

FPGA를 이용한 영구자석 동기 전동기 벡터 제어기의 구현

金石丸^{*}, 林廷圭^{**}, 徐銀敬^{***}, 慎輝範^{****}, 李鉉雨[§], 鄭世教[†]

Implementation of Vector Controller for PMSM Using FPGA

Seok-Hwan Kim, Jeong-Gyu Lim, Eun-Kyung Seo, Hwi-Beom Shin,
Hyun-Woo Lee and Se-Kyo Chung

요 약

고성능 DSP 또는 마이크로프로세서를 통해 구현되던 벡터제어를 프로그램이 가능한 소자인 FPGA를 통해 하드웨어로 구현하였다. 이를 위해 벡터제어 알고리즘을 구성하는 제어 블록들을 VHDL을 통해 모듈화 하고, 모듈화한 벡터제어 알고리즘을 FPGA에 프로그래밍 하여 하드웨어 벡터제어기를 구현하였다. 그리고 하드웨어 벡터제어기의 성능을 검증하기 위해 영구자석 동기 전동기 구동을 위한 벡터제어 시스템을 구성하고, 소프트웨어 기반 벡터제어 시스템과 벡터제어 알고리즘 연산시간 및 성능에 대한 비교연구를 수행하였다.

ABSTRACT

This paper describes a fully hardware realization of vector controller for the permanent magnet synchronous motor (PMSM) using high density field programmable gate arrays (FPGA). In the proposed system, the vector controller including vector transformation, PI regulator, position and speed measurement, current measurement, and space vector PWM blocks is implemented in a FPGA using a VHSIC hardware description language (VHDL). The experimental results using a 1.1kW PMSM are provided to show the validity of the proposed system.

Key Words : Permanent magnet synchronous motor, Vector control, FPGA, VHDL

1. 서 론

최근 급속한 산업자동화로 자동화기기 및 산업기기에 있어 전동기가 널리 사용되고 있다. 특히, 고효율, 고성능 적은 보수비와 같은 특징을 가지는 교류 전동기의 사용이 늘어나고 있으며, 산업자동화 기기의 고성능화를 위한 고성능 제어기가 요구되고 있다.

일반적으로 교류 전동기의 제어 시스템은 크게 전동기 구동을 위한 제어 알고리즘을 내포한 제어기, 전동기 구동에 필요한 전력을 공급하기 위한 전력변환 시스템, 교류 전동기 등으로 구분할 수 있다. 교류 전동기의 정밀제어를 위해서는 복잡한 수식이 포함된 제어 알고리즘이 요구된다. 특히 본 논문에서 구현한 벡터제어 알고리즘은 행렬 연산, sine/cosine 함수 연산이 포함되어 있어 복잡한 연산과정이 요구된다^[2,3]. 따라서 벡터제어 알고리즘을 수행하기 위해서는 복잡한 연산과정으로 인해 오랜 수행시간이 요구되었다. 그로 인해 시스템의 제어 주기 안에 제어 알고리즘을 수행하기 위해 고성능 Digital Signal Processor(DSP) 또는 마이크로프로세서를 통해 소프트웨어적으로 구현되었다. 그리고 벡터제어 알고리즘을 소프트웨어로 구현하

[†]교신저자 : 정희원, 경상대 전기전자공학부 부교수

E-mail : skchung@gsnu.ac.kr

^{*}정희원, LG전자 DA사업부

^{**}학생회원, 경상대 제어계측공학과 박사과정

^{***}학생회원, 경상대 제어계측공학과 석사과정

^{****}정희원, 경상대 전기전자공학부 교수

[§]정희원, 경남대 전자전기공학부 교수

접수일자 : 2005. 12. 28 1차 심사 : 2006. 1. 16

심사완료 : 2006. 2. 6

게 되면 제어기를 설계하는데 있어 상당한 시간을 소프트웨어 공정을 위해 소모하여야 한다. 또한, 적용 시스템에 따라 새로운 소프트웨어가 필요하게 되어 그만큼의 시간적 소모가 야기되었다. 이러한 소프트웨어 기반 벡터제어의 단점을 개선하기 위한 방법으로 하드웨어 기반 벡터제어를 들 수 있다. 하드웨어 기반의 벡터제어는 벡터제어 알고리즘을 하드웨어로 구현하여 별도의 소프트웨어 공정이 필요하지 않으며, 또한 벡터제어 알고리즘을 최적으로 하드웨어 벡터제어 칩을 통해 연산하게 되므로 소프트웨어 기반의 벡터제어 시스템보다 빠른 제어 알고리즘 연산을 수행할 수 있으며, 벡터 제어에 필요한 알고리즘을 모듈화하여 다양한 교류 전동기에 쉽게 적용할 수 있는 장점을 지니고 있다^[9,10].

본 논문에서는 hardware description language인 VHDL을 통해 벡터제어 알고리즘을 적용한 영구자석 동기 전동기를 위한 벡터 제어기를 구현하였다. 벡터 제어기는 Altera사의 Stratix 1S25 FPGA를 이용하여 구현하였으며, 전체적인 벡터제어 시스템은 벡터제어에 필요한 속도, 위치, 전류정보를 얻기 위해 Resolver to Digital Converter (RDC), Hall-effect current sensor, Analog to Digital Converter (ADC)를 외부에 추가하였다. 또한 전동기에 공급되는 전원을 제어하기 위한 인버터로는 Smart Power Module (SPM)을 사용하여 구현하였다. 이렇게 구성된 영구자석 동기 전동기 벡터제어기의 성능을 검증하기 위해 1.1kW급인 Brushless P.M. servomotor를 사용하여 그 성능을 검증하였다. 그 결과 기존 디지털 프로세서를 통해 소프트웨어적으로 구현되던 벡터제어 알고리즘 연산시간을 약 10배 이상 단축하였으며, 교류전동기를 위한 전용 벡터제어 칩 설계가 가능하게 하였다.

2. 교류 전동기 벡터제어 모듈

일반적으로 교류 전동기 벡터제어 시스템은 그림 1과 같이 구성할 수 있다. 교류 전동기 벡터 제어 시스템은 벡터제어 알고리즘을 포함한 제어기, 교류 전동기에 전압을 공급하기 위한 인버터, 교류 전동기 이렇게 크게 3부분으로 구성된다^[4,5]. 본 논문을 통해 구현한 벡터제어기는 벡터 기준좌표를 변환하기 위한 Clark's 변환 블록, Park's 변환 블록, 역 Park's 변환 블록, 속도 제어와 전류 제어를 위한 PI 제어기 블록, 3상 인버터의 게이트 신호를 발생하기 위한 공간벡터 PWM 블록, 외부 소자 인터페이스 블록 등으로 구성

된다. 이와 같이 구성한 벡터제어기의 구성 블록들에 대한 자세한 설명은 다음절에서 논하였다.

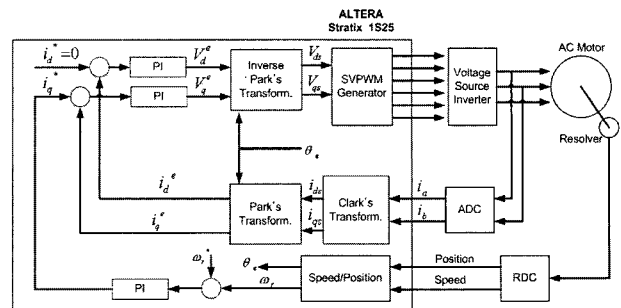


그림 1 영구자석 동기 전동기에 대한 벡터제어 시스템 구성
Fig. 1 Configuration of vector controller system for the PMSM

2.1 위치 및 속도 측정 모듈

교류 전동기 간접 벡터제어를 위해서는 전동기의 회전자 위치정보와 속도정보는 필수적이다. 본 논문에서 구현한 위치 및 속도 측정 모듈은 리졸버를 장착하고 있는 교류 전동기에 적합하도록 구현하였다. 또한 리졸버에서 발생되는 아날로그 신호를 디지털 신호로 변환하기 위한 RDC는 전용 소자를 사용하여 벡터제어기 외부에 설치하고, 위치 및 속도 측정 모듈 내부에 RDC 인터페이스 블록을 첨가하였다.

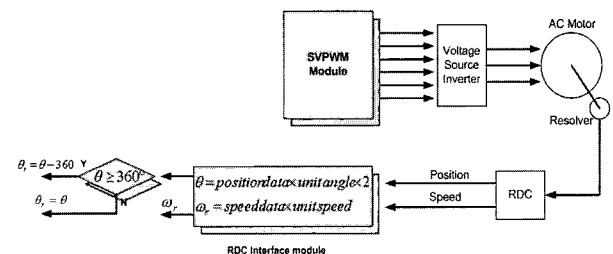


그림 2 위치 및 속도 측정 모듈 블록도
Fig. 2 Block diagram of position and speed measurement module

위치 및 속도 측정 모듈은 그림 2와 같이 구성하였다. 구성된 모듈은 매 샘플링 시간마다 RDC 인터페이스 블록을 통해 RDC 제어신호를 출력하고, RDC를 통해 검출한 회전자의 위치정보와 속도정보를 입력받는다. 이와 같이 입력된 위치정보와 속도정보는 최소비트에 대한 분해능을 곱하여 실제 회전자의 위치와 속도를 얻는다. 그리고 본 논문의 검증을 위해 사용한 영구자석 동기 전동기는 기계적으로 4-pole로 구성되

어 있어 회전자가 1회전하는데 4π 의 정현파가 필요하다. 따라서 그림 2와 같이 입력된 위치정보를 2배하여 실험에 사용한 영구자석 동기 전동기의 위치정보를 검출하였다.

2.2 전류 측정 모듈

교류 전동기의 간접 벡터제어에 있어 전류정보는 회전자의 위치정보와 속도정보와 함께 필수적인 정보로 사용된다. 본 논문에서는 정동기의 각 상전류를 측정하기 위해 홀 센서(Hall-sensor)와 홀 센서에서 출력되는 전압신호를 디지털 신호로 변환하기 위한 ADC를 전용소자를 사용하여 벡터제어기 외부에 구성하고, ADC를 제어하기 위한 ADC 인터페이스 블록을 전류 측정 모듈에 첨가하였다.

그림 3과 같이 전류 측정 모듈을 구성하였다. 이와 같이 구성된 모듈은 위치 및 속도 측정 모듈과 같이 매 샘플링 시간마다 ADC 제어신호를 발생하고, ADC는 아날로그 신호를 디지털 신호로 변환한다. ADC는 매 샘플링 시간마다 a상과 b상에 부착된 홀 센서의 기준전압과 전류정보에 대한 전압을 디지털 신호로 변환하여 전류 측정 모듈에 전달한다. ADC로부터 입력받은 디지털 정보를 바탕으로 전류 측정 모듈은 입력된 디지털 정보에 최소비트 분해능을 곱하여 실제 전류값으로 변환한다.

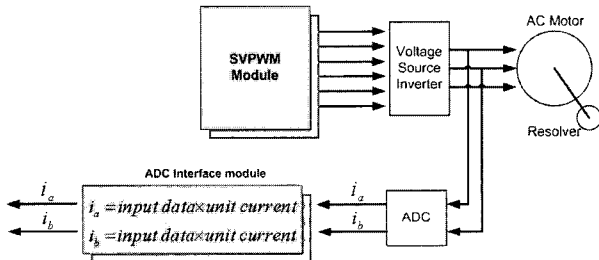


그림 3 전류 측정 모듈 블록도
Fig. 3 Block diagram of current measurement module

2.3 벡터 변환 모듈

측정된 전동기의 위치 및 속도 정보와 전류 정보를 바탕으로 교류 전동기를 제어하기 위해서는 고정 좌표계를 기준으로 하여 여자전류 성분과 토크전류 성분을 제어하는 것보다 동기 좌표계를 기준으로 하여 제어하는 것이 더욱 효율적이다. 그로 인해 그림 4와 같이 측정된 전동기 전류 정보를 동기 좌표계의 값으로 변환하기 위해서 Clark's 변환 모듈과 Park's 변환 모듈이 필요하며, 제어기를 통해 얻어진 제어정보를 다시

고정 좌표계로 변환하기 위해 역 Park's 변환 모듈이 필요하다^[1-3]. 또한 Park's 변환과 역 Park's 변환에 포함된 sine/cosine 함수를 수행하기 위해서 별도의 Lookup table(LUT)로 벡터 변환 모듈이 구성된다.

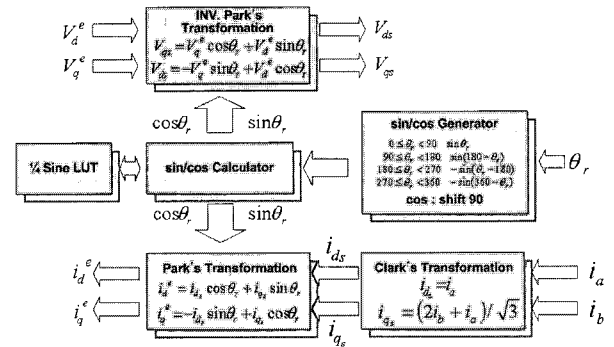


그림 4 벡터 변환 모듈 구성
Fig. 4 Configuration of vector transformation modules

이와 같이 구성된 벡터 변환 모듈들의 역할은 다음과 같으며, Park's 변환과 역 Park's 변환의 벡터 변환은 그림 5와 같다.

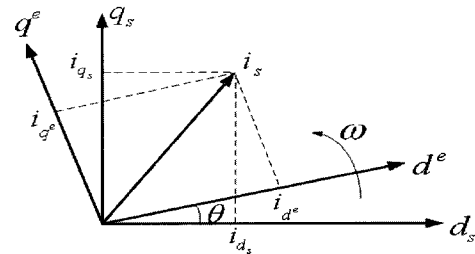


그림 5 벡터회전 기준 좌표
Fig. 5 Reference frame of vector rotation

가. sine/cosine 계산 모듈 : Park's 와 역 Park's 변환은 행렬 연산에 sine과 cosine 함수를 포함하고 있다. 일반적으로 소프트웨어 기반 벡터제어기에서는 소프트웨어 라이브러리를 이용하여 손쉽게 삼각함수를 연산할 수 있으나, 오랜 연산시간을 소모하는 단점을 가지고 있다. 반면 하드웨어를 통한 sine, cosine함수 연산은 테일러급수, Coordinate Rotation by Digital Computer(CORDIC), LUT 등을 사용할 수 있다. 그러나 테일러급수는 부동소수점 연산과 오랜 반복연산이 필요하며, CORDIC 알고리즘 또한 오랜 반복연산 작업이 필요하므로 빠른 연산이 필요한 시스템에 적합하지 않다. 따라서 본 논문에서는 빠른 연산 능력을 가

진 sine-LUT와 sine/cosine 전처리 부분으로 구성하였다.

나. Clark 변환 모듈 : 전류센서를 통해 측정된 3상 전동기의 고정자 전류정보를 여자전류 성분 I_q 와 토크 전류 성분 I_d 로 분해하는 역할을 수행하며, 3상 좌표를 2축 좌표로 변환하는 Clark's 변환은 다음과 같다.

$$\begin{bmatrix} f_{ds} \\ f_{qs} \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} f_{as} \\ f_{bs} \\ f_{cs} \end{bmatrix} \quad (1)$$

그리고 대칭 3상 시스템에서 순시 전류는 0이므로 다음과 같이 나타낼 수 있다.

$$i_{as} + i_{bs} + i_{cs} = 0 \quad (2)$$

식 (1)과 식 (2)를 통해 다음과 같이 단순화된 Clark's 변환을 얻을 수 있다.

$$\begin{bmatrix} f_{ds} \\ f_{qs} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \end{bmatrix} \quad (3)$$

Clark 변환 모듈은 식 (3)을 바탕으로 그림 4와 같이 구성되어 내부적으로 2x2 행렬을 연산하도록 하였다.

다. Park 변환 모듈 : 고정좌표축 값을 동기좌표축 값으로 변환하는 역할을 하는 모듈로서, Park's 변환은 다음과 같다.

$$\begin{bmatrix} f_{d}^e \\ f_{q}^e \end{bmatrix} = \begin{bmatrix} \cos\theta_r & \sin\theta_r \\ -\sin\theta_r & \cos\theta_r \end{bmatrix} \begin{bmatrix} f_{ds} \\ f_{qs} \end{bmatrix} \quad (4)$$

Park 변환 모듈은 식 (4)를 바탕으로 Clark 변환 모듈에서 변환한 고정좌표에서의 여자전류와 토크전류 성분을 동기좌표에 대한 여자전류와 토크전류로 변환하기 위한 2x2 행렬 연산을 내부적으로 수행한다.

라. 역 Park 변환 모듈 : 동기좌표축 값을 고정좌표축 값으로 변환하는 역할을 하는 모듈이며, 역 Park's 변환은 다음과 같다.

$$\begin{bmatrix} f_{ds} \\ f_{qs} \end{bmatrix} = \begin{bmatrix} \cos\theta_r & -\sin\theta_r \\ \sin\theta_r & \cos\theta_r \end{bmatrix} \begin{bmatrix} f_{d}^e \\ f_{q}^e \end{bmatrix} \quad (5)$$

역 Park 변환 모듈은 식 (5)를 바탕으로 제어기를 통해 제어된 동기좌표에서의 지령전압을 고정좌표축에 대한 지령전압으로 변환하기 위한 2x2 행렬 연산을 모듈 내부적으로 수행한다.

2.4 비례-적분(PI) 제어기 모듈

비례-적분 제어기 모듈은 한 개의 속도 PI 제어기와 두 개의 전류 PI 제어기로 구성되어 있다. 속도제어 루프에 포함된 PI 제어기는 q축 토크 전류의 기준 전류를 발생하는데 사용되며, 전류제어 루프에 포함된 PI 제어기는 각각 d축과 q축에 대한 전압 명령을 발생하는 역할을 한다.

본 논문에서 구현한 PI 제어기 모듈은 다음과 같이 s-평면에 대해 나타낼 수 있다.

$$y(s) = \left(K_p + \frac{K_i}{s} \right) \{ x^*(s) - x(s) \} = \left(\frac{K_p s + K_i}{s} \right) e(s) \quad (6)$$

여기서, $x^*(s)$ 는 기준값, $x(s)$ 는 피드백 값, $e(s)$ 는 오차를 나타내며, K_p 와 K_i 는 각각 비례이득과 적분이득을 나타낸다.

식 (6)과 같이 s-평면에서 설계한 제어기를 이산시간영역으로 변환하기 위해 backward transform을 사용하면 다음과 같이 식 (6)을 변환할 수 있다^[8].

$$y(n) = y(n-1) + K_p \{ e(n) - e(n-1) \} + K_i T_s e(n) \quad (7)$$

식 (7)과 같이 이산영역으로 변환한 제어기와 제한기를 추가한 비례-적분 제어기 모듈의 구성은 그림 6과 같다.

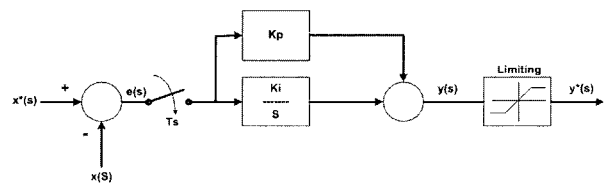


그림 6 비례-적분 제어기 모듈의 블록도
Fig. 6 Block diagram of PI regulator modules

2.5 섹터 검출 모듈

인버터 게이트 신호를 변조하는 공간벡터 PWM 모듈에서 섹터에 따라 변조되는 게이트 신호가 바뀌기 때문에 섹터 정보는 필수적으로 필요하다. 이러한 섹터 정보는 전류 PI 제어를 통해 얻어지는 전압명령의 위상각을 통해 얻어진다. 그리고 전압명령의 위상

각을 얻기 위해서는 arctangent 연산을 하여야 하며, 하드웨어를 통한 arctangent 연산은 테일러급수, LUT, CORDIC 알고리즘을 통해 얻어진다. 그러나 Taylor series와 LUT는 하드웨어 구현이 쉽지 않다. 반면 CORDIC 알고리즘을 통한 arctangent 연산은 벡터 변환에서와 달리 보다 간단하게 구현할 수 있다^[9,11].

CORDIC 알고리즘은 Rotating mode와 Vectoring mode, 이렇게 두 가지 모드가 있다. Rotating mode는 벡터 회전에 응용할 수 있어 앞서 다루었던 벡터 변환 모듈에 적용할 수 있다. 그러나 어느 정도의 정확도를 얻기 위해서는 많은 반복연산이 필요하므로 긴 연산시간이 요구된다. 반면 Vectoring mode는 벡터 크기와 사이 각을 계산하는데 사용되며, 반복연산은 최대 10 회만으로 높은 정확도를 얻을 수 있어, 본 논문에서 섹터 정보를 얻기 위해 CORDIC 알고리즘을 이용하여 전압명령에 대한 위상각을 구하였다.

CORDIC 알고리즘의 Vectoring mode는 y가 m번 반복 연산한 결과가 0으로 수렴하면 z값은 $\tan^{-1}(y/x)$ 의 값을 가지며, 알고리즘은 다음과 같다.

$$\begin{aligned} x^{(i+1)} &= x^{(i)} - d_i y^{(i)} 2^{-i} \\ y^{(i+1)} &= y^{(i)} + d_i x^{(i)} 2^{-i} \\ z^{(i+1)} &= z^{(i)} - d_i \tan^{-1}(2^{-i}) \\ \text{where, } y^{(i)} < 0 &\rightarrow d_i = 1 \\ y^{(i)} \geq 0 &\rightarrow d_i = -1 \end{aligned} \quad (8)$$

식 (8)이 m번 반복 연산한 결과는 다음과 같다.

$$\begin{aligned} x^{(m)} &= K\sqrt{x^2 + y^2} \\ y^{(m)} &= 0 \\ z^{(m)} &= z_0 + \tan^{-1}(y/x) \end{aligned} \quad (9)$$

여기서, K는 알고리즘 이득을 나타낸다.

식 (8)과 식 (9)를 바탕으로 구현한 섹터 검출 모듈을 그림 7과 같이 입출력 로직, 반복연산 제어 블록, 반복연산 블록, LUT로 구성되어 있다. 구성된 섹터 검출 모듈은 전류 PI 제어를 통해 발생한 전압명령을 입력으로 취하고, 반복연산 제어 블록을 통해 y값을 검사하여 0으로 수렴하면 반복 연산을 중단하고 결과값을 출력하게 된다.

2.6 공간벡터(Space Vector) PWM 모듈

SVPWM은 d축과 q축의 전압명령을 통해 전압원 인버터(voltage source inverter)를 제어하는 6개의 계

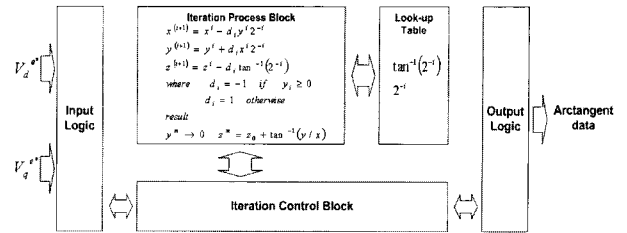


그림 7 섹터 검출 모듈 구성
Fig. 7 Configuration of sector detector module

이트 신호를 발생하는 역할을 한다^[6,12,13]. 3상 인버터가 조합할 수 있는 스위칭 패턴을 2차원 평면에 나타내면 그림 8과 같이 유효전압 벡터(V1~V6)와 제로 벡터(V0,V7)로 나타낼 수 있다.

SVPWM의 기본원리는 한 주기 내에 지령전압 벡터와 평균적으로 동일한 전압을 인버터의 유효전압 벡터와 제로 벡터를 이용하여 합성해 내는 것이다. 즉 지령전압 벡터와 인접한 두 유효전압 벡터와 제로 벡터를 이용하여 평균적으로 지령전압과 동일한 전압을 발생시키는 것이다. 지령전압 벡터와 인접한 두 유효전압 벡터의 인가시간은 다음과 같이 얻을 수 있다.

$$\begin{aligned} X &= V_{qs} \\ Y &= \frac{1}{2}(\sqrt{3}V_{ds} + V_{qs}) \\ Z &= \frac{1}{2}(-\sqrt{3}V_{ds} + V_{qs}) \end{aligned} \quad (10)$$

여기서, V_{ds} 와 V_{qs} 는 지령전압 벡터의 d축과 q축 값.

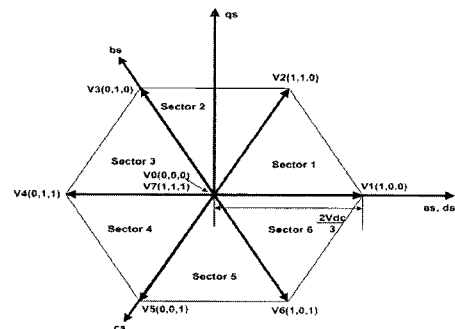


그림 8 SVPWM의 공간 전압 벡터도
Fig. 8 Space voltage vectors of SVPWM

인접한 두 유효전압 벡터의 인가시간은 섹터정보에 따라 식 (10)중 2개의 수식을 사용한다. 이렇게 얻어진 유효전압 벡터 인가시간을 바탕으로 실제 3상 전압원 인버터의 게이트 신호 인가시간은 다음 수식을 통해서 얻을 수 있다.

$$\begin{aligned}
 t_{on} &= \frac{(T_s/2) - t_1 - t_2}{2} \\
 t_{bn} &= t_{an} + t_1 \\
 t_{cn} &= t_{bn} + t_2
 \end{aligned}
 \tag{11}$$

여기서, T_s 는 샘플링 시간을 나타낸다.

3상 전압원 인버터의 게이트 신호 인가시간은 식 (11)을 섹터정보에 따라 배치하여 얻을 수 있다. 이와 같이 얻어진 게이트 신호 인가시간은 인버터 게이트의 arm short를 방지하기 위해 데드 타임(dead time)을 삽입한다. 데드 타임을 삽입한 게이트 신호 인가시간은 삼각 대칭 캐리어와 비교하여 실제 3상 전압원 인버터 게이트 신호를 변조해 낸다. 이와 같은 공간벡터 PWM 모듈의 게이트 신호 변조 과정은 그림 9와 같다.

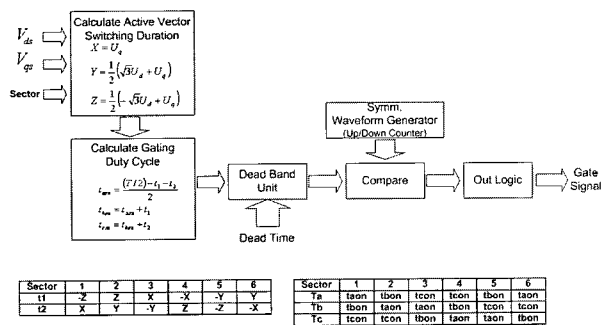


그림 9 공간벡터 PWM 모듈의 구성
Fig. 9 Configuration of SVPWM module

3. 시스템 구성 및 실험 결과

본 논문에서 구현한 PMSM을 위한 벡터제어 시스템은 그림 10과 같이 벡터 제어기, DC link단, 인버터 회로, 전류 및 속도, 위치 측정회로로 구성되었다. 벡터 제어 알고리즘을 하드웨어로 구현한 벡터제어기는 Altera사의 Quartus II 소프트웨어 환경하에서 VHDL을 통해 벡터제어 블록들을 구현하였다. VHDL을 통해 구현된 벡터제어 블록들은 Quartus II를 이용하여 symbol로 변환하고, symbol로 변환한 벡터제어 블록을 역시 Quartus II를 이용하여 그림 11과 같이 각각의 symbol들을 결선하여 벡터제어기를 설계하였으며, 이렇게 설계한 벡터제어기의 사양은 표 1과 같다. 이와 같이 구현한 벡터제어기는 Altera사의 Stratix EP1S25F672 FPGA를 사용하여 구현하였으며, DC link단은 full-bridge rectifier와 평활용 콘덴서로 구성하였다. 그리고 전력 변환회로는 6개의 IGBT와 게이

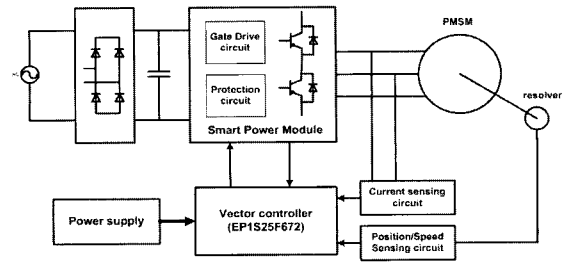


그림 10 PMSM 제어 시스템 구성
Fig. 10 Configuration of PMSM control system

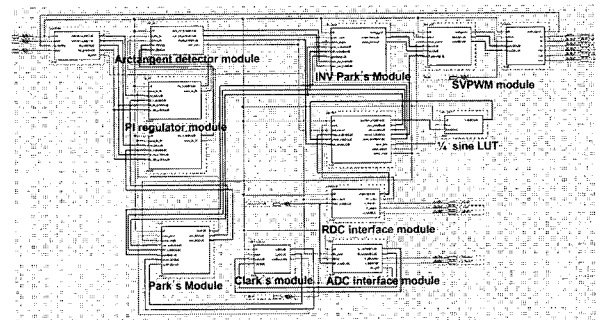


그림 11 벡터제어기 회로도
Fig. 11 Schematic diagram of the vector controller



그림 12 실험시스템
Fig. 12 Photograph of the experimental system

트 드라이브 회로, 보호회로 등을 하나의 칩에 집적화한 Fairchild사의 Smart Power Module(SPM)을 사용하여 구성하였다. 그리고 전류 정보를 얻기 위해 전류 검출 회로, 속도와 위치 검출 회로, 시스템 구동을 위한 직류 전압원 등으로 구성하였다. 또한 구현한 벡터제어기의 성능을 검증하기 위하여 최대 정격속도가 7500 RPM이며, 4극 1.1kW급인 PMSM을 사용하였다. 전체시스템은 그림 12와 같다.

벡터제어기의 system clock은 50MHz를 사용하였으며, 스위치 주파수와 샘플링 주파수는 10kHz를 사용하였다. 또한 벡터제어기의 data word는 16bit를 사용하였으

며, PWM resolution과 dead time resolution 또한 16bit를 사용하였다. 그리고 벡터제어 알고리즘에 포함된 부동소수점 연산을 수행하기 위해 Q-math를 하드웨어적으로 사용하였다.

그림 13은 벡터제어기를 통해 변조된 게이트 파형을 나타내며, 그림 14는 벡터제어기를 통해 변조된 게이

표 1 EP1S25F672를 통해 구현한 벡터제어기 사양
Table 1 Design summary of the vector controller with EP1S25F672

EP1S25F672	Resource	Design usage
Logic elements	25660	4171 (16%)
DSP blocks	80	58 (72%)
Pins	474	28 (5%)

표 2 하드웨어와 소프트웨어 연산시간 비교
Table 2 Compare with H/W and S/W calculation time

System	EP1S25F672(H/W)	F2812(S/W)
Calculation time	7 μ s(at 50MHz)	100 μ s(at 150MHz)

트 신호와 이에 따른 실제 전동기에 흐르는 상전류를 나타낸다. 그리고 그림 15는 속도제어를 통해 상전류를 나타내며, 그림 16은 전동기의 회전방향을 전환했을 때 상전류를 나타낸다.

그리고 하드웨어로 구현한 벡터제어기의 성능을 검증하기 위해 소프트웨어 기반의 벡터제어기와의 알고리즘 연산시간의 비교 결과 표 2와 같다.

4. 결 론

본 논문에서는 벡터제어알고리즘을 내장한 벡터제어기를 구현하고 벡터제어 시스템을 구성하여 그 성능을 검증하였다. EP1S25F672 FPGA를 이용하여 구현한 벡터제어기는 VHDL을 통해 구현된 벡터 변환 모듈, 위치 및 속도 측정 모듈, 전류 측정 모듈, 비례-적분 제어기 모듈, 외부 소자 인터페이스 모듈을 하드웨어화하여 내장하였다. 또한 구현된 하드웨어 블록은 모듈화되어 있어 다양한 교류 전동기에 사용할 수 있도록 하였다.

또한 구현한 벡터제어기를 사용한 벡터제어 시스템은 실험을 통해 표 2와 같이 소프트웨어 기반의 벡터제어기에 비해 약 10배 이상의 연산시간을 단축하였다. 또한 PLL회로를 추가하여 system clock을 향상시키면 더 빠른 연산시간을 기대할 수 있다.

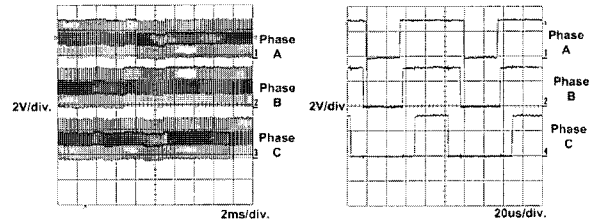


그림 13 벡터제어기를 통해 변조된 게이트 파형
Fig. 13 modulated PWM waveform by vector controller

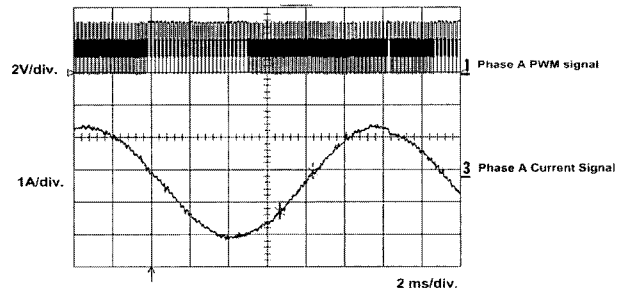


그림 14 a상 게이트 파형 및 전류 파형
Fig. 14 Phase a gate waveform and current waveform

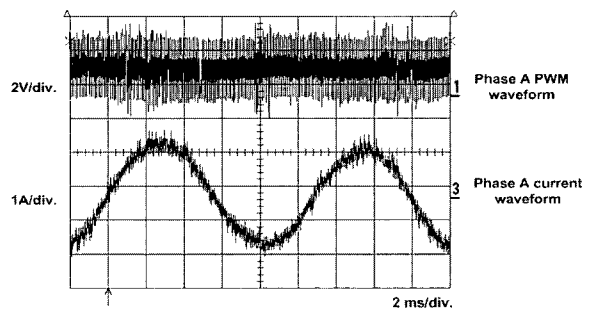


그림 15 기준속도 3000rpm일 때 게이트 및 전류 파형
Fig. 15 Phase a gate waveform and current waveform when reference speed is 3000rpm

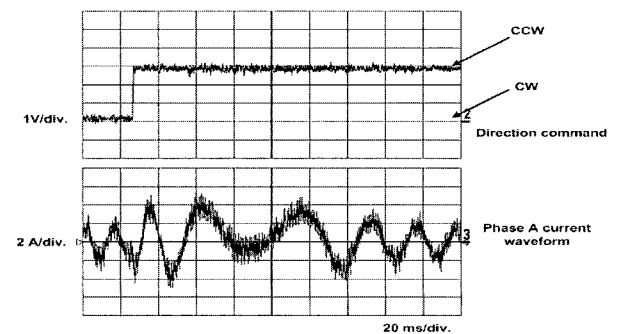


그림 16 회전방향에 따른 a상 전류 파형
Fig. 16 Direction command and phase a current waveform

따라서 본 논문에서와 같이 소프트웨어 기반의 벡터 제어를 하드웨어로 구현하여 얻을 수 있는 장점은 하드웨어를 통해 복잡한 벡터 제어 알고리즘을 연산하기 때문에 최적 연산을 통해 연산시간을 단축시킬 수 있다. 또한 하드웨어로 구현한 벡터 제어를 사용하여 설계과정에 있어 제어 알고리즘의 소프트웨어 설계공정을 제거할 수 있다. 마지막으로 하드웨어 모듈을 라이브러리화하면 다양한 교류 전동기 제어에 있어 특정 목적의 제어를 쉽게 설계할 수 있는 장점이 있다.

이 논문은 산업자원부에서 시행한 전력산업 인프라 구축지원사업의 지원에 의하여 수행된 연구결과입니다.

참 고 문 헌

- [1] Paul C. Krause, "Analysis of Electric Machinery", McGraw-Hill, 1987.
- [2] Bimal K. Bose, "Modern Power Electronics and AC Drives", Prentice Hall, 2002.
- [3] R. Krishnan, "Electric Motor Drives, Modeling, Analysis, and control", Prentice Hall, 2001.
- [4] Ion Boldea, Syed A. Nasar, "Vector Control of AC Drives", CRC press, 1992.
- [5] Peter Vas, "Vector Control of AC Machines", Oxford Science Publication, 1990.
- [6] 설승기, "전기기기제어론", 브레인코리아, 2002.
- [7] 정동화, "전력전자 및 전동기 제어", 인터비전, 2005.
- [8] Benjamin C. Kuo, "Digital Control Systems", Saunders College Publishing, 1992.
- [9] Zhaoyong Zhou, Tiejai Li, T. Takahashi, E. Ho, "FPGA Realization of a High-performance Servo Controller for PMSM", Applied Power Electronics Conference and Exposition, Vol. 3, 2004, pp.1604~1609.
- [10] Y. S. Kung, P. G. Huang, C. W. Chen, "Development of a SOPC for PMSM Drives", IEEE International Midwest Symposium on Circuit and Systems, Vol 2, 2004, pp.329~332.
- [11] Ray Andraka, "A survey of CORDIC algorithms for FPGA based computers", Proc. of ACM/SIGDA sixth Interantional Symposium on FPGAs, 1998, pp.191~200.
- [12] D. Grahame Holmes, Thomas A. Lipo, "Pulse Width Modulation for Power Converters", IEEE press, 2003.
- [13] Texas Instruments, "Digital Motor Control", Oct. 2003.

저 자 소 개



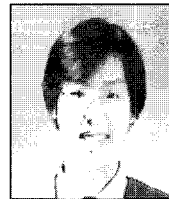
김석환(金石丸)

1977년 5월 1일생. 2004년 경상대 공대 전기전자공학부 졸업. 2006년 2월 경상대학교 제어계측공학과 졸업(석사). 2006년 2월~LG전자 DA사업부 연구원.



임정규(林廷圭)

1977년 6월 4일생. 2003년 경상대 공대 전기전자공학부 졸업. 2005년 동 대학원 제어계측공학과 졸업(석사). 2005년~현재 동 대학원 제어계측공학과 박사과정.



서은경(徐銀敬)

1982년 7월 27일생. 2005년 경상대 공대 전기전자공학부 졸업. 2005년~현재 동 대학원 제어계측공학과 석사과정.



신휘범(愼輝範)

1982년 서울대 전기공학과 졸업. 1985년 한국과학기술원 전기 및 전자공학과 졸업(석사). 1992년 동 대학원 전기 및 전자공학과 졸업(공학박). 1990년~1993년 현대전자(주) 선임연구원. 2000년~2002년 미국 University of Wisconsin Madison 방문교수. 1993년~현재 경상대 전기전자공학부 교수.



이현우(李鉉雨)

1953년 4월 28일생. 1979년 동아대 전기제어공학과 졸업. 1984년 영남대 대학원 전기공학과 졸업(석사). 1992년 동아대 대학원 전기공학과 졸업(공학박). 1993년 일본 오사카공업대 객원연구원. 1985년~현재 경남대 전자전기공학부 교수 및 중앙인사위원회 비상임 위원.



정세교(鄭世敎)

1966년 11월 26일생. 1989년 경북대 전자공학과 졸업. 1992년 한국과학기술원 전기 및 전자공학과 졸업(석사). 1997년 동 대학원 전기 및 전자공학과 졸업(공학박). 1997년~현재 경상대 전기전자공학부 부교수 및 공학연구원 연구원. 1999년 12월~2000년 2월 일본 큐슈공업대 객원교수. 2002년 2월~2003년 2월 미국 Texas A&M 대학 Visiting Scholar. 당 학회 편집위원.