MATLAB/SIMULINK의 모델기반 제어기 설계를 이용한 표면부착형 영구자석 동기전동기의 벡터제어 구현

論 文
57-8-12

Implementation of Vector Control for SPMSM Using Model Based Controller Design in MATLAB/SIMULINK

池俊根† · 李鎔碩* · 車貴守**
(Jun-Keun Ji · Yong-Seok Lee · Guee-Soo Cha)

Abstract - This paper presents an implementation of vector control for SPMSM using model based controller design in MATLAB/SIMULINK. The model based controller design enables fast development of control system for motor by designing controllers and performing simulation on the GUI (Graphic User Interface) platform, converting program code directly into real-time programs, and then performing tests for the responses from controllers. The controllers designed in this paper are PI speed controller and decoupling PI current controller. Also space vector modulation method using offset voltage is used in PWM scheme. And system stability is also secured by close magnitude overmodulation method, maintaining dynamics of load when overmodulation occurs. The validity of vector control implemented is verified through simulations and experiments.

Key Words : MATLAB/SIMULINK, Model Based Controller Design, SPMSM, Vector Control

1. 서 론

오늘날 현대 산업의 발달로 우주 항공 산업과 각종 산업 분야의 자동화 및 제어기기 분야에서 동특성이 우수한 서보 제어 시스템이 요구되고 있다. NC 선반, 공작기계, 로봇 등과 같이 정밀하고 빠른 응답특성이 요구되는 서보제어용 전동기로는 대부분 직류전동기가 사용되어 왔으나, 직류전동기는 구조상 정류자 및 브러시를 가지고 있어서 정기적인 유지 보수를 필요로 하는 단점이 있고, 최근에 와서 제어기술의 발전과 더불어 반도체 기술의 향상으로 유지, 보수가 용이한 교류전동기로의 전환이 적극적으로 이루어지고 있다.[1]

교류전동기중 회전자 표면에 영구자석을 비돌극형으로 부착한 표면 부착형 영구자석 동기전동기(Surface Mounted Permanent Magnet Synchronous Motor : SPMSM)는 전동기의 역기전력이 정현파 형태를 갖기 때문에 전동기의 전류를 정현파 형태로 인가해 주어야 하는 제어상의 어려움이 있지만, 일단 정현파 전류를 인가해 주면 맥동 토크가 없는 이상적인 토크를 발생한다는 장점이 있다. 더욱이 1980년대 이후 전력용 반도체의 발달로 인버터에 의한 교류전동기 방식이 실용화되어 무보수화, 고점착 성능, 에너지 절감, 경량화 등의 성과를 이루었다.[2]

전력용 반도체 소자의 발달과 전동기 제어를 위한 주변기기가 내장된 DSP의 발전은 교류전동기의 구동 및 제어에 있어 하드웨어를 간소화시켰고 C언어를 사용한 코딩에도 충분한 연산속도를 가지는 단계까지 발전하였다. 본 논문에서 사용된 TMS320F2812에는 SVPWM 및 PWM 발생기, QEP(Quadrature-Encoder Puls)회로와 12비트의 고속 AD 변환기 등을 내장하여 DSP 단일 칩 하나로 교류전동기의 제어 시스템 구현을 가능하게 만들었다.[3]

일반적인 전동기 제어 시스템은 먼저 컴퓨터 시뮬레이션을 통해 제어기를 설계한 후, C나 assembly 언어로 실시간(real-time) 제어 프로그램을 작성하여 구현한다. 그러나 최근 들어 MATLAB/SIMULINK에서 TI C2000 DSP 플랫폼에서의 임베디드 시스템을 지원하는 기능이 추가되었다. 그리하여 MATLAB/SIMULINK를 이용한 모델기반 설계는 전반적인 제어 시스템 설계를 개념 단계에서 초기에 규정하고 자세한 부분의 설계단계에서는 GUI(Graphic User Interface)기반으로 필요한 블록을 그때 그때 추가하는 계층적 설계 프로세스를 적용함으로써 신속한 설계의 반복과 매개변수 최적화를 가능하게 하여 개발 시간을 현저히 감소시킬 수 있다.[4,7,8]

영구자석형 동기전동기는 회전계자형 동기전동기와 같은 구조로 되어 있어 벡터제어(vector control) 이론을 적용하여 우수한 동특성을 갖는 속도제어 시스템을 구현할 수 있다. 벡터제어는 3상 교류입력을 2상 입력으로 등가 변환하여 고정자 전류를 토크를 발생하는 q축 전류와 자속을 발생하는 d축 전류로 분리하여 각각 독립적으로 제어함으로써 타여자 직류전동기와 유사한 특성을 얻도록 하는 것이다. 따라서 본 연구에서는 자속과 토크를 완전히 독립적으로 제어하기 위해 전류제어기 자체에서 상호간섭성분과 역기전력에 대한 보상을 한 비간섭 전류제어기와 최소 거리 과변조 기법을 적용하여 0.63kW 표면부착형 영구자석 동기전동기의 벡터

† 교신저자, 正會員 : 順天鄕大 工大 情報技術工學部 敎授 · 工博
E-mail : jkji@sch.ac.kr
* 學生會員 : 順天鄕大 工大 電氣通信工學科 碩士
** 正 會 員 : 順天鄕大 工大 情報技術工學部 敎授 · 工博
接受日字 : 2008年 4月 17日
最終完了 : 2008年 6月 30日

제어 시스템을 구현하였다.

본 논문에서의 표면부착형 영구자석 동기전동기 벡터제어 시스템은 TI사의 eZdsp 2812보드와 MATLAB/ SIMUINK의 Embedded Target for TI C2000 DSP 라이브러리를 이용하여 구현하였고, 시뮬레이션과 실험을 통하여 속도 및 전류 제어기 설계의 타당성을 검증하였다. 이 시스템은 GUI 기반으로 제어기를 설계하고 Simulation을 한 다음 직접 실시간 프로그램으로 자동변환하여 제어기의 응답을 확인함으로써 전동기의 제어 시스템 개발을 짧은 시간 안에 매우 편리하게 구현할 수 있음을 소개하고자 한다.

2. MATLAB/SIMULINK의 모델기반 제어기 설계

2.1 MATLAB/SIMULINK의 Embedded Target for TI C2000 DSP 플랫폼

Embedded Target for TI C2000 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 C언어 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) 할 수 있다.[4]

2.2 임베디드 제어 시스템의 모델기반 설계

전동기 제어를 위한 모델기반 설계의 활용은 Clarke 및 Park 변환, PWM, PI 루프 등을 비롯한 벡터 제어 시스템의 모든 요소들을 다룰 수 있는 미리 구성된 블록들을 포함하는 블록셋(Blocksets)의 도입을 통해 활성화되었다. 이러한 새로운 툴을 통해 설계 엔지니어들은 직접 코딩한 C코드를 필요할 때만 통합하면서 원시 요소와 고급 알고리즘을 대표하는 사전 제작된 블록들을 이용한 도식 모델을 신속하게 구축할 수 있게 되며, 다양한 블록과 주변 기기간의 모든 통합 과정은 자동적으로 이루어진다. 사용자들은 중요한 엔지니어링 프로젝트에서 이 툴을 사용하게 되었고 더불어 산업체에서의 인지도도 높아졌다. 그림 1은 임베디드 전동기 제어 시스템의 실행을 위한 모델기반 설계 흐름을 보여준다. 대표적인 소프트웨어 툴은 각 단계에 표시되어 있다.

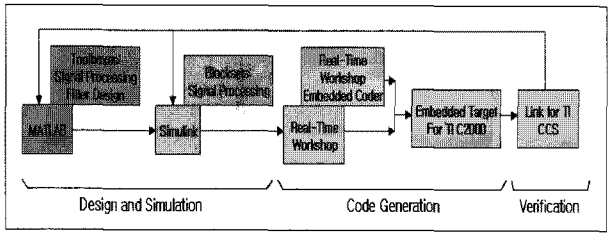

그림 1 모델기반 설계 흐름 및 소프트웨어 툴

Fig. 1 Model based design flow and SW tool

엔지니어들은 모델기반 설계를 도입함으로써 소프트웨어 상에서 제어기를 시뮬레이션할 수 있고, 이 과정에서 신속하게 모델을 변경하여 그에 따른 성능의 성과변화를 관찰할 수 있게 되어 빠른 반복을 통한 설계 성능 향상이 가능해진다. 이러한 접근법의 주요 이점은 하드웨어 프로토타이핑(Prototyping)에 많이 투자하지 않고 새로운 설계를 평가할 수 있는 유연성이다. 예를 들어, 엔지니어들은 모의 실시간 플랫폼에 제어기 모델을 구동할 수 있고, 이 때 전동기의 성능을 시뮬레이션할 수 있는 모델을 제어하거나 코드를 생성하여 실제 하드웨어에서 구동하고, 이를 실제 전동기나 전동기의 실시간 시뮬레이션 모델을 제어하는데 사용할 수 있다.[4]

2.3 자동 코드 생성을 통한 자동 실행 경로

MATLAB의 리얼타임 워크샵(Real Time Workshop)과 Embedded Target for TI C2000 DSP는 TI C2000 DSP용 SIMULINK 프로그램으로부터 고정 및 부동 소스 코드를 생성한다. 생성된 소스 코드는 자동적으로 그 다음 단계인 코드 컴포저 스튜디오 프로젝트로 옮겨진다. 생성 코드는 장치 드라이버와 제어 알고리즘 외에 실시간 스케줄러, 링커 커맨드 파일, 프로젝트 세팅 등을 포함하고 있다.[4]

3. 영구자석형 동기전동기의 모델링 및 벡터제어

3.1 3상 정지좌표계와 2상 회전좌표계의 상호 변환

3상 a, b, c 좌표계 전류를 2상 d, q 회전좌표계로 변환하는 행렬식은 a, b, c 좌표계를 α, β 정지좌표계로 변환하고, 이를 다시 d, q 회전좌표계로 변환하는 과정을 거치게 되므로 다음과 같이 정리된다.

$$\begin{pmatrix} i_{ds}^r \\ i_{qs}^r \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix} \frac{2}{3} \begin{pmatrix} cos0 & \cos\frac{2\pi}{3} & \cos\frac{4\pi}{3} \\ sin0 & \sin\frac{2\pi}{3} & \sin\frac{4\pi}{3} \end{pmatrix} \begin{pmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{pmatrix}$$

$$= \frac{2}{3} \begin{pmatrix} cos\theta & cos(\theta - \frac{2\pi}{3}) & cos(\theta - \frac{4\pi}{3}) \\ -sin\theta & -sin(\theta - \frac{2\pi}{3}) & -sin(\theta - \frac{4\pi}{3}) \end{pmatrix} \begin{pmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{pmatrix} \quad (1)$$

2상 d, q 회전좌표계를 a, b, c 좌표계로 변환하는 경우에도 역시 d, q 회전좌표계를 α, β 정지좌표계로 변환하고, 이를 다시 a, b, c 좌표계로 변환하는 과정을 거치게 되므로 위의 식을 역변환하면 다음 식이 성립한다.

$$\begin{pmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta \\ cos(\theta - \frac{2\pi}{3}) & -sin(\theta - \frac{2\pi}{3}) \\ cos(\theta - \frac{4\pi}{3}) & -sin(\theta - \frac{4\pi}{3}) \end{pmatrix} \begin{pmatrix} i_{ds}^r \\ i_{qs}^r \end{pmatrix} \quad (2)$$

3.2 영구자석 동기전동기의 모델링

회전자 좌표계로 좌표변환을 하여 영구자석형 동기전동기의 전압방정식을 구하면 다음과 같다.

$$V_{ds}^r = R_s i_{ds}^r + L_d \frac{di_{ds}^r}{dt} - w_r L_q i_{qs}^r \quad (3)$$

$$V_{qs}^r = R_s i_{qs}^r + L_q \frac{di_{qs}^r}{dt} + w_r L_d i_{ds}^r + w_r \lambda_f \quad (4)$$

발생 토크는 전류의 성분 ids와 iqs에 의해 다음과 같다.

$$T_e = \frac{3}{2}\frac{P}{2}(\lambda_f i_{qs}^r + (L_d - L_q) i_{ds}^r i_{qs}^r) \quad (5)$$

표면부착형 영구자석 동기전동기의 경우에는 회전자에 돌극성이 없으므로 전동기의 d축 리액턴스와 q축 리액턴스가 같게 되어 발생토크는 다음과 같다.

$$T_e = \frac{3}{2}\frac{P}{2}\lambda_f i_{qs}^r \quad (6)$$

회전자는 영구자석으로 되어 있어서 회전자의 쇄교자속이 일정하므로, 결국 고정자 전류의 토크발생 전류성분인 iqs를 제어함으로써 전동기의 토크를 직접 제어할 수 있게 된다.

전동기가 부하를 구동할 때의 역학적인 운동방정식은 다음과 같다.

$$T_e = J\frac{dw_m}{dt} + Bw_m + T_L\ ,\ w_m = \frac{P}{2}w_r \quad (7)$$

식 (6)과 (7)로부터 속도 제어 시스템 구성을 위한 PMSM의 모델은 아래의 식으로 나타낼 수 있다.

$$\frac{d}{dt}w_m = -\frac{B}{J}w_m + \frac{K_t}{J}i_{qs}^r - \frac{1}{J}T_L\ ,\ K_t = \frac{3}{2}\frac{P}{2}\lambda_f \quad (8)$$

3.3 영구자석 동기전동기의 벡터제어

벡터제어 기법은 교류전동기의 고정자 전류를 자속발생 전류성분인 ids와 토크발생 전류 성분인 iqs로 분리시켜 각각 독립적으로 제어함으로써, 타여자 직류전동기와 유사한 특성을 갖도록 제어하는 것이다. 이를 위해 전동기의 고정자 전류를 자속각을 기준으로 하여 제어하는데, 이를 벡터제어 혹은 자속기준제어라 한다. 일반적으로 동기전동기를 정격속도 이내에서 운전할 경우에는 q축 전류를 가변하여 토크를 제어하며, 공극자속은 회전자의 영구자석에 의한 것만을 이용하고, d축 전류는 0으로 유지하는 "ids = 0 제어"를 사용하는 것이 바람직하다.[5,6]

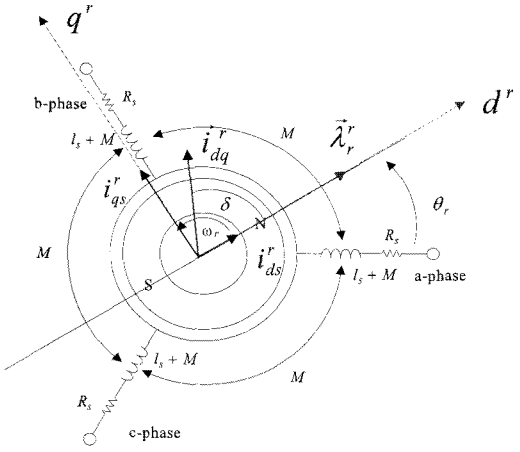

그림 2 표면부착형 영구자석동기전동기 벡터도

Fig. 2 Vector diagram of surface mounted PMSM

4. MATLAB/SIMULINK를 이용한 시뮬레이션

그림 3은 SPMSM의 벡터 제어 S/W 구성도이다. 제어 프로그램은 속도 지령과 전동기의 실제 속도의 오차를 이용하여 기준 전류인 $i_{qs}^r{}^*$을 생성하는 PI 속도 제어기를 가지고 있으며, $i_{ds}^r{}^*$는 '0'으로 설정하였다. 그리하여 생성된 d-q축의 전류지령 ($i_{ds}^r{}^*$, $i_{qs}^r{}^*$)과 d-q축의 실제전류(i_{ds}^r, i_{qs}^r)의 오차를 이용하여 PI 전류 제어기를 통해 기준 전압값을 생성하였고, 전류 제어기 출력에 d축은 q축의 전류에 의해 영향을 받고 또한 q축은 d축의 전류에 영향을 받는 간섭항을 $-w_r L_s i_{qs}^r$과 $w_r L_s i_{ds}^r$ 값으로 보상하였다. 그리고 역기전력을 $\lambda_f w_r$의 값으로 q축 전류 제어기의 출력에 전향보상(Feedforward Compensation)하였다.

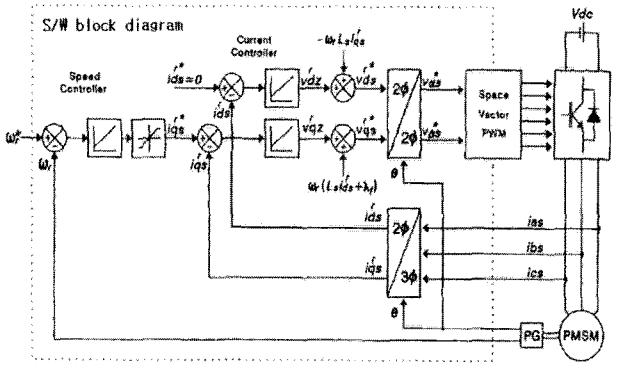

그림 3 SPMSM의 벡터 제어 S/W 블록도

Fig. 3 S/W block diagram for vector control of SPMSM

그림 4는 시뮬레이션에 사용하기 위한 시스템 블록도로서, 제 3절에서 구한 영구자석 동기전동기의 모델링과 벡터제어 및 속도 제어기, 전류 제어기를 이용하여 MATLAB/SIMULINK 환경에서 시뮬레이션을 실시하여 설계된 제어기의 성능을 확인하였다. 영구자석 동기전동기의 모델링은 전압 방정식, 토크 방정식, 기계적 운동 방정식으로 구현하였고, 속도 제어기는 비례-적분 제어기를 사용하였으며 속도 제어 주기는 1ms, 속도 제어 주파수 대역은 200[rad/s]로 설정하였다. 전류 제어기는 비간섭 비례-적분 제어기를 사용하였고, 전류 제어 주기는 50μs, 전류 제어 주파수 대역은 2000[rad/s]로 설정하였다. 또한 그림 5, 6, 7에 각각 속도 제어 블록, 전류 제어 블록, 영구자석 동기전동기의 모델링 블록들에 대한 내부 구조를 나타내었다. 그리고 시뮬레이션에 사용된 영구자석 동기전동기 파라미터는 표 1에 나타내었다.

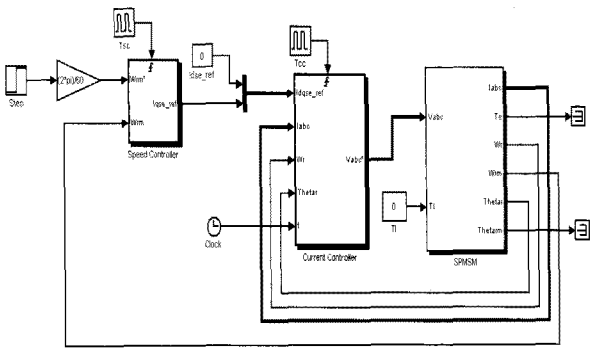

그림 4 시뮬레이션 블록도

Fig. 4 Simulation block diagram

표 1 영구자석 동기전동기의 상수
Table 1 Parameters of PMSM

전동기 정수	단위	값
정격 출력	kW	0.63
정격 토크	N · m	2.0
정격 전압	V	400
정격 전류	A	1.25
정격 회전속도	RPM	3000
관성모멘트	kg · m2	1.4E-4
전기자 저항	Ω(ph-ph)	24.5
전기자 인덕턴스	mH(ph-ph)	57.9

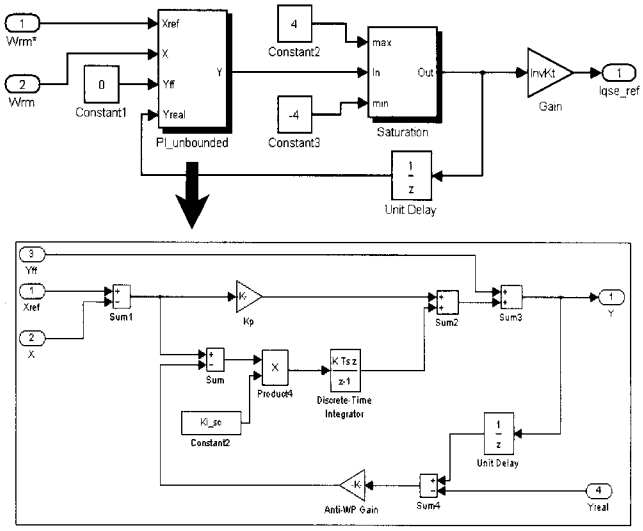

그림 5 속도 제어기 블록도
Fig. 5 Block diagram of speed controller

그림 5는 속도 제어기 블록을 보여준다. 속도제어기 블록은 비례-적분 제어기 블록과 Saturation 블록으로 구성되어져 있다. 그리고 제어기의 내부 블록을 함께 나타내었다.

그림 6은 전류 제어기의 블록을 나타낸다. 전류제어기 블록은 비례-적분 제어기와 전향보상(Feedforward) 제어기로 구성되어져 있다. 비례-적분 제어기의 구성은 앞에서의 속도 제어기와 같은 구조로 되어 있고 전향보상 제어기의 구성은 화살표 아래에 나타내었다.

그림 4에서 영구자석 동기전동기의 모델링 블록을 열면 그림 7과 같다. 그림 7은 좌표 변환 블록들과 제 3절에서 구한 동기 좌표계에서의 영구자석 동기전동기의 전압 방정식, 기계적 운동 방정식, 그리고 토크 방정식을 블록도로 표현하였다.

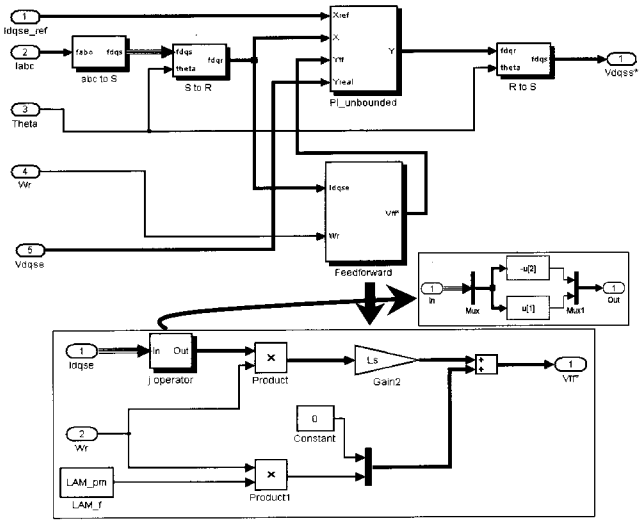

그림 6 전류 제어기 블록도
Fig. 6 Block diagram of current controller

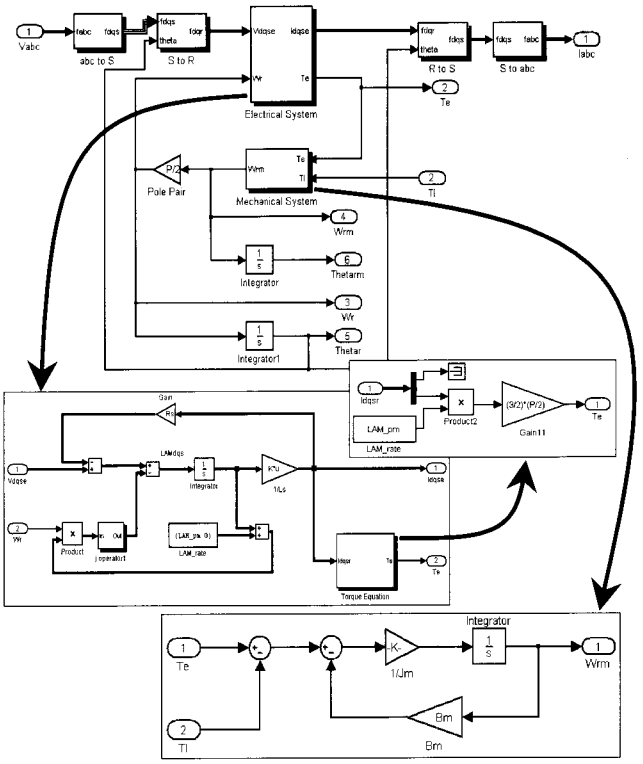

그림 7 영구자석 동기전동기의 모델링 블록도
Fig. 7 Block diagram of PMSM modelling

그림 8, 9, 10은 무부하시 속도지령을 0[RPM]에서 1000[RPM]으로 가속하고 다시 0[RPM]으로 감속 제어하였을 경우 각각 속도 제어 응답, q축 전류 제어 응답, d축 전류 제어 응답을 보여준다. 또한 그림 11, 12, 13은 관성부하시 속도지령을 -1000[RPM]에서 1000[RPM]으로 정역 회전시켰을 경우 각각 속도 제어 응답, q축 전류 제어 응답, d축 전류 제어 응답을 보여준다.

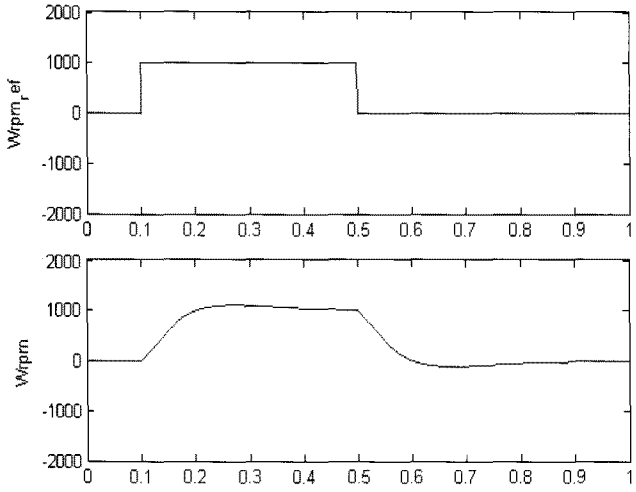

그림 8 속도 제어 응답(0 ↔ 1000[RPM])
Fig. 8 Speed control response(0 ↔ 1000[RPM])

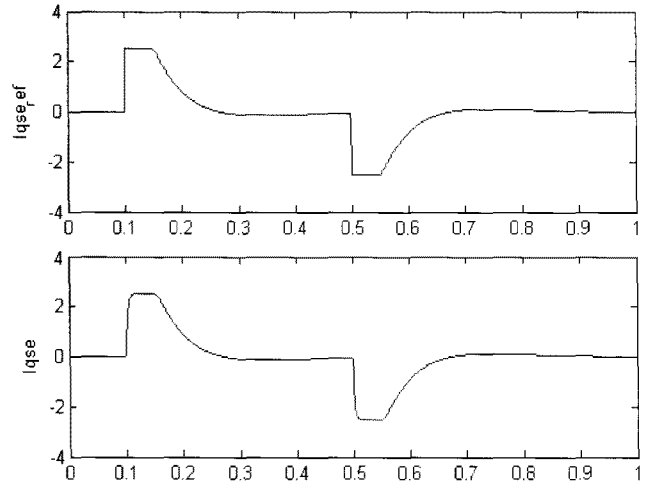

그림 9 q축 전류 제어 응답(0 ↔ 1000[RPM])
Fig. 9 q-axis current control response(0 ↔ 1000[RPM])

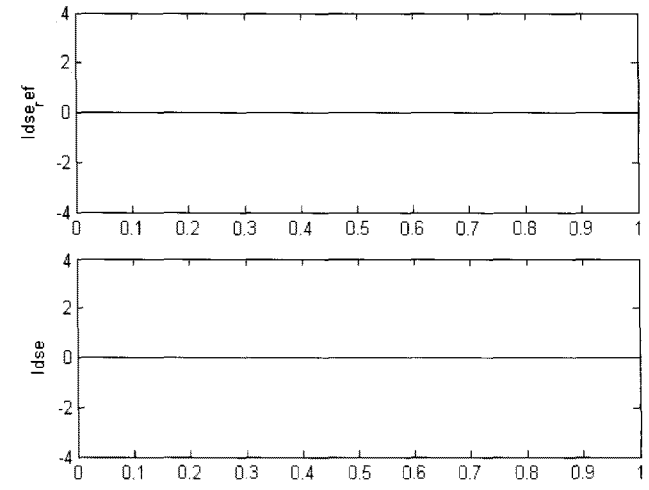

그림 10 d축 전류 제어 응답(0 ↔ 1000[RPM])
Fig. 10 d-axis current control response(0 ↔ 1000[RPM])

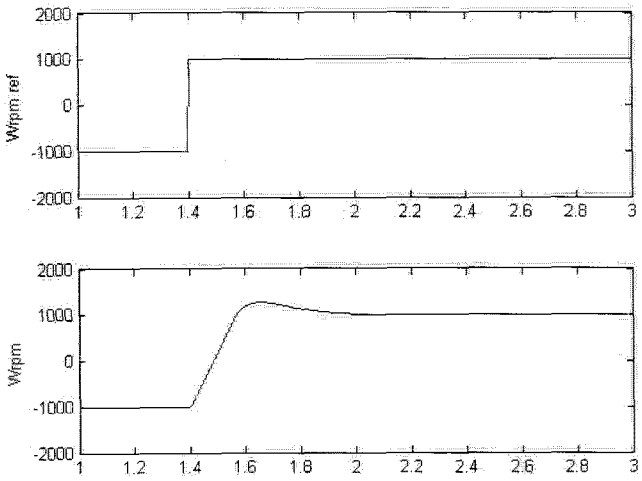

그림 11 속도 제어 응답(-1000 → 1000[RPM])
Fig. 11 Speed control response(-1000 → 1000[RPM])

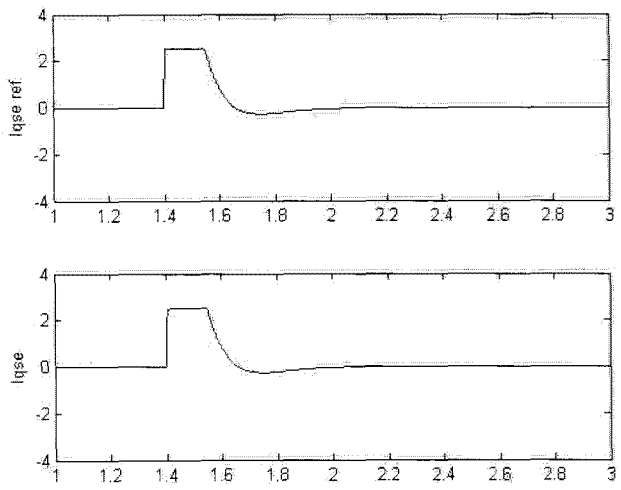

그림 12 q축 전류 제어 응답(-1000 → 1000[RPM])
Fig. 12 q-axis current control response(-1000→1000[RPM])

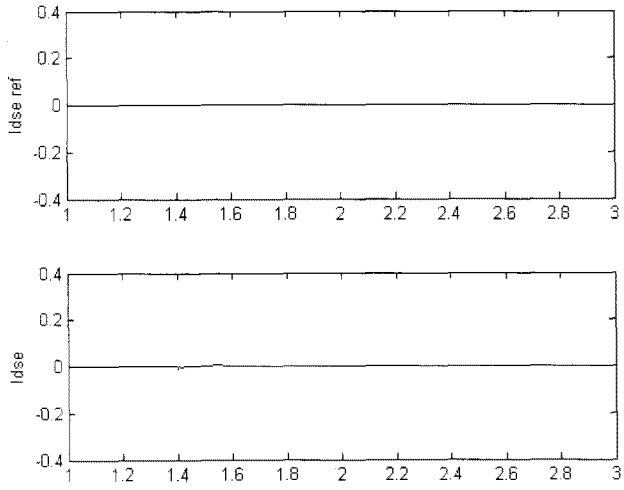

그림 13 d축 전류 제어 응답(-1000 → 1000[RPM])
Fig. 13 d-axis current control response(-1000→1000[RPM])

5. SPMSM의 벡터제어 시스템 구현

5.1 전체 시스템

그림 14에서 보듯이 전체 시스템 구성은 제어부인 eZdsp 2812 DSP 보드와 3상 인버터, 센서부 및 영구자석 동기전동기로 이루어져 있다. DSP 보드에 전류 및 속도제어기가 설계되어 있으며, 전류센서를 통해 ADC로 전동기의 전류값을 받고, 전동기의 엔코더를 통해 QEP로 전동기의 속도값을 받아 제어기가 동작하여 최종 출력은 PWM 신호로 내보낸다.

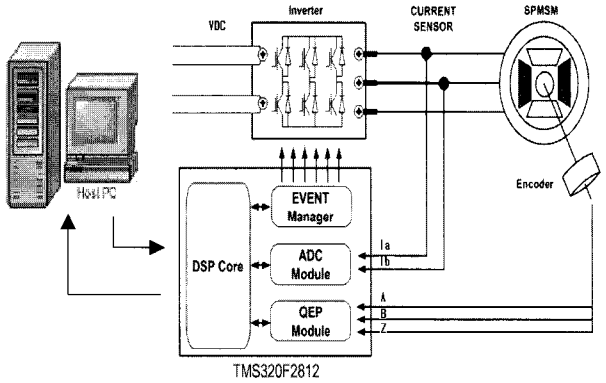

그림 14 전체 시스템 구성도
Fig. 14 Entire system configuration diagram

5.2 임베디드 타겟을 이용한 실시간 제어 프로그램

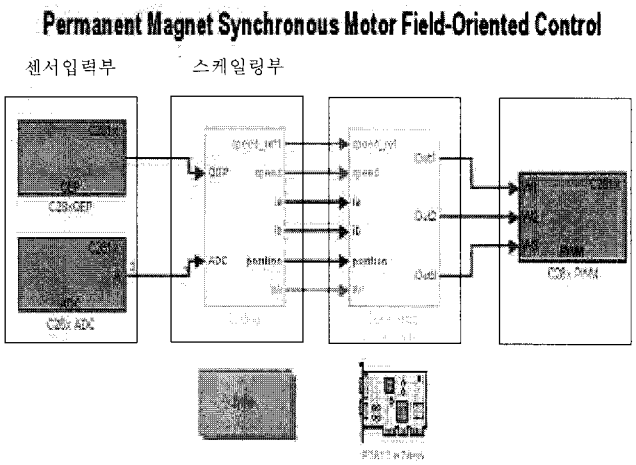

그림 15 SPMSM의 실시간 벡터제어 프로그램
Fig. 15 Real-time vector control program of SPMSM

그림 15와 같이 영구자석 동기전동기의 실시간 벡터제어 프로그램은 SIMULINK 프로그램으로 작성되었으며, 크게 센서입력부, 스케일링부, 제어부, 출력부로 나눌 수 있다. 센서값을 센서입력부로 받고 입력받은 값을 스케일링부에서 전동기의 속도 및 회전자 위치 측정과 고정자전류로 스케일링을 한다. 제어부에서는 속도제어, 고정자전류의 좌표변환 및 전류제어, 공간전압벡터 PWM 신호를 생성하여 출력부에서 최종 출력인 PWM 신호를 내보낸다.

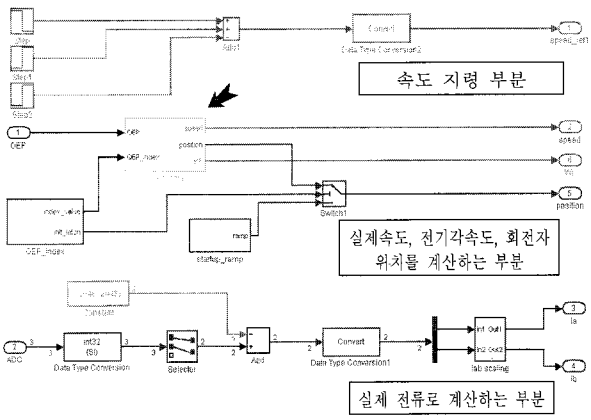

그림 16 스케일링부의 블록도
Fig. 16 Block diagram of Scaling

그림 16의 스케일링부의 블록은 크게 3가지로 나누어진다. 위에서부터 속도지령부분, Encoder 신호를 QEP로 입력받아 실제속도, 전기각속도, 회전자 위치를 계산하는 부분과 전류센서를 통해 입력받은 ADC 값을 실제 전류로 계산하는 부분이다.

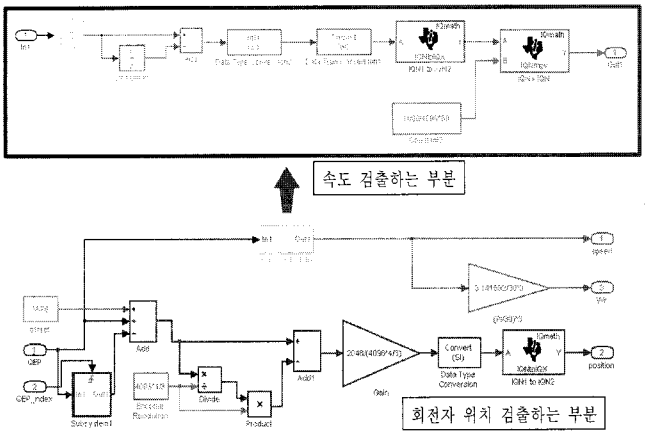

그림 17 실제 속도, 전기 각속도 및 회전자 위치 검출 블록도
Fig. 17 Block diagram of detecting real speed, Wr, and rotor position

그림 16에서 화살표로 표시한 블록을 열어보면 그림 17과 같다. 이 블록에서는 실제 속도, 전기 각속도 및 회전자 위치를 검출한다. 실제 속도를 측정하는 방식은 M, T, M/T방식들이 있는데 여기서는 M방식을 이용하여 속도측정을 하였다.

그림 15에서 제어부(Generating Space Vectors) 블록을 열면 그림 18처럼 나타난다. Generating Space Vectors 블록은 speed control 블록, Generating Raw Space Vectors 블록, PWM scaling 블록으로 나누어진다. Speed control 블록과 Generating Raw Space Vectors 블록은 아래에 설명되어져 있으며, PWM scaling 블록은 Generating Raw Space Vectors 블록의 출력값이 그림 18에서 출력부인 C28xPWM 블록의 입력에 맞도록 스케일링하는 블록이다.

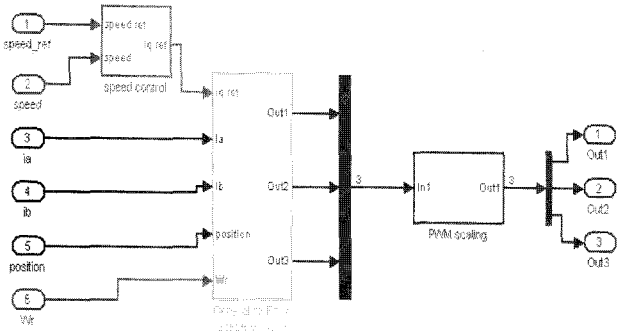

그림 18 제어부(Generating Space Vectors) 블록도
Fig. 18 Block diagram of Generating Space Vectors

그림 18에서 speed control 블록을 열면 그림 19처럼 나타난다. 그림 19는 설계한 속도 제어 블록이다. 속도 제어 블록은 비례-적분 제어기와 Anti-Windup 제어기로 구성되어 있다. 속도제어기의 제어주기와 제어주파수 대역은 시뮬레이션과 같게 설정하였고 Anti-Windup 제어기는 Limiter에 의해 제한된 출력의 편차를 궤환하여 적분기의 앞단에서 포화요소를 제거하였다.

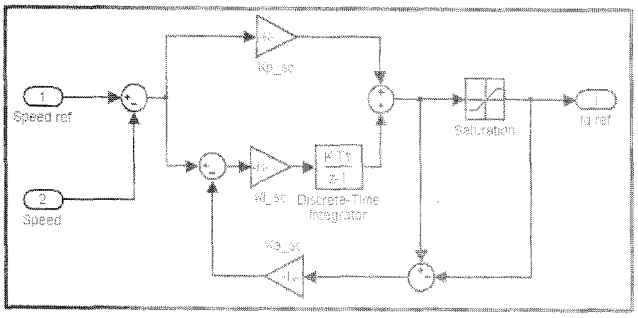

그림 19 속도 제어 블록도
Fig. 19 Block diagram of speed control

그림 18에서 Generating Raw Space Vectors 블록을 열면 그림 20처럼 나타난다. 이 블록은 좌표변환을 위한 Clarke Transfomation 블록, Park Transfomation 블록, Inverse Park Transfomation 블록, 공간 벡터 전압 발생을 위한 Space Vector Generator 블록과 전류 제어를 위한 Current control 블록으로 구성되어 있다.

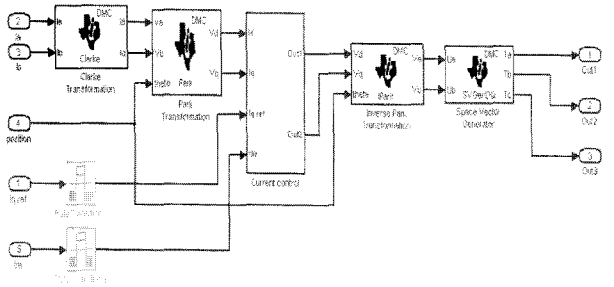

그림 20 Generating Raw Space Vectors 블록도
Fig. 20 Block diagram of Generating Raw Space Vectors

그림 20에서 Current control 블록을 열어 보면, 설계한 전류제어 블록인 그림 21가 나타난다. PI 제어기는 앞에서의 PI 속도 제어기와 같은 구조로 되어 있으며, 전류 제어기의 제어주기와 제어주파수 대역 또한 시뮬레이션과 같게 설정하였다.

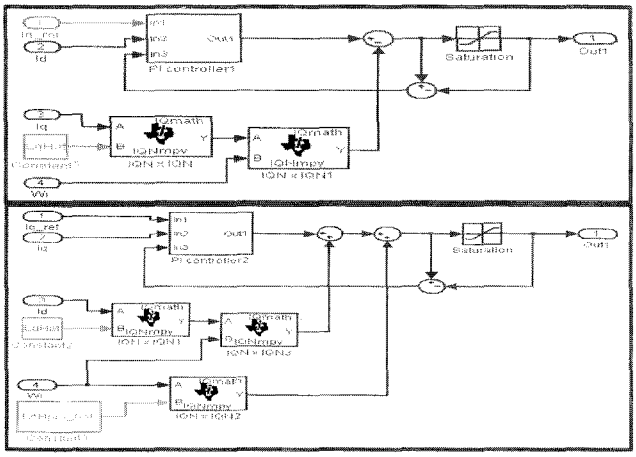

그림 21 비간섭 전류 제어 블록도
Fig. 21 Block diagram of decoupling current control

그림 21에서 보듯이, 전류제어기는 d축 전류와 q축 전류로 분리시켜 각각 독립적으로 제어하였고 전류 제어 블록 안에 있는 PI controller 블록은 앞에서 설명한 속도제어기 블록과 같은 구조로 되어있다. d축은 q축의 전류에 의해 영향을 받고 또한 q축은 d축의 전류에 영향을 받는 간섭 때문에, 이를 위해 $-w_r\hat{L}_s i_{qs}^r$과 $w_r\hat{L}_s i_{ds}^r$값으로 보상하였고 역기전력(E)을 $\lambda_m w_r$의 값으로 q축 전류 제어기의 출력에 전향보상(Feed-forward Compensation)하여 구성하였다.

또한 지령 전압이 인버터가 출력할 수 있는 범위를 벗어나게 되면 인버터가 이를 구현할 수 없게 되는데, 이 경우를 인버터의 과변조현상이라고 한다. 인버터가 이러한 과변조 구간에 진입하게 되면 출력 전압/지령 전압의 비율에 있어 선형성이 깨어지게 되므로 전압 제어의 성능이 심하게 왜곡되어 전체 시스템의 성능을 저하시키게 된다. 그래서 그림 22와 같이, 설계한 프로그램에서는 옵셋 전압을 이용한 공간 벡터 전압 변조 방식을 사용하여 출력전압을 결정하였다. 옵셋 전압은 위의 블록에서 보는 것처럼 $-(V_{\max}^* + V_{\min}^*)/2$의 값으로 결정하였고, 극전압을 Vdc/2, -Vdc/2사이의 값으로 제한하여 최소 거리 과변조 기법을 적용하였다.

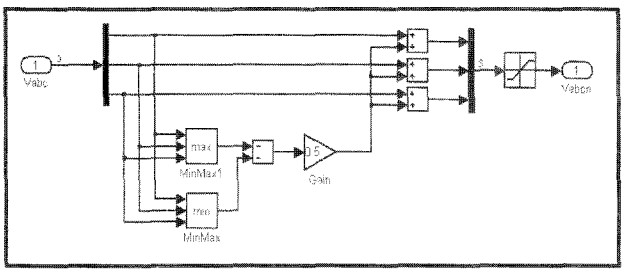

그림 22 옵셋 전압을 이용한 공간 벡터 전압 변조 블록도
Fig. 22 Block diagram of SVPWM using offset voltage

6. 실험 결과

본 연구에서는 영구자석 동기전동기 구동시스템의 성능을 확인하기 위해 그림 23과 같이 실험 장치를 구성하였다. 실험에 사용된 전동기의 상수는 시뮬레이션과 동일하며, 또한 속도 제어기 및 전류 제어기의 제어 주기와 제어 주파수 대역도 시뮬레이션과 같도록 설정하여 구동하였다.

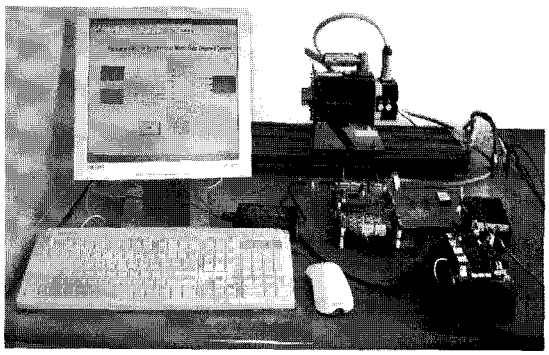

그림 23 영구자석 동기전동기 구동 시스템의 실험 장치
Fig. 23 Experimental setup of PMSM driving system

그림 24, 25, 26은 무부하시 0.2초에서 1000[RPM]으로 가속후 다시 0.6초에서 0[RPM]으로 다시 감속시켜 설계한 프로그램의 속도 및 q축과 d축 전류제어 응답을 확인하여 보았다. 그림에서 보듯이, 실험 결과 설계한 제어기는 속도 및 전류응답이 모두 속도 및 전류 지령을 잘 추종하였다.

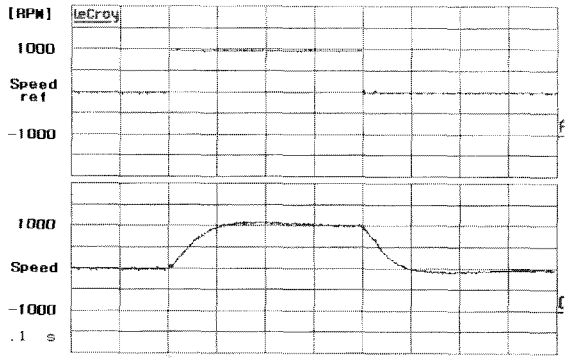

그림 24 속도 제어 응답(0 ↔ 1000[RPM])
Fig. 24 Speed control response(0 ↔ 1000[RPM])

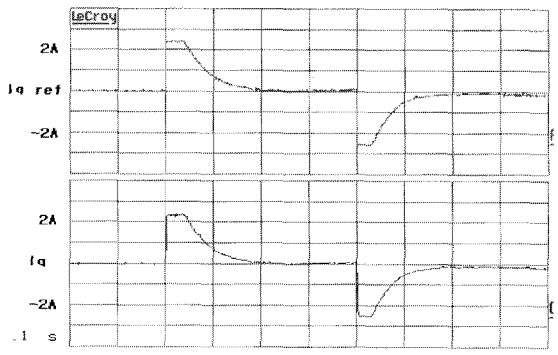

그림 25 q축 전류 제어 응답(0 ↔ 1000[RPM])
Fig. 25 q-axis current control response(0 ↔ 1000[RPM])

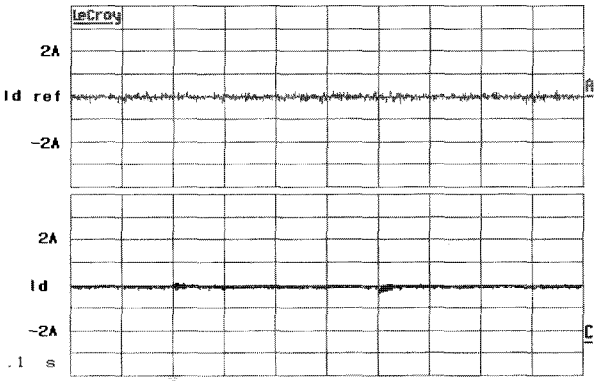

그림 26 d축 전류 제어 응답(0 ↔ 1000[RPM])
Fig. 26 d-axis current control response(0 ↔ 1000[RPM])

또한 그림 27, 28, 29는 관성부하시 속도 지령을 -1000[RPM]에서 1000[RPM]으로 변화시켜 영구자석 동기전동기를 정역으로 회전시켰을 때 설계한 프로그램의 속도 및 q축과 d축 전류제어 응답을 확인하여 보았다. 그림에서 보듯이 속도 제어기의 출력인 q축 전류 지령이 Limiter에 의해 제한된 경우로서 Anti-windup 제어기가 동작하게 된다. 설계한 제어기는 전류 지령을 잘 추종하는 것을 확인하였으며, 속도 응답 또한 0.6s의 정착시간으로 빠르게 추종하는 것을 볼 수 있었다.

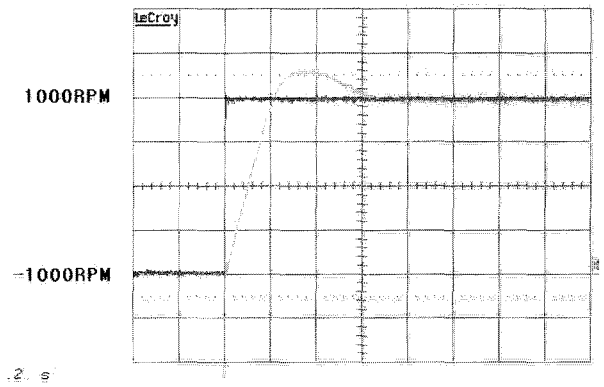

그림 27 속도 제어 응답(-1000 → 1000[RPM])
Fig. 27 Speed control response(-1000 → 1000[RPM])

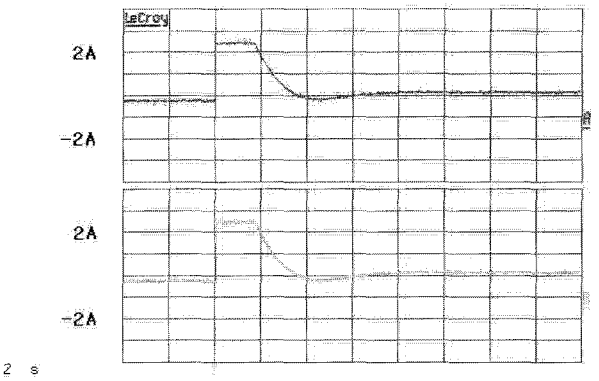

그림 28 q축 전류 제어 응답(-1000 → 1000[RPM])
Fig. 28 q-axis current control response(-1000→1000[RPM])

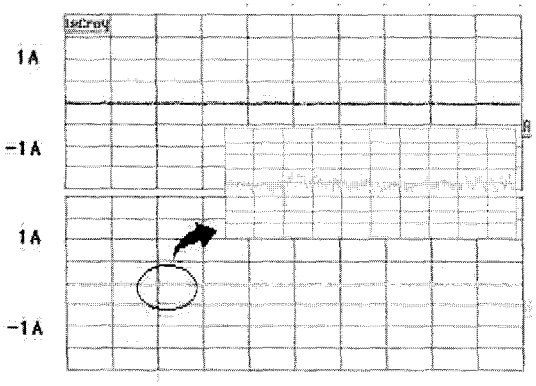

그림 29 d축 전류 제어 응답(-1000 → 1000[RPM])

Fig. 29 d-axis current control response(-1000→1000[RPM])

그림 30은 과변조 구간에서 공간 벡터에서의 출력 전압과 α, β축에서의 출력 전압을 나타낸 것이다. 과변조 구간에서는 지령 전압 벡터가 육각형 전압 제한을 벗어나는 경우로서, 위의 그림에서는 과변조 기법을 사용하지 않는 경우 출력전압이 6각형의 내접원까지 출력되지만, 아래의 그림에서는 설계한 프로그램에서 최소 거리 과변조 기법을 사용하여 출력 전압을 6각형의 전압 제한선상까지 출력할 수 있는 것을 확인할 수 있다.

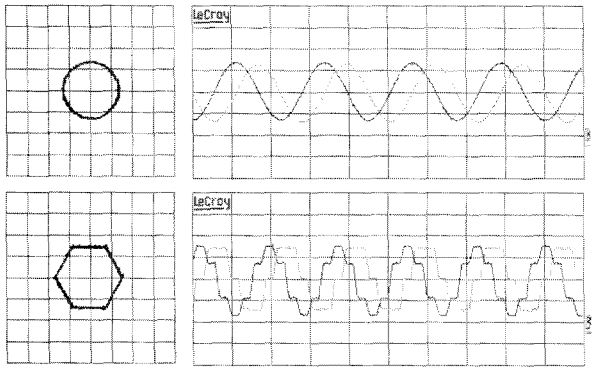

그림 30 공간 벡터 출력 전압과 α, β축에서의 출력 전압

Fig. 30 Output voltages in space vector and α, β axis

7. 결 론

본 논문에서는 MATLAB/SIMULINK의 Embedded Target for TI C2000 DSP 라이브러리를 사용한 모델기반 제어기 설계를 이용하여 표면부착형 영구자석 동기전동기의 벡터 제어를 구현하였고, 설계된 제어 시스템을 0.63[KW] 영구자석 동기전동기에 적용하여 시뮬레이션과 실험을 통하여 타당성을 검증하였다.

TI사의 eZdsp 2812보드와 MATLAB/SIMULINK 프로그램을 이용한 모델 기반 제어기 설계는 기존의 벡터 제어시스템 개발시보다 제어기의 설계 및 구현이 쉽고 수정과 보완이 용이하여 제어시스템 해석 및 설계에 관하여 쉽게 이해하고 단기간에 구현할 수 있었다.

참 고 문 헌

[1] 윤덕용, "공간전압벡터 PWM기법을 이용한 동기전동기의 속도제어," 단국대학교 전기공학과 공학박사학위논문, 1995년 2월.

[2] 김경수, "개선된 비간섭 전류제어기를 이용한 영구자석 동기전동기의 성능 향상에 관한 연구," 부산대학교 전기공학과 공학 석사 학위 논문, 2005년 2월.

[3] 최치영, "TMX320F2812를 이용한 AC서보 모터의 벡터제어에 관한 연구," 호서대학교 정보제어공학과 공학 석사 학위 논문, 2003년 7월.

[4] TechWorld, "Real-Time Embedded World," pp. 52-56, 2007년 2월호.

[5] 조성욱, "적응제어 비간섭 전류 제어기를 이용한 매입형 영구자석 전동기의 제어 성능향상에 관한 연구," 창원대학교 전기전자제어공학과 공학 석사 학위 논문, 2006년 6월.

[6] 설승기, "전기기기제어론," BRAIN KOREA, 2002. pp. 299-352.

[7] 최승필, 이용석, 지준근, "MATLAB/SIMULINK의 모델기반 제어기 설계를 이용한 직류전동기 구동 시스템의 임베디드 제어 시스템," 대한전기학회 하계학술대회 논문집, pp. 1954-1955, 2007. 7.

[8] 이용석, 지준근, 차귀수, "MATLAB/SIMULINK의 TI C2000 DSP 임베디드 타겟을 이용한 동기전동기 구동 시스템," 전력전자학회 하계학술대회 논문집, pp. 400-402, 2007. 7.

저 자 소 개

지 준 근 (池 俊 根)

1964년 8월 13일생. 1986년 서울대 전기공학과 졸업. 1994년 동 대학원 전기공학과 졸업(박사). 현재 순천향대 정보기술공학부 교수

Tel : 041-530-1371

E-mail : jkji@sch.ac.kr

이 용 석 (李 鎔 碩)

1982년 5월 25일생. 2005년 순천향대학교 정보기술공학부 졸업. 2008년 동 대학원 전기통신시스템공학과 졸업(석사).

Tel : 041-557-7905

E-mail : huntes@hanmail.net

차 귀 수 (車 貴 守)

1956년 7월 8일생. 1978년 서울대 공대 공업교육학과 졸업. 1987년 동 대학원 전기공학과 졸업(박사). 현재 순천향대 정보기술공학부 교수

Tel : 041-530-1334

E-mail : gsoocha@sch.ac.kr