

Matlab/Simulink와 DS1104를 이용한 유도 전동기 벡터제어

임종빈*, 고성철*, 배재남*, 강동우*, 조인해*, 이주*
 한양대학교 전기공학과*

Vector Control of Induction Motor using Matlab/Simulink and DS 1104

Jong-Bin Im*, Sung-Chul Go*, Jae-Nam Bae*, Dong-Woo Kang*, In-Hae Cho*, Ju Lee*
 Hanyang University*

Abstract - This paper presents a impletation of vector control of induction motor using Matlab/Simulink and dSPACE DS1104. System consists of digital input block, digital ouput block, ADC block, protection block, motor control block and PWM block. It is applied indirect vector control and PI controller to speed controller and current controller.

1. 서 론

전동기 제어는 일반적으로 먼저 시뮬레이션을 통해 제어기를 설계한 후, 다시 C나 어셈블리 언어로 프로그램을 작성하여 구현하였다. 그러나 제어이론 및 제어기 설계 기술의 발전으로 제어 시스템이 더욱 복잡하고 다양화됨에 따라 기존의 과정으로 제어를 구현하기에 어려움이 많았다.[1]-[2]

이러한 시스템을 보다 쉽고 간편하게 구현하기 위해 Matlab, RT-Lab 과 같은 시스템 시뮬레이터를 사용하게 되었다. 시스템 시뮬레이터는 수학적인 모델로 표현되는 시스템을 모델링하기 적합하여 전기기계와 제어 시스템의 시뮬레이션에 적합하다. Matlab의 경우 수학적인 모델링이 필요하나 Simulink Tool Box를 사용하면 비선형·시변 시스템의 모델링을 쉽게 할 수 있다.[3]

이처럼 Matlab/Simulink를 이용하여 텍스트 방식이 아닌 그래픽 방식으로 제어기를 설계하고 시뮬레이션 한 다음, dSPACE 시스템을 이용하여 직접 실시간 프로그램으로 자동 변환하여 제어기의 응답을 테스트함으로써 전동기 구동 시스템의 제어기 설계를 짧은 시간 안에 매우 편리하게 구현할 수 있다.

본 논문에서는 dSPACE 시스템 중 DS1104를 사용하여 유도전동기의 벡터 제어를 실시하였다.

2. 유도 전동기의 모델링

고정자 및 회전자 전압 방정식은 식 (1)과 같다.

$$v_{abc_s} = R_s i_{abc_s} + \frac{d\lambda_{abc_s}}{dt}, \quad v_{abc_r} = R_r i_{abc_r} + \frac{d\lambda_{abc_r}}{dt} \quad (1)$$

위 식 (1)에 기준 좌표계 이론을 도입하면 abc 3상 좌표계를 임의의 속도 ω 로 회전하는 dq축 2상 좌표계로 표시된 유도 전동기의 고정자 전압 방정식을 구할 수 있다.

$$v_{ds} = R_s i_{ds} + \frac{d\lambda_{ds}}{dt} - \omega \lambda_{qs}, \quad v_{qs} = R_s i_{qs} + \frac{d\lambda_{qs}}{dt} + \omega \lambda_{ds} \quad (2)$$

회전자가 ω_r 의 속도로 회전하면 회전자에 고정된 dq축은 회전좌표계를 기준으로 ω_{sl} 의 속도로 회전한다. 벡터 제어는 간접 벡터 제어방식을 사용하였다. 그러므로 농형 유도 전동기의 회전자 전압 방정식은 다음과 같다.

$$v_{dr} = R_r i_{dr} + \frac{d\lambda_{dr}}{dt} - \omega_{sl} \lambda_{qr} = 0, \quad v_{qr} = R_r i_{qr} + \frac{d\lambda_{qr}}{dt} + \omega_{sl} \lambda_{dr} = 0 \quad (3)$$

그리고 고정자와 회전자 자속은 다음과 같이 표현된다.

$$\lambda_{ds} = L_s i_{ds} + L_m i_{dr}, \quad \lambda_{qs} = L_s i_{qs} + L_m i_{qr} \quad (4)$$

$$\lambda_{dr} = L_r i_{dr} + L_m i_{ds}, \quad \lambda_{qr} = L_r i_{qr} + L_m i_{qs} \quad (5)$$

회전자 자속이 d축에만 있다고 가정하면, $\lambda_r = \lambda_{dr}^e + j\lambda_{qr}^e = \lambda_{dr}^e$ 이다. 즉,

$$\lambda_{qr}^e = 0 \quad (6)$$

가 되고, 식 (6)를 식 (3)에 대입하면,

$$\omega_{sl} = \omega_e - \omega_r = -\frac{R_r i_{qr}^e}{\lambda_{dr}^e} = \frac{R_r}{\lambda_{dr}^e} \frac{L_m}{L_r} i_{qs}^e \quad (7)$$

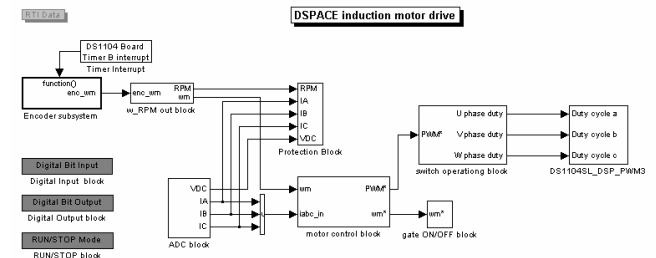
$$\theta = \int (\omega_r + \omega_{sl}) dt \quad (8)$$

이 된다.

그러므로 출력 토크는

$$T_e = \frac{3}{2} \frac{P}{L_r} \frac{L_m}{L_r} (\lambda_{dr}^e i_{qs}^e - \lambda_{qr}^e i_{ds}^e) = \frac{3}{2} \frac{P}{L_r} \lambda_{dr}^e i_{qs}^e \quad (9)$$

이 된다.



<그림 1> Real Time Interface(RTI)에서 동작하는 구동 프로그램

<표 1> 유도 전동기 상수

정격용량[kW]	0.37	$R_s [\Omega]$	0.782
정격전압[V]	380	$R_r [\Omega]$	0.583
정격전류[A]	1.3	$L_s, L_r [mH]$	671
정격속도[rpm]	1670	$L_m [mH]$	650
극수	4		

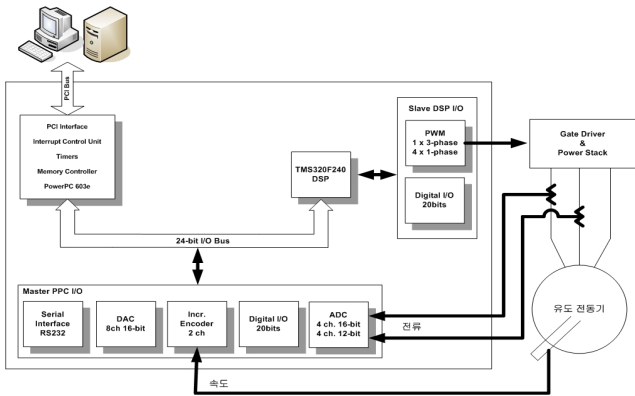
그림 1과 같이 유도 전동기 구동 프로그램은 디지털 입력부와 디지털 출력부, ADC부, 시스템 보호부, 전동기 제어부, PWM부로 구성되어 있다. 전동기 제어부는 급속한 속도 변화를 방지하기 위한 블록과 PI 제어기를 사용한 속도 제어부와 전류 제어부로 구성되어 있다. 표 1은 본 논문에 사용된 유도 전동기의 상수를 나타내고 있다.

3. 시스템 구현 및 실험결과

3.1 dSPACE장비를 이용한 시스템 구현

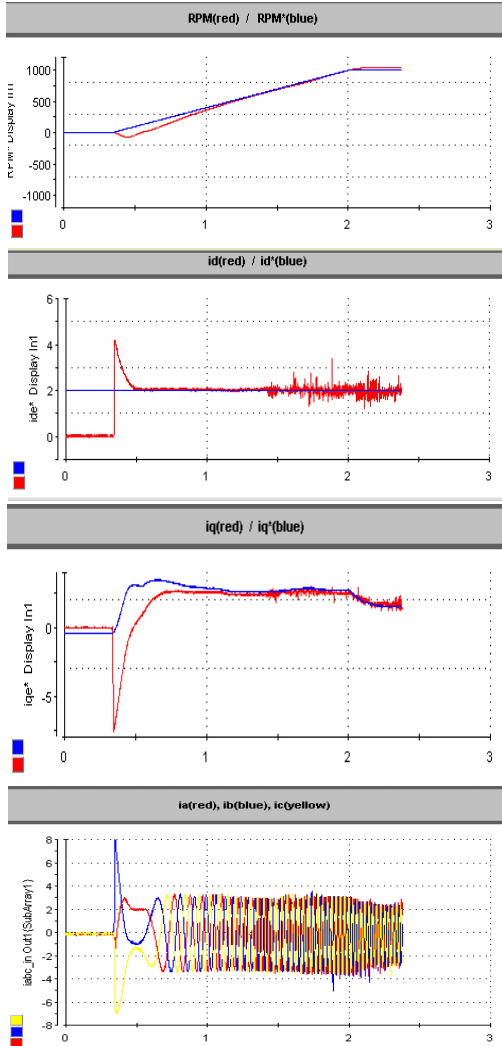
그림 2는 DS1104보드의 구성도이다. 실제 유도 전동기 구동 시스템 dsm 크게 하드웨어와 소프트웨어로 나뉘는데, 하드웨어는 크게 호스트 컴퓨터, DS1104 R&D Controller Board, IGBT 인버터 파워 스택, 인터페이스 보드등 구성되어 있고, 소프트웨어는 Matlab/Simulink의 실행장에서 실행이 가능한 구현 소프트웨어인 RTI(Real Time Interface)와 윈

도우장에서 별도로 실행할 수 있는 동작 소프트웨어인 ControlDesk로 구성되어 있다.



〈그림 2〉 dSPACE장비(DS1104)의 구성도

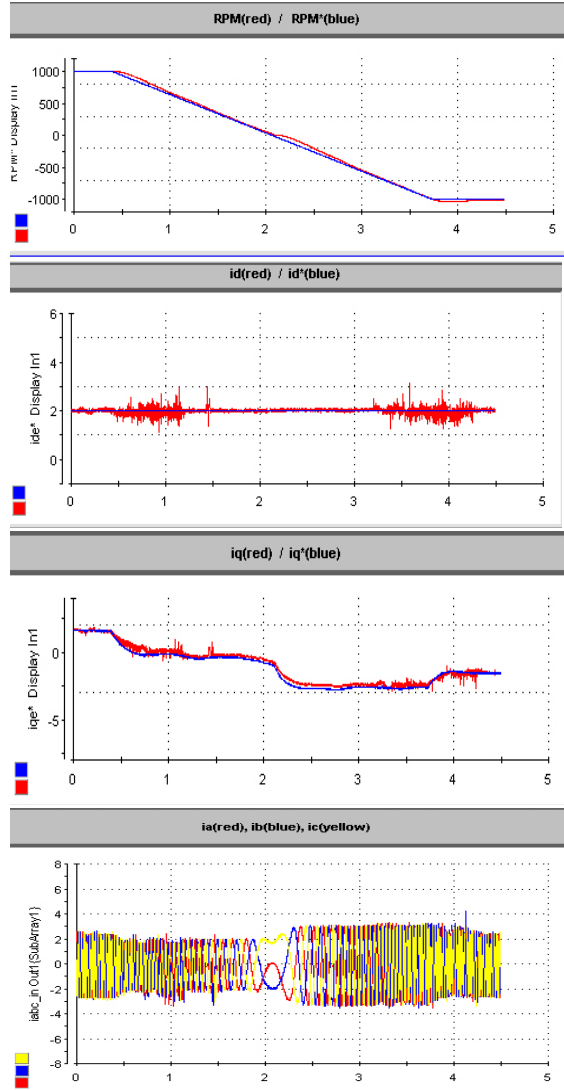
3.2 실험 결과



〈그림 3〉 지령속도 변화(0→1000[rpm])시 속도, d,q축 및 3상전류 특성

그림 3은 무부하 상태에서 지령속도를 0에서 1000[rpm]으로 변화시켰을 때 실제속도와 그에 따른 d,q축과 3상 전류 파형을 나타낸 그림이다. d축 전류지령치는 2[A]로 고정하였으며, 안정성을 고려하여 지령속도를 1초당 300[rpm] 변화도록 하였다. 초기에 실제속도가 지령속도를 쫓아가지 못하였으나 0.5초 이후에 바로 쫓아가는 것을 확인할 수 있었다. 지령속도와 실제속도와 오차가 커짐에 따라 이를 줄이기 위해 d축 전류가 상승하다 오차가 없어짐에 따라 지령치인 2[A]로 간다. q축 전류의 경우 d축 전류의 값이 상승함에 따라 q축 전류는 음의 값을 가지다 d축

전류가 지령치인 2[A]에 가까워짐에 따라 q축 지령값을 따라가게 된다. 지령속도를 1000에서 -1000[rpm] 뿐 나머지 속도에서는 오차가 거의 없는 것을 확인할 수 있다. 전류가 양속도에서 음속도로 바뀔 때 따라 3상 전류도 역전됨을 확인할 수 있었다.



〈그림 4〉 지령속도 변화(1000→-1000[rpm])시 속도, d,q축 및 3상전류 특성

4. 결 론

본 논문에서는 Matlab/Simulink를 이용하여 시뮬레이션을 하고 이를 DS1104보드를 통해서 간접벡터제어 방식을 통해 유도전동기 속도제어를 구현하였다. 기존에 DSP보드를 사용한 제어는 DSP보드 제작시간이 오래 걸리고 이를 구동하기 위한 코딩 등으로 인해 제어 알고리즘 개발에 시간이 오래 걸렸으나 dSPACE를 사용하면 시간을 많이 단축할 수 있어 제어 알고리즘 연구 및 실험을 효과적으로 할 수 있다. 또한 과전류 차단이나 과전압 차단, 과열 차단 등 시스템 보호 기능을 갖추고 있어 안전하게 작동할 수 있는 장점이 있다.

【참 고 문 헌】

[1] 지준근, “전력전자공학의 실험실습 교육을 위한 소프트웨어 및 하드웨어 소개”, 전력전자학회지, 제 12권 제 3호, pp. 25-31, 2007
 [2] 이학주, 권성철, 양승권, 오성업, 성세진, “Matlab/Simulink기반 유도전동기 벡터제어 시스템의 실시간 시뮬레이션”, 전력전자학술대회 논문집, pp. 19-22, 2004
 [3] 김배순, 한우용, 이찬구, “Matlab/Simulink & dSpace 보드를 이용한 유도 전동기 벡터 제어 시스템의 구현”, 전기학회 논문지, 49B권 10호, pp 675-682, 2000