

# 유도전동기 벡터제어를 위한 Simulink/RTW 기반 실시간 제어시스템 개발

論 文

50D-3-5

## Development of a Simulink/RTW-Based Realtime Control System for an Induction Motor Vector Control

康 文 浩\*  
(Moon-Ho Kang)

**Abstract** - In this research a Simulink/RTW-based realtime control system was developed for an induction motor vector control. On the Simulink window, the control system is designed in the form of block diagrams, program codes are produced automatically with the RTW(Real Time Workshop), then an DSP c compiler compiles the program codes. With this automatic program producing method rapid prototyping is realized with the least time-consuming manual programming procedures. To show effectiveness of the proposed system designing scheme a DSP-based induction motor vector controller was constructed and implemented

**Key Words** : Simulink, RTW, DSP, induction motor, vector control

### 1. 서 론

제어이론과 제어기 설계기술이 발전함에 따라 제어시스템이 더욱 복잡 다양화되어 미분 방정식이나 행렬식 등을 이용하여 제어시스템을 일일이 수동으로 해석 및 설계하는 것은 매우 어렵고 많은 시간이 소요된다. 또한 복잡한 계산뿐만 아니라 다양한 그래프 및 시각화들을 포함하는 사용자 인터페이스 작업이 일반적으로 요구되고, 제어시스템 설계과정에서 필수적으로 동일한 일의 반복작업이 포함되는 특성을 가지기 때문에, 종래의 방법인 Fortran, C, C++ 및 기타 프로그래밍 언어들을 이용하여 직접 프로그래밍 하는 방식은 용이한 작업이 아니고 많은 시간을 들여야 하므로 비효율적이다. 본 연구에서는 이러한 문제점을 해결하기 위해서 제어시스템 설계, 시뮬레이션 및 실험을 위한 실행파일 생성과 구현까지의 전과정이 최소한의 프로그램 코딩과 반복작업으로 수행되는 실시간 제어시스템을 개발하였다. 시스템 개발 과정은 다음과 같은 세단계로 수행된다.

첫 번째 단계에서는 시뮬링크(Simulink)[1] 윈도우상에서 시뮬링크 블록들과 S-Function[2]으로 설계된 사용자가 생성한 블록들을 이용하여 블록 다이어그램 형태로 제어시스템을 설계한다. 두 번째 단계는 시스템 검증단계로, 첫 번째 단계에서 생성된 제어시스템에 대해서 시뮬링크를 이용하여 PC상에서 시뮬레이션을 행하여 입력조건들에 대해 각 시스템 블록들에

서 올바른 출력이 발생하는지를 확인한다. 세 번째 단계는 실험을 위한 자동 C코드 생성 및 구현단계로, RTW(Real Time Workshop)[3]를 이용하여 두 번째의 시뮬레이션 단계에서 검증된 동일한 시스템 블록에 대해서 C코드를 자동 생성시킨 후, DSP C컴파일러를 이용하여 컴파일과 링크를 수행하고 실행파일을 생성한다. 생성된 실행파일은 PC의 병렬포트를 통해 DSP(TMS320C32)[4]보드로 다운로드 되어져 제어작업이 수행된다. 한편, 제어작업중 발생하는 데이터들은 Matlab의 mat파일[3]형태로 변환되어 병렬포트를 통해 PC로 업로드되어져 PC상에서 확인될 수 있도록 하였다.

상기 본 연구의 시스템개발 환경에 의하면, 윈도우상에서 제어시스템 블록이 구성되어 시뮬레이션 검증이 된 이후에는, 별도의 수정없이 컴퓨터 시뮬레이션에서 사용된 동일한 블록 으로부터 실험을 위한 C코드가 자동으로 생성되고, 제어시스템을 변동시키는 경우에도 윈도우상에서 간단히 블록의 파라미터나 블록상호간의 연결상태만을 바꾸면 되기 때문에 시스템개발과 변동시 소요되는 시간을 최소화시킬 수 있다.

개발된 시스템의 유용성을 보이기 위해서 DSP와 IPM (PM100CVA060)[5]을 이용한 유도전동기 벡터제어[6],[8],[9],[10] 장치를 제작하고 시뮬링크 윈도우상에서 제어시스템을 설계한 후, 실험을 행하여 시스템 성능을 고찰하였다.

### 2. 시스템 구성

#### 2.1 소프트웨어 구성

제어시스템의 rapid prototyping을 위한 가장 중요한 요소는 시스템 설계시 필요한 제어알고리즘, 제어결과 분

\* 正 會 員 : 鮮 文 大 工 大 機 械 · 制 御 工 學 部 助 教 授 · 工 博  
接 受 日 字 : 2000年 10月 20日  
最 終 完 了 : 2001年 2月 2日

석을 위한 데이터의 처리 및 호스트(컴퓨터)와 플랜트(DSP)사이의 통신 등을 구현하기 위한 프로그램 코드를 자동으로 생성, 컴파일, 링크하므로써 기존에 수작업에 의한 처리와 비교하여 시스템 개발에 소요되는 시간을 대폭 줄이는 기능이다. 본 연구에서는 MathWork사의 시뮬링크(Simulink(V3))[1]와 RTW(RealTimeWorkshop)(V2.1)[3]를 이용하여 시스템 설계시 필요한 코드들을 자동생성한 후, 생성된 C코드들과 TI 사의 C-컴파일러를 연계시켜 컴파일, 링크시키고 DSP에서 구동되는 실행파일을 생성한다. 이러한 일련의 과정에서 사용되는 소프트웨어 기능에 대해 간략히 설명하면 다음과 같다.

- ①Simulink : 시스템을 구성하는 제어알고리즘, 사용자가 작성한 I/O인터페이스부, 인터럽트처리부 및 모듈실행부, 실험데이터 입/출력부 등이 각각 블록형태로 시뮬링크 윈도우상에서 설계된다.
- ②RTW : ①과정에서 생성되는 시스템블록파일(예를 들어 fn.mdl)이 지나는 모든 블록정보를 취합한 후, Target system(TMS320C32[4])에 대응하는 특정한 TLC(Target Language Compiler)[3]를 호출하여 Target system에서 구현될 수 있는 C코드들 생성한다.
- ③nmake : ②과정에서 생성된 C코드들, 사용자가 작성한 S-function[2] C코드 및 기타 C코드들을 컴파일, 어셈블, 링크시킨 후, 실행파일을 생성한다. 이때, nmake는 DSP 컴파일러가 제공하는 응용프로그램들을 호출해 가면서 컴파일, 어셈블, 링크한 후, 최종적으로 실행파일(fn.out)로부터 DSP보드에 다운로드 가능한 파일형식을 갖는 헥사파일(fn.hex)을 생성한다.
- ④VC++로 작성된 응용프로그램 : 헥사파일을 PC로부터 DSP로 전송하고 전송된 파일을 실행시키며, 실험이 종료되면 DSP로부터 PC로 실험데이터들을 업로드시켜 실험데이터를 분석 가능하게 하는 사용자가 생성하는 응용파일.

상기한 프로그램들이 상호 연결되어 전체작업이 수행되는 과정을 보이면 그림 1과 같다.

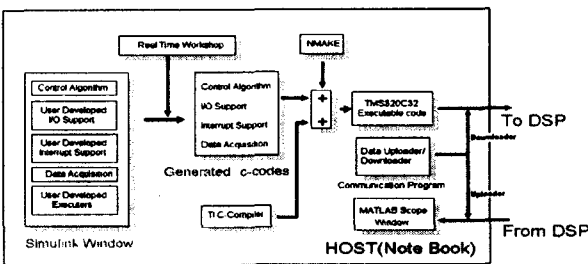


그림 1 자동코드생성, 컴파일, 링크작업 흐름도.  
Fig. 1 Automatic code generation, compile and link operations flow.

## 2.2 하드웨어 구성

실험에 사용된 하드웨어의 구성을 보이면 그림 2와 같다. 시뮬링크, RTW와 C-컴파일러를 비롯한 시스템 설계에 필요한 프로그램들은 PC상에 탑재된다. RTW에 의해 C-코드들을 생성한 후 이를 컴파일, 링크시킨 후 발생하는 실행 이진 파일

은 노트북의 프린터 포트를 통해 ADC, 카운터, 인버터구동을 위한 공간벡터 PWM발생부등의 I/O와 메모리, 각종 보호회로와 주변장치 등을 갖는 DSP보드(TI, TMS320C32)로 다운로드된다. 유도전동기의 속도와 전류정보는 각각 속도카운터와 ADC를 통해 DSP의 내부 타이머에 의해 설정되는 일정 샘플링시간 마다 DSP측으로 귀환되어 속도 및 전류제어 알고리즘에 따른 제어입력 계산에 사용된다. 이 값은 공간벡터PWM방식[7]에 의해 전압변조된 후, 프로그래머블 타이머에 의해 IPM (MITSUBISHI, PM150CVA060)[5]을 채용한 인버터 구동 펄스로 변환되어 유도전동기를 구동한다. 실험도중 발생하는 결과 데이터들은 실험종료후 프린터 포트를 통해 PC로 업로드되어 플로트상에서 분석될 수 있도록 하였다. 유도기 구동중에 부하를 인가하기 위하여 유도기와 직렬로 2Kw급 타여자 직류전동기를 연결하였고, 직류전동기 고정자 전류를 제어하여 필요한 부하토크가 발생할 수 있도록 직류전동기 드라이브와 제어기를 별도로 제작하였다.

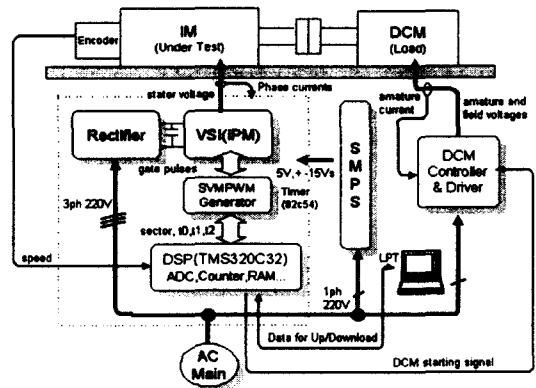


그림 2. 하드웨어 구성.  
Fig. 2 Hardware configuration.

## 3. 시뮬링크를 이용한 인버터 공간벡터 전압변조 시뮬레이션

### 3.1 인버터 공간벡터 전압변조[7]

3상 인버터의 최대 출력전압은 그림 3에 보인 바와 같이 벡터들의 조합으로 정육각형을 형성하고, PWM을 행하면 평균적인 의미에서 정육각형 내부의 임의의 위치에 놓일 수 있다. 따라서 기준전압 벡터  $V^*$ 에 해당하는 출력전압벡터는 삼각형의 두 변에 해당하는 전압벡터로부터 합성할 수 있고, 두 전압벡터의 인가시간의 합이 샘플링시간  $T$ 보다 적으면 나머지 시간만큼 '0'전압벡터를  $t_0$  동안 인가한다. 예를 들어, 기준전압  $V^*$ 가 섹터 I에 존재하면  $V^*$ 에 인접한  $V_1$ 과  $V_2$  벡터로부터  $V^*$ 를 합성할 수 있으므로 이를 수식으로 표현하면 다음과 같다.

$$V_1 t_1 + V_2 t_2 = V^* T \quad (1)$$

여기서  $t_1, t_2$  는 각각  $V_1, V_2$  를 인가할 시간이다.  $V_1, V_2$  및 영전압벡터( $V_0$  또는  $V_7$ )가 인가되는 시간  $t_1, t_2, t_0$  는 식(1)을 직각 좌표상에서 정리하면 식(2)와 같이 표현된다.

$$\begin{aligned}
 t_1 &= T \frac{|V^*|}{2/3V_{dc}} \sin\left(\frac{\pi}{3} - \theta\right) \\
 t_2 &= T \frac{|V^*|}{2/3V_{dc}} \sin \theta \\
 t_0 &= T - t_1 - t_2, \quad 0 < \theta < \pi/3
 \end{aligned}
 \tag{2}$$

한편,  $V_0$  와  $V_7$  에 상관없이 두 벡터중 하나를 선택하면 필요한 기준전압을 생성시킬 수 있으나 인버터의 커뮤테이션 손실과 리플의 크기등에 있어서는 두 영전압벡터를 조합하는 방식에 따라서 상이한 특성을 나타낸다. 일반적으로 영전압벡터를 조합하는 방식으로 DI(Direct Inverse)방식과 DD(Direct Direct)방식이 사용되는데, DI방식이 DD방식에 비하여 커뮤테이션 횟수를 줄일 수 있기 때문에 유도전동기 구동 인버터의 경우 주로 DI방식이 사용된다. 본 논문에서는 인버터 전압변조방식으로 DI방식을 사용하였다. 식(2)에서의 변조계수,  $1.5T |V^*|/V_{dc}$  연산을 위한 시뮬링크 블록을 보이면 그림 4와 같고, 기준전압벡터의 위상,  $\theta$ 와 최종 스위칭 시간값( $t_1, t_2, t_0$ )들은 3.2절에서 소개될 섹터와 스위칭시간연산 s-function 블록(sector, t1, t2, t0 calculation)에서 구한다.

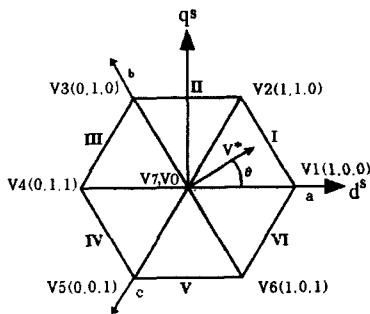
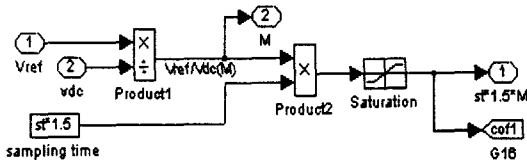


그림 3 PWM VSI의 공간 전압벡터.  
Fig. 3 Voltage space vector of PWM VSI.



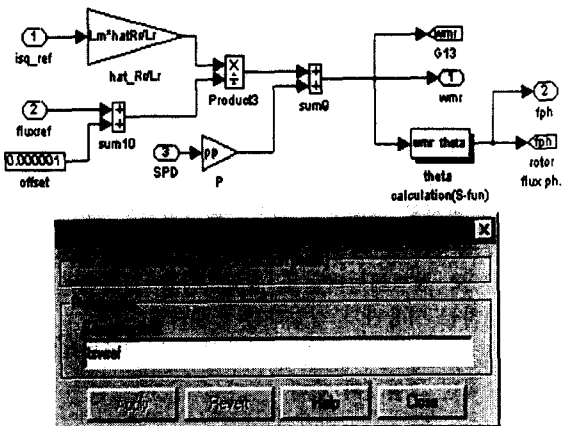
Vref: 기준전압 Vdc: dc-link 전압 st: sampling time(T)

그림 4 변조계수 연산 시뮬링크 블록  
Fig. 4 Simulink block for modulation coefficient calculation.

### 3.2 유도기 간접벡터제어

유도기 간접벡터제어방식[9],[10]은 유도기 토크성분 기준 전류, 자속 기준값 및 회전자 파라미터들을 이용하여 연산되는 슬립각속도에 유도기각속도를 더하여 회전자 자속의 각속도와 위상을 구하고 이를 순시 토크제어시 사용하는 방식으로, 전류제어가 충실히 수행되어 토크, 자속성분 기준전류들과 실제 전류가 일치함을 전제로 한다. 따라서 간접벡터제어성능이 제대로 나타나기 위해서는 견실한 전류제어가 선행되어야 하는데, 본 논문에서는 유도기 구동시 발생하는 역기전력성분을 토크 및 자속전류 제어기에 전향(feedforward)시켜 토크와 자속전류 발생시의 상호간섭항을 제거하는 비간섭 전류제어기법[6]을 사용하였다.

그림 5는 회전자자속의 각속도와 위상을 연산하기 위한 시뮬링크 블록도이고, 그림 6은 비간섭전류제어를 위한 시뮬링크 블록도를 보인다. 이 때, 회전자자속의 각속도로부터 위상을 구하는 적분기 부분은 S-function (theta calculation)으로 작성하였다.



wmr: 회전자자속 각속도 fph: 회전자자속 위상  
fluxref: 회전자자속 기준값 isq\_ref: 토크전류 기준값  
SPD: 유도전동기 속도

그림 5 회전자자속 각속도와 위상 연산 시뮬링크 블록.  
Fig. 5 Simulink block for rotor-flux angular speed and phase calculation.

### 3.2 시뮬링크 블록구성 및 시뮬레이션

그림 7은 실험을 하기 이전에 컴퓨터 시뮬레이션을 통해 시스템의 각 블록들이 올바르게 동작되는 지를 확인하기 위해 구성된 벡터제어시스템 시뮬링크 블록이다. 즉, 그림 4와 동일한 내부구조를 갖는 변조지수 연산블록(M calculation), 그림 5와 동일한 내부구조를 갖는 자속각 연산블록(wmr and flux phase cal), 그림 6과 동일한 내부구조를 갖는 자속-토크전류 비간섭제어블록(Decoupling controller)과 S-function으로 작성

된 섹터 및 스위칭시간 연산블록(sector, t1, t2, t0 calculation), PWM데이터/타임 연산블록(PWM data/time calculation) 등, 공간벡터 전압변조와 유도전동기 벡터제어시 요구되는 필수적인 연산을 행하는 블록들을 상호연계하여 시스템을 구성한 후, 유도가 1500rpm 구동을 위한 모의 기준전압(전기각속도 3000rpm, 50Hz) 신호를 인가하여 이에 따른 올바른 인버터 게이트 구동펄스가 발생하는지를 실험을 하기 전에 컴퓨터 시뮬레이션상에서 일차로 확인한다. 그림 8은 시뮬레이션 결과로, 기준전압벡터가 2차원 d-q 평면상에서 위치하는 섹터와 스위칭타임( $t_1, t_2, t_0$ )의 시간에 따른 변화모양을 플로팅한 모습이다.

시뮬레이션 결과 기준전압이 섹터 I에서 섹터 VI 까지 순차적으로 이동하고, 각 섹터에 위치하는 도중에 연산되는 스위칭타임도 식(2)와 일치됨을 보임으로써 그림 7의 시뮬레이션블록이 올바르게 구성되었음을 알 수 있다.

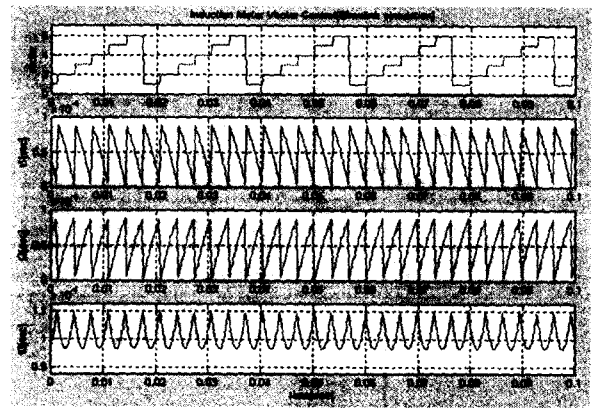
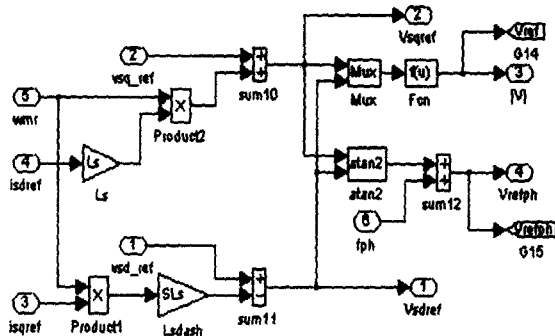


그림 8 그림 7에 대한 시뮬레이션결과.(기준전압섹터(sector), 스위칭타임(t1, t2, t0), 주파수 : 50Hz, T=200µs)  
 Fig. 8 Simulation result for the Fig. 7.(reference voltage sector (sector), switching time(t1, t2, t0), frequency : 50Hz, T=200µs)



isdref, isqref:자속, 토크전류 기준값  
 vsd\_ref, vsq\_ref:자속, 토크전류 PI제어기 출력  
 Vsdfref, Vsqref:비간섭전류제어기 출력  
 |V|:기준전압 크기 Vrefph:기준전압 위상

그림 6 비간섭전류제어 시뮬링크 블록.  
 Fig. 6 Simulink block for decoupling current control.

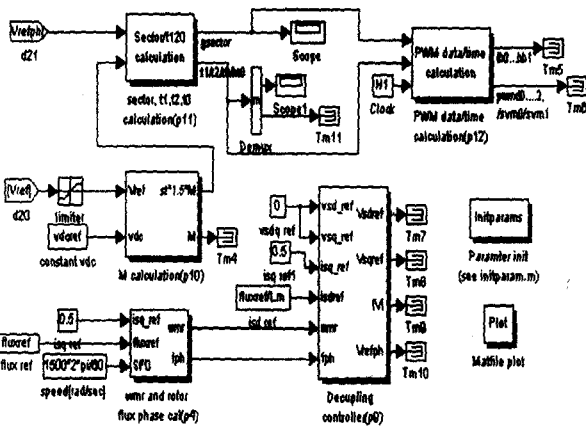


그림 7 공간벡터 PWM 시뮬레이션 블록.  
 Fig. 7 Simulation block for space vector PWM.

#### 4. 실험

##### 4.1 인버터 공간벡터 전압변조

그림 9는 시뮬레이션단계에서 사용한 그림 7과 동일한 블록으로부터 C코드를 발생시켜 실험을 하기 위한 공간벡터 PWM 실험 블록이다. 이때, 실험결과 데이터의 업로드를 위해 S-function으로 작성된 Data upload LPT1 블록과 실험진행과정을 보이기 위한 LED 인터페이스블록(7Seg. led) 등을 추가시켰다. Parameter init 블록에 의해 시스템 파라미터들이 초기화되면, RTW에 의해 그림 9의 블록들로부터 C코드가 발생되고, 컴파일, 링크과정을 거쳐 hex파일이 생성되면 DSP보드로 다운로드되어 실험이 수행된다. 실험이 종료되면 실험중에 발생하는 결과데이터들이 PC상으로 업로드되어 플로팅되는데, 이 경우 그림 10과 같이 분석이 필요한 시뮬링크블록의 각 포인트들을 시뮬링크 From 블록[1]을 이용하여 윈도우상에서 Data upload LPT1 블록에 연결시켜 주기만하면 업로드될 데이터가 선택되도록 하였고, 사용자 인터페이스 다이얼로그박스를 이용하여 업로드되는 데이터의 개수, 업로드 샘플링주기, 이름 및 저장되는 데이터 파일(mat 파일)의 이름 등을 사용자가 설정할 수 있도록 하였다.

그림 11은 실험 결과로, 시뮬레이션의 경우와 마찬가지로 유도가 1500rpm 구동을 위한 기준전압(전기각속도 3000rpm, 50Hz) 신호를 인가한 경우 기준전압벡터의 섹터와 스위칭타임의 시간에 따른 변화모양을 플로팅한 모습이다. 이로부터 그림 8의 시뮬레이션 결과와 동일하게 기준전압이 각 섹터를 순차적으로 이동하며 발생되고, 각 섹터에 위치하는 도중에 연산되는 스위칭타임이 식(2)와 일치하여 발생됨을 알 수 있다. 결과적으로, 컴퓨터 시뮬레이션시의 시뮬링크블록이 실험 구현시 별도의 수정없이 사용되어 시뮬레이션과 동일한 결과를 얻었기 때문에 시스템에 대한 신뢰성과 유용성을 확인 할 수 있었고, 시스템설계, 시뮬레이션 및 실험 검증에 이르는 시스템개발과정에서 소요되는 시간을 기존의 시스템 개발방식에 비하여 현저히 감소시킬 수 있었다.



그림 14는 유도기 기준속도를 무부하 상태에서 1500rpm에서 500rpm으로 감속시킨 경우 응답특성들을 보인다. 이때 유도기 구동시점인 0초에서 1.5초까지는 유도기 기준속도는 0으로, 자속성분전류의 기준값은 3[A]로 제어하여 이 기간동안 회전자 자속이 정상상태에 도달하도록 하였다. 그림 14에서 (a)는 자속성분전류(isd)응답특성을 보이고, (b)는 토크성분전류의 기준값(isqref)이고, (c)는 토크성분전류응답이다. (d)는 속도응답특성이다. 실험결과 데이터들은 실험 종료후 DSP보드로부터 PC로 Matlab(V5.3)의 mat파일 형식으로 업로드되도록 하여 Matlab의 다양한 플로팅 기능을 사용할 수 있도록 하였다. 그림 15는 감속시 제어특성을 자세히 관찰하기 위해서 Matlab의 zoom 기능을 이용하여 감속기간의 응답을 확대하여 도시한 것이다. 실험결과, 속도 및 전류들이 유도기 전동작영역에 걸쳐 기준값을 잘 추종하고 있음을 알 수 있다.

표 1 유도전동기 파라미터.

Table 1 Parameters of Induction Motor.

3 $\phi$ , 4P, 220V, 60Hz, 3.6A	
Rs	3.63[ $\Omega$ ]
Rr	3.87[ $\Omega$ ]
Ls	0.104[H]
Lr	0.103[H]
Lm	0.1[H]
J	0.012[Kg m <sup>2</sup> ]

표 2 제어기 이득과 파라미터.

Table 2 Controller gains and parameters.

제어주기	속도제어	1ms
	전류제어	200 $\mu$ s
속도제어이득	Kp : 0.3	Ki : 62.4
전류제어이득	토크성분전류제어	Kp:30 Ki:3120
	자속성분전류제어	Kp:30 Ki:3120

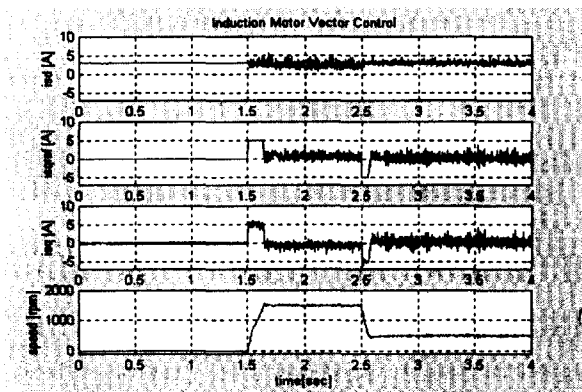


그림 14 유도전동기 벡터제어응답. ((a)자속전류(기준값:3[A]) (b)토크전류기준값 (c)토크전류 (d)속도).

Fig. 14 Induction motor vector control response. ((a)flux current(reference : 3[A]) (b)torque current reference (c)torque current (d)speed)

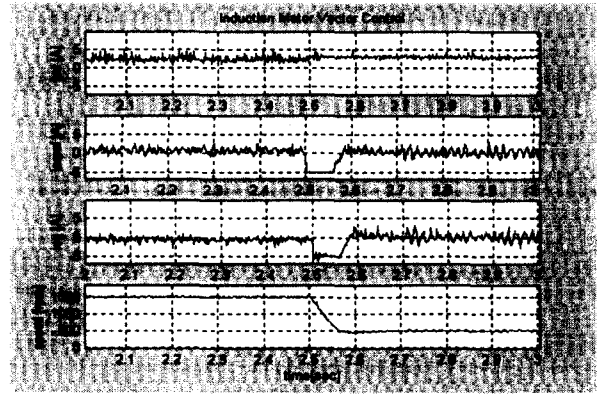


그림 15 유도전동기 벡터제어응답.(부분확대) ((a)자속전류(기준값 : 3[A]) (b)토크전류기준값 (c)토크전류 (d)속도).

Fig. 15 Induction motor vector control response.(zoomed) ((a)flux current (b)torque current reference (c)torque current (d)speed)

### 5. 결론

본 연구에서는 시뮬링크 윈도우상에서 블록다이어그램 형태로 제어시스템이 설계되면 설계된 제어시스템으로부터 실험을 위한 C코드의 생성, 컴파일 및 링크가 자동으로 수행되는 실시간 제어시스템 설계 환경을 구축하고, DSP와 IPM을 이용한 유도전동기 가감속 드라이브 개발에 적용하였다. 본 방식에 의하면, 실험을 위해 DSP로 다운로드 될 C코드가 컴퓨터 시뮬레이션에서 사용된 시뮬링크 블록과 동일한 블록으로부터 자동으로 생성되기 때문에 실험을 위한 별도의 프로그래밍과정이 필요없고, 제어시스템을 변동시키는 경우에도 윈도우상에서 간단히 블록의 파라미터나 블록상호간의 연결상태만을 바꾸면 되기 때문에 시스템개발과 변동시에 소요되는 시간을 최소화시킬 수 있다. 개발된 시스템에 대하여 공간벡터 PWM 변조를 위한 컴퓨터 시뮬레이션과 실험을 행한 결과, 시뮬레이션과 실험이 동일한 결과를 보였고, 유도기 가변속제어 실험을 행한 결과, 속도와 전류들이 유도기 전동작영역에 걸쳐 기준값을 잘 추종함이 확인되어 시스템에 대한 신뢰성을 검증할 수 있었다.

### 참고 문헌

- [1] Simulink, Dynamic System Simulation for MATLAB, Using Simulink, the Math Works Inc., 1998.
- [2] Simulink, Dynamic System Simulation for MATLAB, Writing S-Functions, the Math Works Inc., 1998.
- [3] Real-Time Workshop, for Use with Simulink, User's Guide, the Math Works Inc., 1998.
- [4] TMS320C3X/4X Optimizing C Compiler User's Guide, the Texas Instruments Inc., 1998.
- [5] "Introduction to Intellimode™ Intelligent Power Modules", Application note, Powerex, Inc.
- [6] P. T. Krein et. al., "Comparative Analysis of Scalar

- and Vector Control Method for Induction Motors", *IEEE PESC*, pp.1139-1145, 1993.
- [7] 김남정, 유지윤, 강문호, "공간벡터 변조기법을 이용한 유도전동기 완전 디지털 벡터제어", *대한전기학회논문지*, 44권, 11호, pp.1447-1456, 1995.
- [8] P.Vas, *Vector Control of AC Machines*, Clarendon Press, Oxford, 1992.
- [9] W. Leonhard, *Control of Electrical Drives*, Springer-Verlag, 1991.
- [10] B. K. Bose, "Technology Trends in Microcomputer Control of Electrical Machines.", *IEEE Trans. on Industrial Electronics*, vol.35, no.1, pp.160-177, 1988.

---

저 자 소 개



강 문 호 (康 文 浩)

1964년 7월 13일 생. 1988년 고려대 전기공학과 졸업. 1995년 동 대학원 전기공학과 졸업(공학). 1997년~현재 선문대 기계및제어공학부 조교수

Tel : 041-530-2339, Fax : 041-530-2981

E-mail : mhkang@omega.sunmoon.ac.kr