

특집 : 전력전자 교육의 현황 및 방향

전력전자공학의 실험실습 교육을 위한 소프트웨어 및 하드웨어 시스템 소개

지 준 근

(순천향대 전기통신시스템공학과 교수)

1. 서론

오랫동안 학부생들의 전력전자공학 과목의 교육을 통해 경험한 바는 학생들의 교육성취도를 높여주고 내용에 대한 충실한 이해 및 활용을 극대화하기 위해서는 이론적인 수업 이외에 컴퓨터 시뮬레이션 및 효과적인 실험실습이 필요하다는 것이다. 이에 따라 저자가 학부 4학년생들을 대상으로 교육하고 있는 방법은 이론적인 교재에 의한 3시간의 강의시간 외에 전력전자 전용 시뮬레이션 프로그램인 PSIM과 Lab Volt사의 전력전자공학 실험실습 기자재인 "Power Electronics Training System" 및 "Data Acquisition & Management System"을 사용하여 수업시간에 배운 내용들을 학생들이 자율적인 실험실습 교육을 통하여 교과목의 내용에 대한 완벽한 이해는 물론 수업을 통해서 확인할 수 없었던 실제적인 현상들의 경험 및 기술 습득을 하도록 하는 것이다. 또한 이를 기초로 하여 4학년들의 졸업논문 주제로 전력전자공학 분야 중에서 전력변환장치와 DSP를 이용한 전동기 제어시스템을 구현하고 프로그래밍을 통한 실험을 하도록 하여 실제적인 H/W 및 S/W 기술을 습득하도록 하고 있다. 여기에서는 전력전자공학의 실험실습 교육을 위해 저자가 사용하고 있는 소프트웨어 및 하드웨어 시스템의 몇 가지 사례를 소개하고자 한다.

2. CEMTool/SIMTool과 고속설계제어기시스템

먼저는 (주)리얼게인의 DSP를 사용하는 "고속설계제어기

시스템"에 의한 AC 서보 모터의 속도관측기 및 속도 제어기 설계에 대하여 소개한다. 고속설계제어기시스템은 CEMTool/SIMTool 소프트웨어와 RG-DSPIO 제어 보드, AUTOTool 프로그램으로 구성되어 있다. SIMTool 블록들을 사용함으로써 짧은 시간동안에 다양한 형태의 제어기를 설계 및 구현할 수 있도록 해주는 장점이 있기 때문에 AC 서보 모터의 제어 목적에 따라서 속도관측기 및 속도제어기를 쉽게 설계하고 구현할 수 있다.

2.1 고속설계제어기(RG-01D)의 개요

고속설계제어기란 전에는 제어 시스템 개발을 각각의 하드웨어에 맞추어 C언어를 이용하여 큰 프로그램을 장기간에 걸쳐 개발을 진행하여 왔지만, 이러한 프로그램을 간단히 블록 다이어그램이나 명령어 방식으로 제어기 등을 설계하여 바로 실시간으로 외부 입력, 출력장치와 연결하여 실제 실험을 할 수 있는 장치이다.

RG-01D 모델은 이러한 기능을 수행하는 고속설계제어기 중의 하나로서 CEMTool과 블록 알고리즘을 구성할 수 있는 SIMTool 및 블록으로 구성된 알고리즘을 자동으로 C코드로 생성하여 주는 C코드 생성기인 AUTOTool을 포함한 소프트웨어와, 고속 연산을 수행할 수 있는 DSP 칩셋을 사용하고 외부 입출력을 할 수 있는 I/O 기능이 있는 RG-DSPIO 보드로 구성되어 있어 가장 빠른 실시간 시스템을 구성할 수 있는 장점이 있다. 고속설계제어기(RG-01D)의 개념도는 그림 1과 같다.

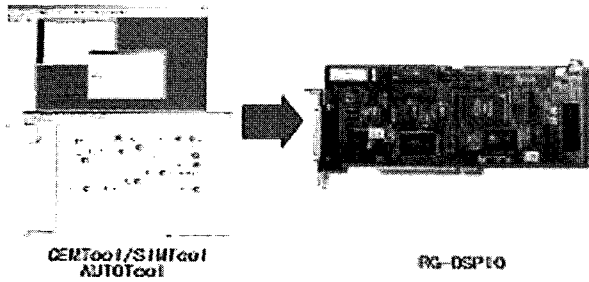


그림 1 고속설계제어기(RG-01D)의 개념도

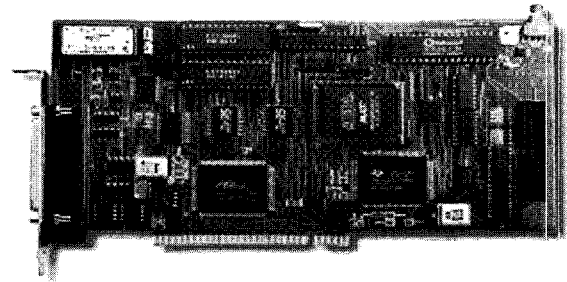


그림 3 RG-DSPIO Board

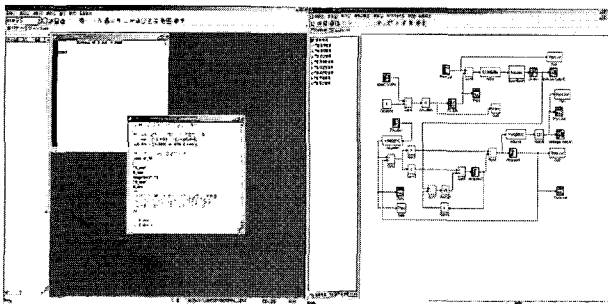


그림 2 CEMTool, SIMTool의 실행화면

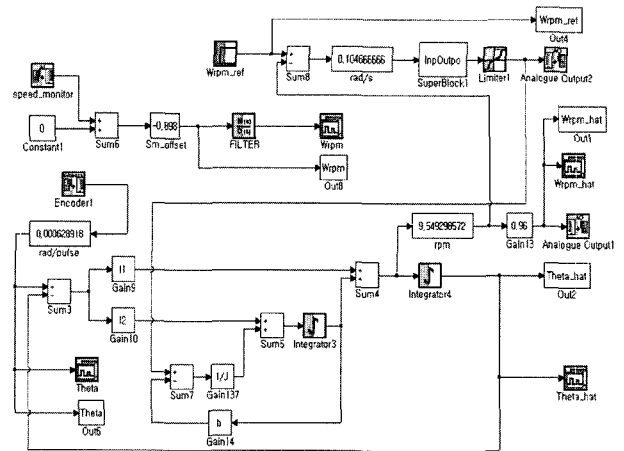


그림 4 속도관측기를 사용한 속도제어기 프로그램

2.2 CEMTool, SIMTool의 구성

CEMTool(Computer-aided Engineering & mathematics Tool)은 수학과 공학 계산을 수행하는 과학기술 범용 패키지이고, SIMTool (Simulation Tool)은 블록 다이어그램 프로그램 방식의 블록 구조 설계 패키지이다. CEMTool의 경우 CEMTool 상의 문법뿐만 아니라 제어용 소프트웨어로 가장 많이 사용되고 있는 MATLAB의 문법도 지원하므로 기존에 작성해 놓은 M-file을 사용할 수 있다. SIMTool에서는 시스템을 형성하는 각종 요소들을 기본적으로 제공되는 100개 이상의 블록들의 조합으로 나타내어 그 특성을 표현함으로써, 복잡한 시스템을 손쉽게 모델링하고 시뮬레이션 할 수 있게 한다. 그리고 선형요소 뿐만 아니라 기존의 프로그래밍 형태의 시뮬레이션 패키지에서는 나타내기 힘들었던 각종 비선형 요소들도 간단히 모델링 할 수 있다. SIMTool을 이용하면 연속 시간 시스템과 이산 시간 시스템 각각의 시뮬레이션은 물론이고, 두 요소가 섞여 있는 하이브리드 시스템의 시뮬레이션 또한 가능하다. 또한 고급기능으로 매우 복잡한 시스템을 간편하고 체계적으로 모델링하기 위한 슈퍼블록(Super block)과 매크로블록(Macro block) 기능이 있다.

2.3 RG-DSPIO 보드의 구성

그림 3은 고속설계제어기의 RG-DSPIO 보드의 모습이다. RG-DSPIO는 고속 데이터 연산을 위한 DSP 기능을 내장하

고 있다. DSP CPU는 TI사의 부동소수점 연산 전용 DSP 칩인 TM320C32를 사용하고, 60MHz의 clock으로 동작한다. 이 DSP 칩은 2개의 타이머, 4개의 사용자 인터럽트를 갖고 있으며 256word의 내부 메모리를 갖고 있어 시스템의 레지스터 등으로 사용이 가능하다.

Monitoring기능을 담고 있는 ROM은 128KB의 용량으로 사용자 프로그램의 다운로드, 실행 데이터 업로딩, 디버깅을 위한 함수 등을 담고 있다. 프로그램의 실행을 위한 메모리는 256KB의 용량으로 no-wait cycle의 고속 SRAM이다. 시스템과 사용자간의 데이터 교환은 PCI 버스를 통해 이루어지는데, PCI 인터페이스 칩과 DSP 사이에 고속 데이터 교환을 위해 16bit 버스를 이용하는 DPRAM(Dual Port RAM)을 장착하고 DPRAM의 인터럽트를 사용함으로써 데이터의 교환에 필요한 overhead를 최소화한다.

2.4 속도관측기를 사용한 속도제어기 실험

그림 4는 속도관측기를 사용한 속도제어기의 설계 및 응답을 위해서 구성한 SIMTool 상의 제어블록도이다. 그림 5는 속도관측기를 사용한 속도제어기의 응답과 속도제어시 관측

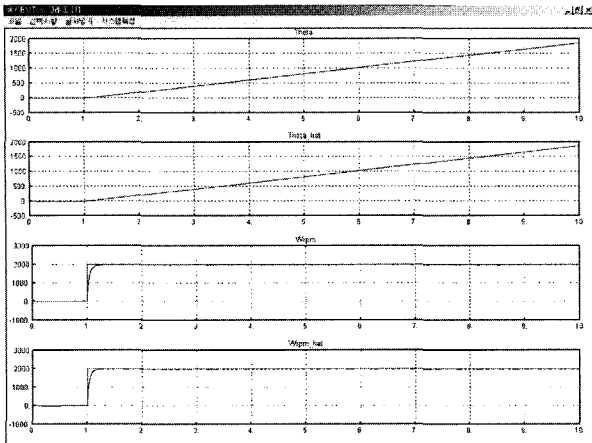


그림 5 속도관측기를 사용한 속도제어기 응답

기의 위치 및 속도 추정 응답을 보여주고 있으며 매우 양호한 것을 보여 준다. 무엇보다도 이러한 실험 모두를 DSP를 사용하는 고속설계제어기를 사용하여 짧은 시간 안에 매우 편리하게 구현할 수 있다.

지금까지 (주)리얼게인의 '고속설계제어기시스템'의 구성과 속도제어기 설계에 대한 활용 예를 살펴보았다. 기존에는 제어 시스템 개발을 위하여 시스템 모델을 바탕으로 시뮬레이션을 행하고 시스템의 특성을 확인한 후 각각의 하드웨어에 맞추어 C언어를 이용하여 큰 프로그램을 장기간에 걸쳐 개발을 진행하여 왔지만 '고속설계제어기시스템'을 이용한 경우 제어 시스템의 개발 속도가 상당히 향상되었다. 또한 제어 기술을 처음 습득하는 학생들에게 좋은 교육 자료가 될 것이다.

3. MATLAB/SIMULINK와 dSPACE 시스템

전동기제어는 일반적으로 먼저 시뮬레이션을 통해 제어기를 설계한 후, 다시 C나 어셈블리 언어로 프로그램을 작성하여 구현하였다. 그러나 dSPACE 1104 시스템과 MATLAB/SIMULINK를 이용하여 텍스트 방식이 아닌 그래픽 방식으로 제어기를 설계하고 시뮬레이션을 한 다음 직접 실시간 프로그램으로 자동 변환하여 제어기의 응답을 테스트함으로써 전동기 구동 시스템의 제어기 설계를 짧은 시간 안에 매우 편리하게 구현할 수 있음을 소개하고자 한다.

여기에서는 직류전동기의 속도 및 전류제어를 위하여 dSPACE 시스템을 이용하여 전류 궤환을 갖는 속도 제어 시스템을 구현하였다. 속도 및 전류제어기의 설계는 MATLAB/SIMULINK 프로그램을 사용하여 간편하고 손쉽게 구현하였으며 직류전동기 속도제어의 안정성과 응답성을 향상시킬 수 있었다. 직류전동기의 전류제어 및 속도제어는 DSP 보드

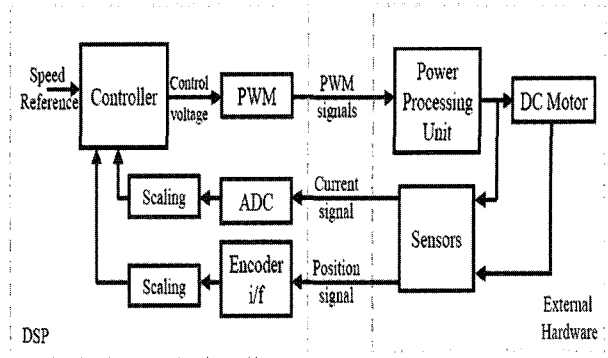


그림 6 전체 시스템 구성도

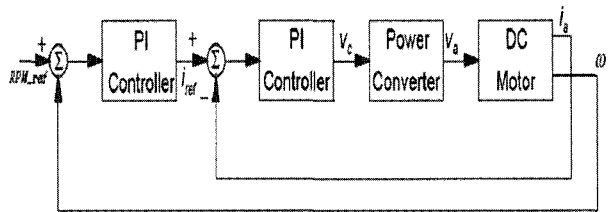


그림 7 제어기 구성도

와 dSPACE 시스템을 사용하여 수행하였으며, 속도의 궤환은 속도센서인 엔코더 펄스를 이용해서 QEP로 처리하였고 전류의 궤환은 전류센서인 홀센서를 통해서 A/D 변환기로 처리하였다. 제어기들은 각각 PI 속도제어기 및 PI 전류제어기를 설계하였고 시뮬레이션과 실험을 통해서 속도 및 전류 응답을 확인하였다.

3.1 전체 시스템 구성도

그림 6의 전체 시스템은 제어부인 DSP 보드와 전력변환장치 및 모터로 이루어져 있다. DSP 보드에서 전류 및 속도제어기가 프로그램으로 설계되어 있으며 전류센서를 통해 ADC로 전동기 전류값을 받고, 모터의 엔코더 신호를 QEP로 받아 제어기가 동작하여 최종출력은 PWM 신호로 내보낸다.

3.2 제어기 구성도

그림 7은 속도 및 전류제어기 구성도이며 바깥쪽이 속도제어기이고 안쪽이 전류제어기이다. 속도제어기 동작 타이밍을 전류제어기보다 1/10로 느리게 설정하여 전류제어기($f_{cc}=10\text{kHz}$)가 10번 동작할 때 속도제어기($f_{sc}=1\text{kHz}$)는 1번 동작한다.

3.3 MATLAB/SIMULINK 시뮬레이션 및 실험

그림 8은 시뮬레이션 제어블록이다. 실험을 하기 전에 먼저 실제 직류전동기를 모델링하여 시뮬레이션을 수행함으로써

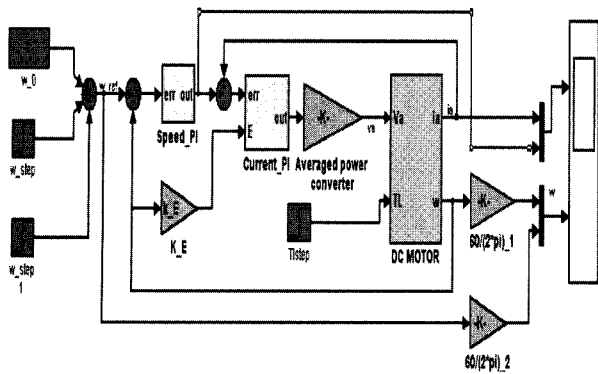


그림 8 시뮬레이션 제어블록

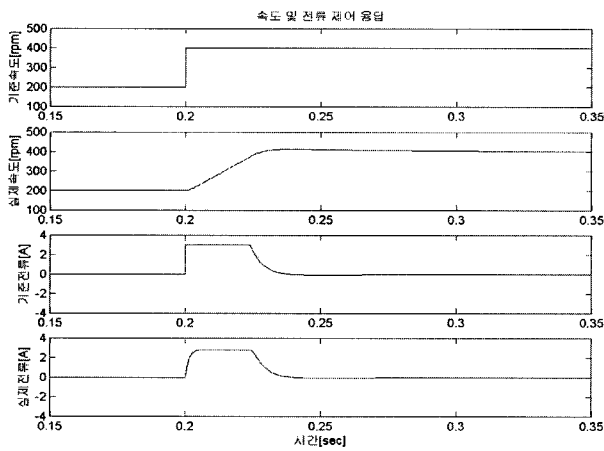


그림 9 속도 지령 가속시 스텝응답(200→400RPM)

실제 실험에서의 실수나 잘못된 제어기 설계 등을 피하고 보다 안전하게 실험을 할 수가 있었다. 그림 9는 속도지령 가속 시 시뮬레이션 결과 파형이다.

그림 10은 실험을 위한 직류전동기 속도 및 전류제어기 블록을 나타내며, 기본주기는 전류제어주기로서($T_{cc}=100\mu s$)로 동작하도록 설정되어 있다. 하지만 속도제어기는 이보다 1/10 정도 느려야 하기 때문에 인터럽트 블록을 만들어 전류제어기와 서로 다른 주파수($T_{sc}=1ms$)로 동작하도록 설계하였다. 제어기가 제대로 동작하는지 확인하기 위해 속도지령, 실제속도, 전류지령, 실제전류의 값들을 각각 DAC로 출력하였고, 각각의 실제 값들과 DAC 출력전압을 서로 맞추어 주기 위해서 DAC 앞단에 적절한 이득을 곱하여 속도는 0.01V/RPM이고 전류는 1V/A가 되도록 설정하였다. PI 전류제어기 설계이론을 바탕으로 제어기의 이득값들을 설정하여 전류제어기 설계의 타당성을 확인할 수 있었다.

그림 11의 전동기 속도지령 가속시 실험결과 파형에서, 각 채널은 1:속도지령, 2:실제속도, 3:전류지령, 4:실제전류를 나타낸다. 본 연구에 사용한 전동기는 그림 12에서 보이는

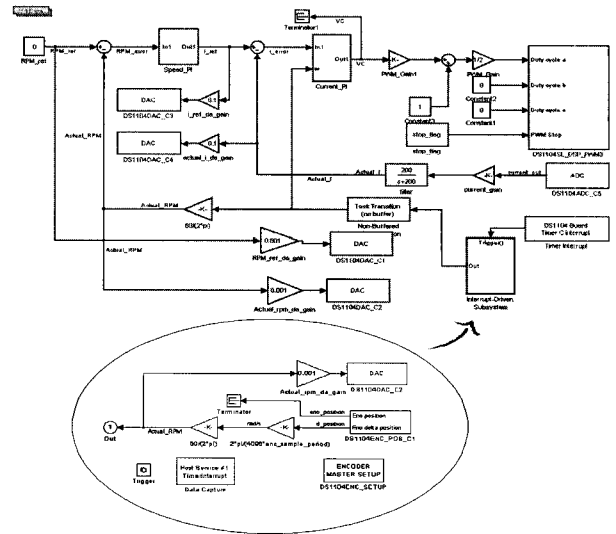


그림 10 직류전동기 제어블록

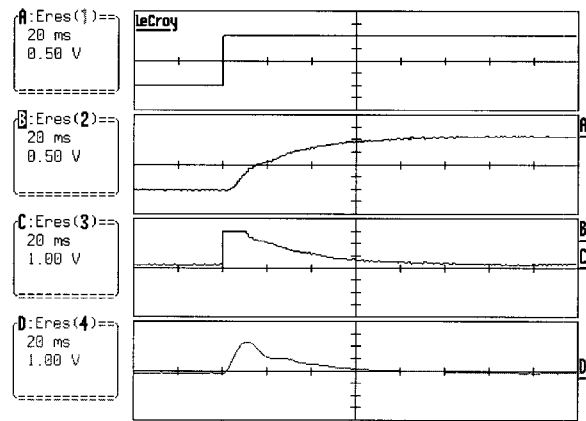


그림 11 속도 지령 가속시 스텝응답(200→400RPM)

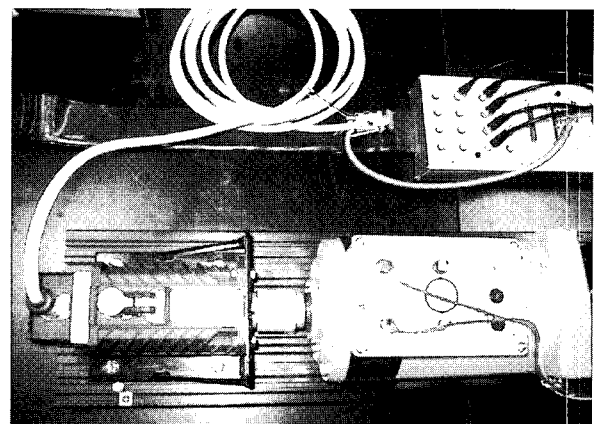


그림 12 실제 전동기 시스템 구성도

LeCroy Somer MG 세트의 컨버터용 타여자 직류 전동기이다. 속도센서인 엔코더가 직류전동기 옆에 있는 BLDC에 달려 있기 때문에 커플링으로 서로 연결하여 사용하는 과정에서 약간의 백래시 현상이 발생하여 스텝명령의 20ms이후 부근에서 진동이 생겨났다.

지금까지 dSPACE 1104 시스템을 사용하여 직류전동기의 속도 및 전류제어기를 구현하여 속도 스텝명령시 속도 및 전류 응답을 살펴보았다. dSPACE 1104 시스템과 MATLAB/SIMULINK 프로그램을 이용할 경우 기존의 제어시스템 개발 시보다 제어기 설계 및 구현이 매우 쉽고 응답결과도 보기에 편하여 제어시스템 해석 및 설계에 관해 쉽게 이해하고 단기간에 설계할 수 있다.

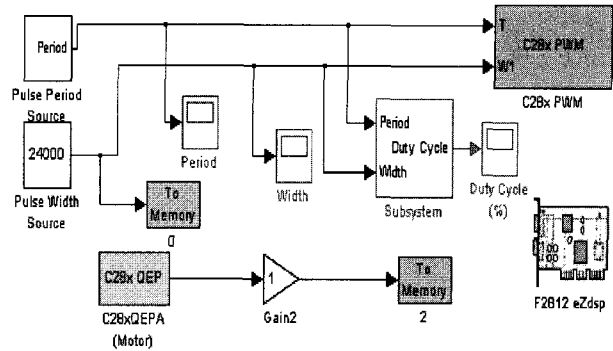


그림 13 PWM Duty Cycle Control via Period Change

4. MATLAB/SIMULINK의 Embedded Target 시스템

TI(Texas Instruments)사의 eZdsp F2811보드와 MATLAB/SIMULINK를 이용하여 텍스트 방식이 아닌 그래픽 방식으로 제어기를 설계하고 시뮬레이션을 한 다음 직접 실시간 프로그램으로 자동 변환하여 제어기의 응답을 테스트함으로써 전동기의 제어 시스템 개발을 짧은 시간 안에 매우 편리하게 구현할 수 있는 모델기반 제어기 설계 및 임베디드 제어시스템 구현을 소개하고자 한다.

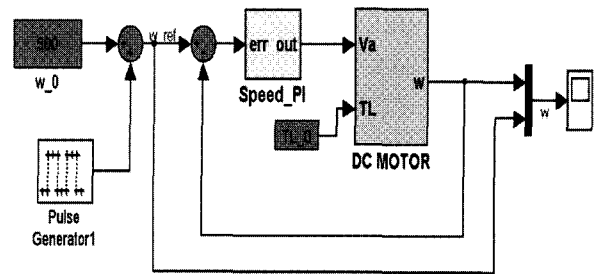


그림 14 시뮬레이션 프로그램

4.1 Embedded Target for TI C2000 DSP

“Embedded Target for TI TMS320C2000 DSP” 플랫폼은 MATLAB/SIMULINK를 TI의 eXpress DSP tool들과 통합한 것이며, 이는 디지털 신호처리와 제어기 설계에 있어서 개발자의 이론 및 개념을 코드 생성을 통한 검증 및 빠른 모형 제작을 가능토록 하였다. 이를 위해 이 플랫폼은 사용자의 SIMULINK Model의 C 언어 구현을 생성하기 위해 MATLAB의 Real-Time Workshop과 TI의 개발도구들을 이용한다. 이 때 Real-Time Workshop은 Code Composer Studio 프로젝트를 만드는데, 사용자는 생성된 프로젝트를 컴파일(Compile), 링크(Link) 및 타겟 보드에 다운로드(Download)를 할 수 있다.

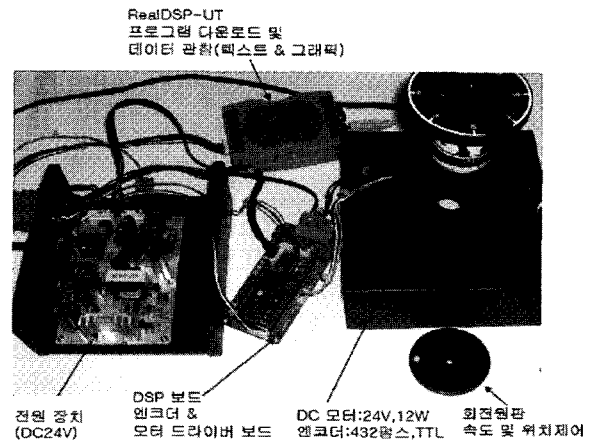


그림 15 eZdsp F2811 DSP 보드와 실험장치 구성

4.2 MATLAB/SIMULINK의 모델기반 설계에

그림 13은 Duty Cycle의 규칙적인 변화를 가진 펄스 파형에 따른 Embedded Target for TI C2000 DSP 플랫폼의 C28x QEP 블록과 C28x PWM 블록의 사용예이다.

QEP 블록은 전동기에 연결된 광학식 엔코더의 구형파 펄스 출력을 받아들이어 전동기의 위치와 회전속도를 계산한다. PWM 블록은 전동기에 인가되는 전압을 PWM 방식으로 조절한다.

4.3 MATLAB/SIMULINK 시뮬레이션 및 실험

실제 실험에서 보다 정확하고 안전한 실험을 위하여 실험 전에 미리 MATLAB/SIMULINK로 직류전동기 모델과 PI 속도제어기를 사용하여 그림 14와 같은 시뮬레이션 프로그램을 수행하면서 속도제어기의 이득을 설계할 수 있다.

그림 15의 시스템 하드웨어 구성은 직류전동기의 속도 측정부, 측정된 속도값을 읽어 속도제어 및 제어기 출력을 발생

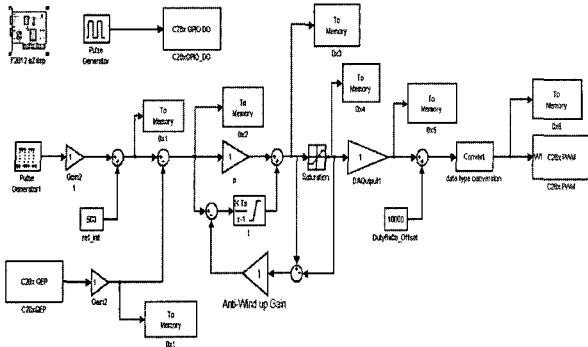
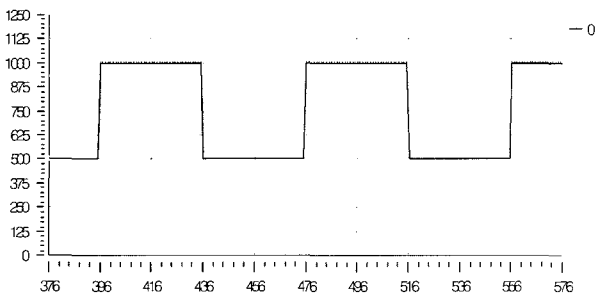
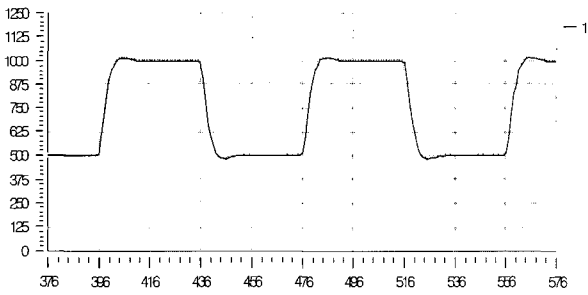


그림 16 eZdsp F2811 DSP 보드를 위한 실시간 프로그램



(a)



(b)

그림 17 PI 속도 제어 후 응답 (a) 기준속도 (b) 실제속도

하는 제어부, 그리고 제어기 출력에 따라 전동기를 구동시키기 위한 구동회로부 및 전동기로 구성된다. 시뮬레이션 이후에 그림 16과 같은 실시간 제어 프로그램을 작성하고 컴파일과 링크를 수행한 후, CCS(Code Composer Studio) v3.1 프로그램을 사용하여 실행파일인 COFF 파일을 생성하였다. 그리고 (주)리얼시스템에서 제공하는 Real DSP DataMON 프로그램을 통해 TMS320F28X의 SCI 부트 기능을 사용하여 프로그램을 DSP 내부 플래시 롬에 다운로드 할 수 있었고 변수 및 메모리 내용의 확인 또는 그래프 표시가 가능하였다.

펄스발생기를 사용하여 기준 속도를 500[rpm]에서 1000[rpm]으로 5초 주기로 변경하였고, 전동기의 속도와 위

치정보를 얻는데 유용한 C28x QEP 블록을 사용하여 실제 속도를 측정할 수 있었다. 이 때 QEP의 Counting mode는 RPM으로 설정하여 분당 전동기의 회전수를 측정하였으며, 사용한 엔코더의 분해능은 432(P/R)서 4채배를 하여 속도를 측정하였다. 기준 속도와 실제 속도의 차이가 PI 속도제어기의 입력으로 들어가게 되고 속도 제어 후 출력된 데이터는 최종적으로 데이터 타입 변환 블록을 거쳐 C28x PWM 블록으로 입력된다.

PI 속도제어시 Anti-Windup 블록이 추가 되었는데, 이는 속도 제어기에 적분기가 포함되어 있을 경우 적분기의 적분값이 제어기의 출력 값에 제한되지 않고 제어기의 제한폭을 넘어 쌓이게(wind-up)되는 현상을 방지하기 위해서이다.

F2811 DSP는 다양한 신호를 발생시키기 위해 사용되는 PWM을 포함한다. PWM 블록을 통해 전동기의 속도를 가변시키는 PWM 신호의 파형을 하드웨어적으로 구성하지 않고도 발생시킬 수 있다.

그림 17은 실험 결과 기준속도와 실제속도를 보여주는 PI 속도제어기의 응답 파형이다. 그림에서 볼 수 있듯이, PI 속도제어를 통해서 속도명령의 스텝변경시 정상상태 오차가 없고 빠른 응답속도를 갖는 제어기를 설계할 수 있음을 확인할 수 있다.

여기에서는 “Embedded Target for TI C2000 DSP” 플랫폼을 이용하여 직류전동기의 속도 제어를 구현하고 속도 스텝 명령시 속도 응답을 살펴 보았다. 기준 속도 명령에 대해 거의 0에 가까운 정상 상태 오차를 가진 응답의 결과를 확인할 수 있었고, 무엇보다도 “Embedded Target for TI C2000 DSP” 플랫폼을 이용할 경우 기존의 제어 시스템 개발시 보다 제어기의 설계 및 구현이 매우 쉽고 응답 결과의 확인이 편리하다. 따라서 이를 통해 제어시스템 해석 및 설계에 관하여 쉽게 이해하고 단기간에 설계할 수 있다.

5. 결론

전력전자공학의 실험실습 교육을 위해 저자가 사용하고 있는 소프트웨어 및 하드웨어 시스템의 몇 가지 사례를 소개하였다. 여기서 소개된 사례들은 학생들로 하여금 전력전자공학 분야 중에서 특별히 전력변환장치와 DSP를 이용한 전동기 제어시스템을 구현하고 프로그래밍을 통한 실험을 하도록 하여 실제적인 H/W 및 S/W 기술을 습득하도록 하는 데 유용하다고 할 수 있다.

참고 문헌

- [1] 리얼계인(사), “고속설계제어기(RG-01D)활용”
- [2] 리얼계인(사), “CEMTool 5.0 사용자 안내서”

- [3] 리얼게인(사), "SIMTool 4.0 사용자 안내서"
- [4] 지준근, "DSP 사용 고속설계제어기에 의한 AC 서보시스템의 속도제어기 설계", 2004년도 전력전자학회 하계학술대회 논문집, 2004. 7, pp. 177-181.
- [5] 황재현, 이용석, 지준근, "고속설계시스템에 의한 직류 전동기 구동시스템의 부하변동에 강인한 속도제어", 2006년도 대한전기학회 하계학술대회 논문집, 2006. 7, pp. 1029-1030.
- [6] 지준근, 이용석, "dSPACE 시스템을 이용한 직류 전동기 구동 시스템의 제어기 설계", 2005년도 대한전기학회 하계학술대회 논문집, 2005. 7, pp. 1796-1798.
- [7] 지준근, 이동민, "MATLAB/SIMULINK와 dSPACE DS1104를 이용한 유도전동기의 간접벡터제어", 2006년도 대한전기학회 하계학술대회 논문집, 2006. 7, pp. 1022-1023.
- [8] (주)리얼시스, "MATLAB을 사용한 DSP28X 제어", 2005.
- [9] 최승필, 이용석, 지준근, "MATLAB/SIMULINK의 모델기반 제어기 설계를 이용한 직류전동기 구동 시스템의 임베디드 제어 시스템", 2007년도 대한전기학회 하계학술대회 논문집, 2007. 7.

〈 저 자 소 개 〉



지준근(池俊根)

1986년 서울대 전기공학과 졸업. 1988년 동 대학원 전기공학과 졸업(석사). 1994년 동 대학원 전기공학과 졸업(공학박사). 1994년~현재 순천향대 전기통신시스템공학과 교수. 당 학회 평의원.