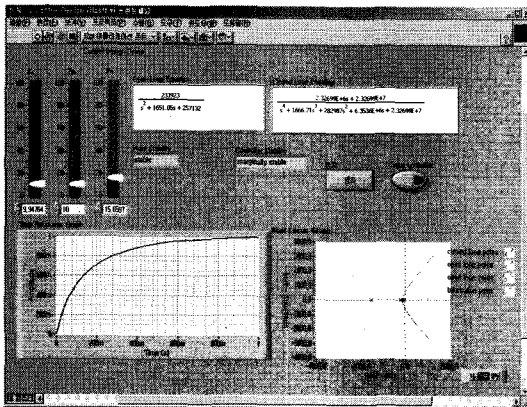


[그림 13] 직류 전동기의 응답과 전달함수의 검증

### 4.2 제어기 설계(Control Design)

시스템 식별에서 모델의 전달함수를 구함으로써 상승 시간(rise time), 정착시간(Setting time), 오버슈트(Overshoot)에 대한 요구조건에 맞는 제어기를 설계할 수 있게 되었다. 그림 14에서 보여지는 Control Design 프로그램의 프러نت 패널은 종속제어기를 플랜트 앞에 놓는 방식에 따라서 페루프시스템의 동작이 어떻게 결정되는 지를 보여준다.[4]

그림 14의 왼쪽 상단에 있는 비례 이득, 영점, 극점 조절 게이지로 제어기를 설계한다. 이러한 방식을 비례 제어, 극배치 방법 중 극점영점상쇄(pole-zero cancellation)라고 한다. 비례 제어란 기준신호와 되먹임신호 사이의 차인 오차신호에 적당한 비례상수 이득을 곱해서 제어신호를 만들어내는 제어기법으로 이 기법에 의한 제어기를 비례 제어기(proportional controller), 또는 영문약자를 써서 P 제어기라 부른다. 극점영점상쇄(pole-zero cancellation)이란 제어기의 영점을 플랜트의 극점과 상쇄되도록 극점과 영점을 재배치하여 페루프시스템의 극점들을 원하는 위치에 적절히 설정하는 방법이다. 극점, 영점의 위치와 시스템 성능과의 관계를 고려하여 페루프시스템 전달함수의 극점과 영점의 위치를 적절히 지정하고 이 위치에 페루프시스템의 극점, 영점이 놓이도록 제어기를 설계함으로써 원하는 성능목표를 이루는 제어기 설계 방식이다. 이 같은 제어기의 단점은 비례 제어기의 상태 개선의 어려움과 낮은 차수의 전달함수의 형태로 표현되는 시스템에서만 극배치 설계방식이 적용될 수 있다는 것이다. 위의 시스템을 2차의 전달함수를 갖는 시스템으로써 계산하였고 좀 더 쉽게 제어기를 설계하기 위하여 극점영점상쇄 방법을 사용하였다.

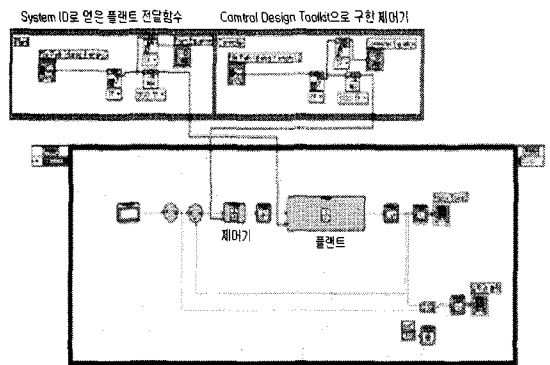


[그림 14] Control Design 프로그램의 프러نت 패널

그림 14에서 설계한 제어기의 전달함수  $\frac{9.94764(s+10)}{s^2 + 15.6597s}$ 와 플랜트의 개루프 전달함수로부터 페루프 전달함수를 구하면 그림에서와 같게 된다. 그림 13의 왼쪽 하단을 보면 페루프 시스템에 대한 단위시스템 응답은 정착 시간이 1초이고 오버슈트(overshoot)는 없다. 또한 그림 14의 오른쪽 하단을 보면 개루프 시스템의 극점과 영점의 위치를 볼 수 있고, 페루프 시스템일 때는 좌반평면에 극점과 영점이 위치하고 모두 실수축에 존재한다는 것을 알 수 있다. 이와 같이 설계된 제어기는 conD.LTI 파일로 저장 하였다. 이는 Simulation Module에서 시뮬레이션을 위해 사용되어진다.

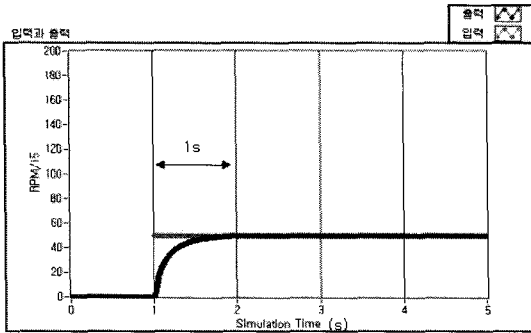
### 5. 시뮬레이션 및 실험결과

System ID를 사용하여 구해진 전동기의 전달함수와 Control Design을 사용하여 구해진 제어기를 가지고 Simulation Module을 사용하여 시뮬레이션을 하여 보았다[6]. 시뮬레이션을 하기 위하여 전동기 전달함수를 저장한 sysID.LTI 파일과 제어기를 저장한 conD.LTI 파일을 불러와서 시뮬레이션 모듈의 시뮬레이션 루프에 넣고 시뮬레이션을 하였다. 그림 15는 시뮬레이션 프로그램을 보여준다.



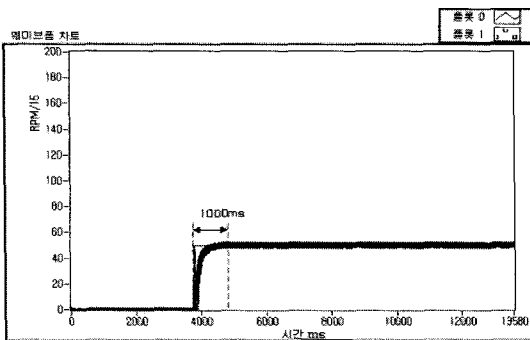
[그림 15] 시뮬레이션 프로그램

그림 16은 시뮬레이션의 결과이다. 결과를 보면 설정 값 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 16] 시뮬레이션 결과 그래프

그림 17은 전동기를 실제로 구동했을 때의 실험 결과이다. 시뮬레이션에서처럼 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 17] 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 Control Design을 이용할 경우 기

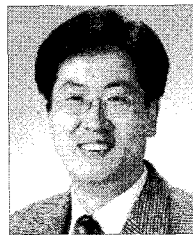
존의 제어시스템 개발시보다 제어기의 설계 및 구현이 매우 쉽고 편리하며, 응답결과도 확인이 편리하여 제어시스템 해석 및 설계를 쉽게 이해하고 빠른 시간에 제어 시스템을 설계할 수 있었다.

## 참고문헌

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

지 준 근(Jun-Keun Ji)

[정회원]



- 1986년 2월 : 서울대학교 전기공학과 (공학사)
- 1988년 2월 : 서울대학교 대학원 전기공학과 (공학석사)
- 1994년 2월 : 서울대학교 대학원 전기공학과 (공학박사)
- 1994년 9월 ~ 현재 : 순천향대학교 전기통신공학과 교수

<관심분야>

전력전자, 전기기계, 전력변환, 전동기 제어, 전원장치, 전력품질제어, 모션제어, 제어응용