

Matlab 환경하에서 유도 전동기 벡터 제어 구현

김 배선\*, 설 남오\*\*, 한 우용\*\*\*, 김 성중\*  
 전북대 전자·정보 공학부\*, 서남대 전기·전자 통신학부\*\*, 전주 공업대 전기과\*\*\*

The Implementation of Field Oriented Control of Induction motor in Matlab environment

Kim Bae-Sun, Sul Nam-Oh, Han Woo-Yong, Kim Sung-Jung  
 ChonBuk univ. SeoNam univ. ChonJu technical Collage. Chonbuk univ.

**Abstract** - An easier implementation method of the field oriented control of induction machine using Matlab/Simulink dSpace board is proposed in this paper. Space Vector PWM and Indirect Fied Oriented Control Algorith is designed for the help of Simulink. And this system is simultaneously simulated and experimented in Matlab/Simulink environment with dSpace borad (DS1102). It is possible that Matlab and dSpace board compiler can make \*.c and \*.obj file of models designed in Matlab/Simulink environment automatically. Experimental results are given

그림 1은 본 연구에서 유도 전동기 벡터 제어 실험을 위해 사용된 전체 시스템의 구성도를 나타내고 있다. 그림 1에서 DS1102 보드의 주 프로세서는 단일 사이클 명령 실행 시간이 33.33(ns)인 32비트 floating-point DSP [TMS320C31]이고 부 프로세서는 16비트 fixed-point DSP [TMS320P14]이다. 부 프로세서는 6개의 PWM 발생 회로와 4개의 타이머, 디지털 입·출력 포트, 4개의 캡처 입력과 시리얼 통신을 지원한다. 128 [KWord] 크기의 zero wait SRAM은 60(MHz) 클럭 속도로 동작하고 사용자가 작성한 프로그램을 다운로드 받아 실행시킨다.

1. 서 론

Matlab은 시스템 시뮬레이터로서 수학적인 모델링이 필요하나 Simulink 툴 박스를 사용하면 비선형·시변 시스템의 모델링이 매우 용이하다. 특히 코딩을 사용하지 않고서도 여러 종류의 제어 알고리즘을 쉽게 구현할 수 있다. 이러한 장점으로 인하여 현재 Matlab/Simulink는 전력 시스템을 분석하거나 유도 전동기 알고리즘을 연구하는데 적용되고 있다[1-4].

문헌 [1],[2]에서는 Matlab/Simulink 환경 하에서 유도 전동기 제어 시스템을 시뮬레이션한 결과와 마이크로 프로세서를 이용하여 실험한 결과가 유사함을 보였다. 그리고 문헌 [3]에서는 유도 전동기의 벡터 제어를 위한 자속과 회전자 저항 추정 알고리즘 등을 Matlab/Simulink 환경 하에서 구현할 수 있는 방법을 제시하였다. 또한 문헌 [4]에서는 SPWM으로 구동되는 인버터 시스템을 Matlab/Simulink로 구현할 수 있는 방법을 제시하였다 [4]. 그러나 이들 방법들은 대부분 Matlab/Simulink 환경 하에서 설계된 유도 전동기 제어 시스템의 시뮬레이션만을 수행하고 있고 실제적인 구현 방법은 제시되어 있지 않다.

따라서 본 논문에서는 Matlab/Simulink 환경 하에서 유도 전동기 벡터 제어 시스템을 손쉽게 구현할 수 있는 방법을 제시하였다. 유도 전동기 벡터 제어를 위한 전체 시스템의 모델은 Simulink 툴 박스를 이용하여 설계하고 이를 dspace 보드 (DS1102)를 통하여 제어 실험을 수행하였다. 이 방법은 Matlab과 dSpace 보드 컴파일러가 설계된 모델의 \*.c와 \*.obj 파일들을 자동으로 생성시켜 주기 때문에 어떤 코딩(예: c 언어로의 변환)없이도 시스템의 시뮬레이션과 실험을 동시에 수행할 수 있다. 그리고 DS1102 보드를 이용한 하드웨어 구성 방법 및 구동 시스템의 구현 방법에 대해서 서술하였고 실험 결과를 통해 제시한 방법의 유용성을 입증하였다.

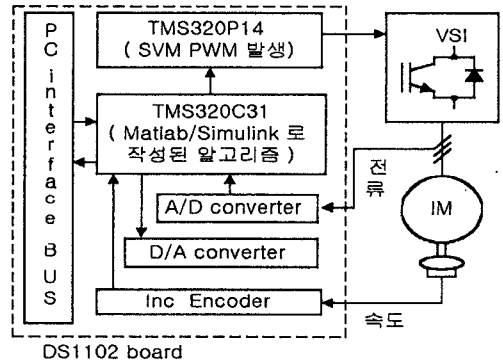


그림 1 DS1102 보드의 구성도

그 밖의 주요 주변회로에 12비트의 800[KHz] 샘플링 A/D 2채널, 16비트 250[KHz] 샘플링 A/D 2채널이 있으며 12비트 D/A가 4 채널, 최소 120[ns] 엔코더 펄스폭을 24비트로 카운팅하는 중분형 인코더 블록이 2채널 있다. 본 논문에서는 주 프로세서에서 제어 알고리즘을 연산하고 부 프로세서는 PWM 신호의 발생을 담당하도록 구성하였다. 또한 12비트 2채널의 A/D 컨버터를 이용하여 전류 신호를 받아 들일 수 있도록 하였다. 그리고 본 논문에서는 이용되지 않았으나 4채널 D/A 컨버터는 알고리즘 구동시 모니터링 채널로 활용할 수 있으며 2채널의 중분형 엔코더 블록 중 1채널로 속도 신호를 받았다. 이들 하드웨어 장치들은 dSpace사에서 제공하는 라이브러리에 의해서 Matlab/Simulink 환경 내에서 툴박스처럼 사용하여 외부 구동부와 인터페이스가 가능하다. PC와의 인터페이스는 내부 버스를 사용한다. 데이터 전송의 불일치를 피하기 위해서 두 번의 16비트 PC-BUS 실행을 한 번의 32 비트 DSP-bus 전송으로 매핑시키는 변환기를 내장하고 있다[7].

2.2 구현 및 실험 방법

그림 2는 Matlab/Simulink 환경 하에서 작성한 유도 전동기 제어 시스템을 구동하는 프로그램이다. 이 구동 프로그램은 벡터 제어부, 속도 제어부, 전류 제어부, 공간벡터 PWM 발생부로 구성되어 있다. 공간 벡터 PWM은

2. 시스템의 구현 및 실험 방법

2.1 DS1102 보드의 구성

S-function 함수를 사용하여 C 언어로 구현되었다. 이를 제외한 나머지 구동 프로그램은 Simulink 툴 박스를 이용하여 구현하였다. 구동 프로그램에서 외부 구동부 하드웨어와 인터페이스하는 데 사용되는 PWM 게이팅 신호를 출력하는 DS1102PWM, 전류 신호를 받는 DS1102ADC, 유도 전동기의 속도 신호를 받는 엔코더를 시뮬레이션 모델의 유도 전동기부와 유도 전동기 모델부에서 출력되는 전류 신호, 그리고 속도 신호를 대체하면 시뮬레이션을 바로 할 수 있다. 이 구동 프로그램의 디버그는 모니터링 소프트웨어인 Control Desk를 이용하면 쉽게 할 수 있다. 왜냐하면 Control Desk는 구동 프로그램을 DS1102 보드에서 실행시키고 원하는 부분의 신호가 그림 7에서처럼 실시간으로 확인되기 때문이다.

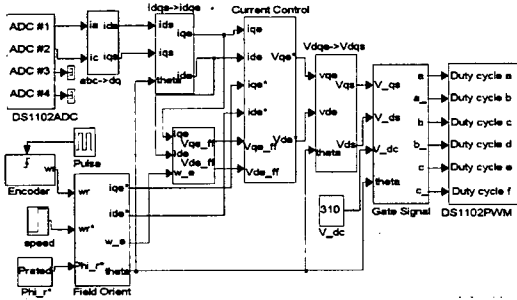


그림 2 DS1102보드에서 실행되는 구동 프로그램

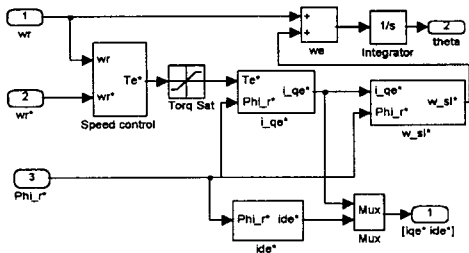


그림 3 간접 벡터 제어 알고리즘의 구성도

그림 3은 그림 2의 벡터 제어 블록을 보여 주고 있으며 자속을 일정하도록 하여 슬립각을 계산하여 전동기 속도와 합한 자속각에 의해 제어되는 간접 벡터를 구성하였다. 그리고 속도 제어기 및 전류 제어기는 PI 제어기로 구성하였으며 Anti-Windup을 첨가 하였다.

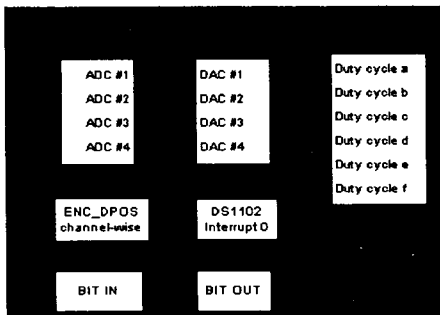


그림 4 DS1102 보드 라이브러리

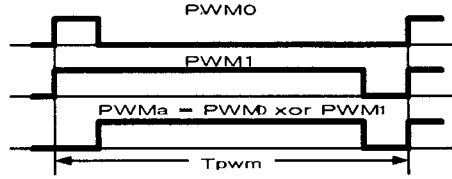


그림 5 xor 게이트로 만들어진 PWM 신호

그림 4는 dSpace사에서 제공하는 DS1102보드에 대한 RTI (Real Time Interface) 라이브러리의 일부분이고 이 라이브러리는 Simulink 툴 박스처럼 사용할 수 있다. dSpace사의 라이브러리 중에서 DS1102PWM을 이용하여 그림 5에서 PWM의 게이팅 신호가 간단한 게이트 로직에 의해 얻어진다. 또한 Matlab/Simulink 환경에서 지원되는 S-function을 이용하여 공간 벡터 PWM 발생에 필요한 기준 전압 명령이 위치한 섹터를 판별하고 유효 벡터 인가 시간을 계산하여 섹터에 따른 게이팅 신호를 발생하도록 프로그램 하였다. 그림 4의 DS1102PWM과 그림 5의 로직을 통해서 IPM 인버터의 구동 회로에 PWM 신호를 인가하였다.

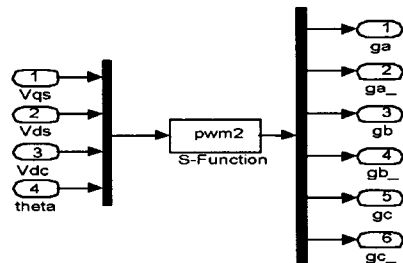


그림 6 S-function 으로 구현한 PWM

그림 6은 Matlab/Simulink 환경에서 S-function으로 구현된 공간 벡터 PWM 블록을 보여주고 있다.

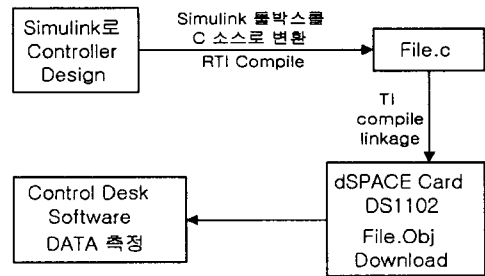


그림 7 프로그램 실행 흐름도

그림 7의 프로그램 실행 흐름도는 Matlab/Simulink 환경에서 작성된 프로그램이 컴파일되고 DS1102 보드에 다운로드 되어 실행되는 흐름도를 나타낸다. 그림 7에서 볼 수 있듯이 Matlab/Simulink 환경하에서 원하는 제어 알고리즘을 툴 박스를 사용하여 설계한 다음 Real Time Workshop에서 컴파일하면 배치 파일에 의해서 Simulink 블록을 \*.C' 소스로 다시 번역한다. 그런 후 TI사의 컴파일러에 의해서 \*.obj를 생성하고 이를 DS1102 보드에 다운로드하여 실행한다. 한 번 프로그램이 실행되면 소프트웨어 Control Desk 상에서만 정지 및 재실행이 가능하다. 또한, 프로그램이 다시 컴파일 될 때까지 현재 다운로드된 프로그램을 유지하고 있어서 구동부의 손상을 예방하도록 되어 있다.

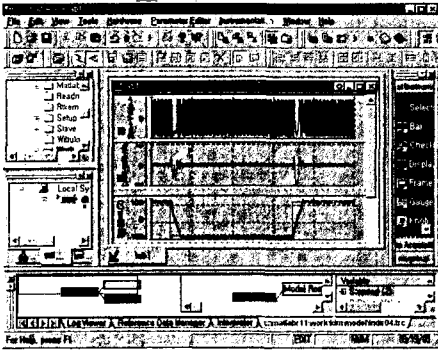


그림 8 Control Desk 소프트웨어

그림 8은 모니터링 소프트웨어인 Control Desk 이다. 그림 8의 중앙부에서 실험시 실시간으로 전류 및 속도, 전압 신호를 모니터링하고 있으므로 Matlab/Simulink 환경 하에서 작성된 알고리즘의 성능을 파악할 수 있다. 또한 모니터링하고 있는 신호를 실시간으로 저장할 수 있으며 알고리즘의 수행 및 정지 등 여러 기능을 제공하고 있다.

### 3. 실험 결과

실험은 소프트웨어로 Matlab/Simulink, 하드웨어로 DS1102 보드, IPM 인버터, 1024 펄스 엔코더, 5[HP] 유도 전동기를 사용하여 실시하였다. 그림 2와 같이 Matlab/Simulink 환경 하에서 구동 프로그램을 설계하여 ode1의 적분 알고리즘을 사용하고 알고리즘이 fixed step : 200[μsec]마다 수행되도록 실험을 실시하였다. 실험 결과는 Control Desk에서 실시간으로 데이터를 저장하였다.

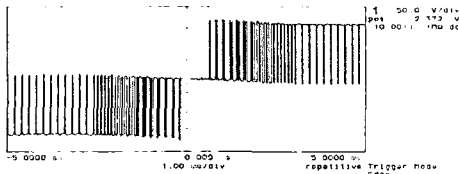


그림 9 선간 전압

그림 9은 150[V]의 DC 링크 전압을 인가하여 그림 2의 프로그램으로 전동기를 구동할 때 발생한 선간전압이다.

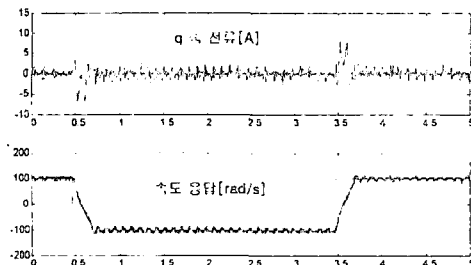


그림 10 무부하시 100[rad/s]에서 q 축 전류와 속도 응답

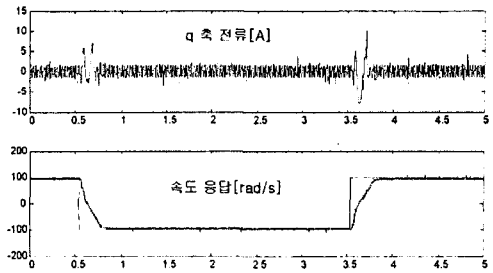


그림 11 일정부하시 100[rad/s]에서 q 축 전류와 속도 응답

그림 10과 11은 각각 100 [rad/s]에서 부하를 인가하지 않았을 때와 일정 부하를 인가하였을 때의 q축 전류와 속도 응답 곡선이다. 부하를 인가하지 않았을 때나 인가하였을 때도 기준 속도 변화에 잘 추종하고 있음을 알 수 있다. 따라서 이상과 같은 실험 결과를 통하여 Matlab/Simulink와 DS1102 보드를 이용하면 코딩없이 제어 시스템의 시뮬레이션과 실험을 동시에 수행할 수 있음을 확인하였다.

### 4. 결 론

본 논문에서는 Matlab/Simulink 환경 하에서 유도 전동기 제어 시스템을 모델링하고 이를 직접 DS1102 보드를 통하여 유도 전동기 제어 실험을 수행할 수 있는 방법을 제시하였다. 제시된 방법에 의하여 모델링된 시스템을 이를 실험한 결과 다음과 같은 장점이 있음을 확인하였다.

- 첫째, Simulink 툴 박스를 사용하기 때문에 코딩을 할 필요가 없어 제어 알고리즘의 설계가 간편하다.
  - 둘째, 알고리즘이 Matlab/Simulink 환경 하에서 컴파일되어 실행되므로 알고리즘 성능을 즉시 확인할 수 있다.
  - 셋째, 시뮬레이션과 실험이 동시에 실시될 수 있기 때문에 제어 알고리즘 개발에 필요한 시간이 많이 단축된다.
- 이런 관점에서 볼 때, 본 연구에서 제시한 방법을 이용하면 유도 전동기 제어 알고리즘을 연구·개발하는 데 Matlab/Simulink와 DS1102보드가 효과적으로 이용될 수 있음을 입증하였다.

### [참 고 문 헌]

- [1] R. Teodorescu, "A Simulink Approach to Power Control Electronics Simulations", EPE'95, pp 3954-3958, 1995
- [2] Liu Guohai, "The Simulation of a Motor-Inverter System with Matlab", IPENC'97, pp 136-140, 1997
- [3] Luis F. A. Pereira, "A Simulation Framework for Flux Estimation and Vector Control of Induction Machines", IEEE, pp 1587-1591, 1998
- [4] B. K. Lee, "A Simplified Functional Model for 3-Phase Voltage Source Inverter Using Switching Function Concept", IEEE, pp 462-467, 1999
- [5] B. K. Bose, "Power Electronics and AC Drivers", Prentice Hall, pp 264-276, 1986
- [6] "TMS320C3x : User's Guide", Texas Instrument, 1994
- [7] "Floating-Point Controller Board DS1102", dSpace GmbH, pp 1-48, 1995
- [8] "controlDesk:Experiment Guide", pp 50-215, dSpace, 1999