

# MATLAB Simulink를 이용한 유도전동기 구동 시뮬레이터 개발

오원현, 강호진, 김종선, 신은철, 유지윤, 김성환\*, 박태식\*\*

고려대학교 전기공학과 산업전자 연구실  
\*목포대학교 제어시스템공학과, \*\*(주)삼성전자

## A Development on Induction Motor Drive Simulator using MATLAB Simulink

Won-hyun O, Ho-Jin Kang, Jong-Sun Kim, Eun-Chul Shin, Ji-Yoon Yoo  
Sung-Hwan Kim\*, Tae-Sik Park\*\*

Industrial electronic laboratory, Electrical engineering, Korea university.

\*Control and instrumentation engineering, Mokpo National University

\*\*Samsung Electronics Co., Ltd.

### ABSTRACT

In this research a Matlab/Simulink/RTW - based on the 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 a DSP c compiler complies the program codes.

하고자 하는 Rapid Prototyping Process는 개발된 알고리즘과 검증을 위한 시뮬레이션 그리고 실행을 위한 소프트웨어가 일련의 과정을 통해 자동으로 생성되기 때문에 일부분의 알고리즘의 수정이나 시뮬레이션의 수정이 있더라도 자동 코드 생성기에 의해 프로그램 코드가 생성되므로 그 만큼의 시간과 노력이 줄어들게 된다.

### 1. 서론

본 연구에서는 일반적인 제어시스템 개발에 소요되는 비용과 시간을 줄이기 위해서 제어시스템 설계, 시뮬레이션 그리고 실험을 위한 실행파일 생성과 구현까지의 전과정을 최소한의 프로그램 코딩과 반복작업의 수행으로 자동 실행되는 실시간 시뮬레이터를 개발하고자 한다.

그림 1은 종래의 제어시스템의 구성과 본 논문에서 제시하고자 하는 제어시스템의 구성을 간략하게 비교한 그림이다. 종래의 제어시스템은 제어 알고리즘의 개발과 개발한 알고리즘의 검증을 위한 시뮬레이션 그리고 검증된 알고리즘의 소프트웨어 디자인의 3단계 과정이 독립적으로 분리되어 있어서 알고리즘 단계에서 수정을 가할 때, 나머지 단계에서도 일일이 검증과 수정을 해야하는 매우 번거롭고 비효율적인 구조였다. 그러나 본 논문에서 제시

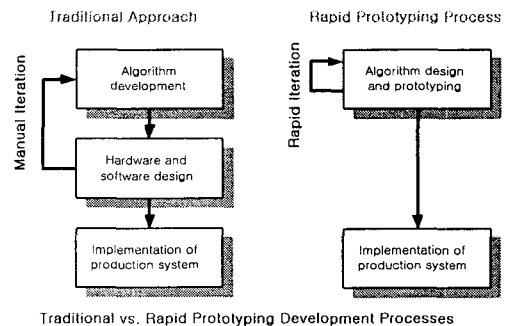


그림 1 기존의 시스템과 Rapid prototyping 시스템  
Fig. 1 Traditional system and Rapid prototyping system

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

## 2. Simulink/RTW 기반의 시뮬레이터 내부구성

### 2.1 시뮬레이터 전체 계략도<sup>[2]</sup>

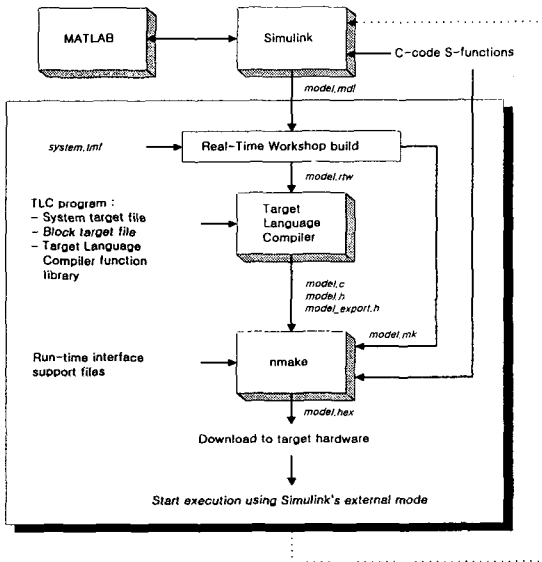


그림 2 시뮬레이터 전체계략도  
Fig. 2 Simulator Total Block Diagram

시뮬레이터는 전체적으로 크게 시뮬링크 상에서 설계된 블록을 코드변환하는 과정과 변환된 코드를 해당 컴파일러로 컴파일하기 위한 mk 파일을 만드는 두 과정으로 나눌 수 있다.

### 2.2 RTW를 이용한 C-code 변환<sup>[2]</sup>

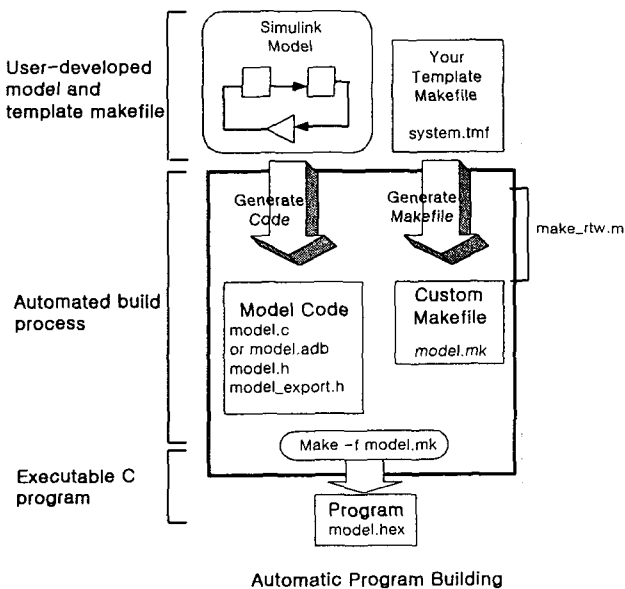


그림 3 RTW를 이용한 code 변환과 mk 파일 생성  
Fig. 3 Code transformation and mk file building using RTW

RTW(Real-Time Workshop)은 make\_rt.w.m 파일을 가지고 model.rtw (model.mdl 파일과 변환된 C파일 사이의 중간단계에서 변환되는 파일로서 mdl 파일의 각 블록들의 모든 정보를 가지고 있는 파일)을 생성하고, RTW은 make\_rt.w.m을 가지고 다시 Target system(ex. TMS320C32)에 대응하는 특정한 TLC(Target Language Compiler ex. grt.tlc)를 호출하고 TLC는 model.rtw로부터 블록 정보들을 수집하여 model.c, model.h, model.prm 등의 ANSI C 코드들을 생성한다.

### 2.3 RTW를 이용한 mk 파일 생성<sup>[2]</sup>

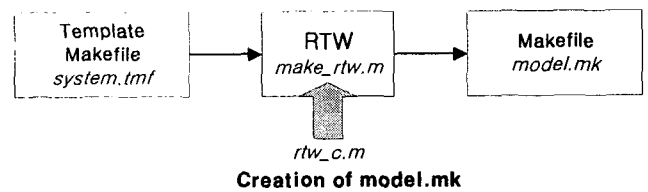


그림 4 mk 파일 생성 과정  
Fig. 4 mk file building process

RTW에서 하는 두 가지 역할 중 하나가 mdl 파일로 생성된 모델을 C-code로 변환시켜주는 일이다. 나머지 하나는 생성된 C 소스들과 사용자가 만든 C 소스 그리고 S-function으로 만든 C 소스들을 컴파일(Compile), 링크(Link) 할 때 그 순서와 연관관계의 정보를 가지고 컴파일과 링크를 순서적으로 행하는 파일인 mk 파일을 만드는 일이다. RTW의 make\_rt.w.m 파일은 rtw\_c.m파일을 호출하여 사용자가 작성한 Template make file로부터 make파일인 model.mk 파일을 생성한다. 그 과정을 간단히 다음의 그림에서 보여주고 있다.

### 2.4 nmake를 이용한 컴파일 및 실행파일생성<sup>[6]</sup>

RTW의 make\_rt.w.m 파일은 Visual C++이 제공하는 nmake.exe을 호출하여 model.mk에 정의되어 있는 순서대로 make 작업을 수행한다.

nmake.exe는 RTW에서 생성된 코드(generation code)인 model.c, model.h, model.prm, model.reg들, RTW에서 제공하는 라이브러리 C코드들(ex. rt\_hypot.c, rt\_look.c, rt\_matrix.c, rt\_sgn.c, rt\_zfcfn.c, rt\_rand.c), 사용자가 작성한 S-function C코드 및 기타 C코드들을 DSP컴파일러가 제공하는 응용프로그램들을 호출해 가면서 컴파일/어셈블/링크시킨 후 실행파일(model.out)을 생성한다.

최종적으로 nmake.exe는 실행파일(model.out)로부터 DSP보드에 다운로드 가능한 파일포맷을 갖는 헥사파일(model.hex)을 생성한다.

다음 표는 사용되는 C-code를 종류별로 나열해 놓은 것이다.

표 1 항목별 소스파일

Table 1 Itemization source code

mdl 모델에서 변환된 C-code	RTW 라이브러리 C-code	사용자가 작성한 S-function C-code	기타 C-code
model.c model.h model.prm model.reg	rt_hypot.c rt_look.c rt_matrix.c rt_sgn.c rt_zfcn.c rt_rand.c	SfunUpload.c sfunc_sector.c sfuncled.c sfunc_pwmdata.c pgcounter.c dec2hex.c DAC.c tm8254.c thetacal.c	Imimain.c Ioptim.c Realview.c receive32.c RINTrev.c

### 2.5 실행 파일구현

실행파일의 구현은 최종 hex파일(model.hex)을 target system에 다운로드, 실행, 데이터 업로딩의 과정으로 구성된다.

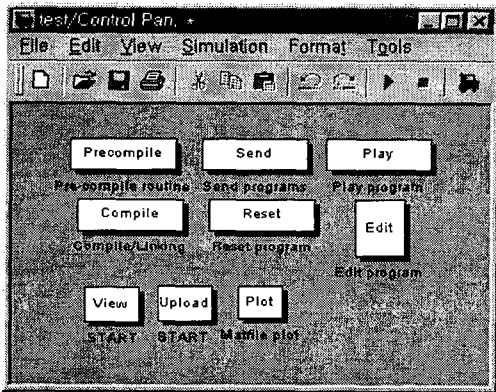


그림 5 사용자 컨트롤 패널  
Fig. 5 User control panel

#### • 다운로드

VC++로 작성된 응용파일들 중에서 다운로드 파일인 send.exe를 시뮬링크 윈도우 상에서 호출하고, 다운로드 파일이 hex파일(model.hex)을 PC로부터 DSP로 직렬전송한다.

#### • 실행

VC++로 작성된 응용파일들 중에서 프로그램수행파일인 play.exe을 시뮬링크 윈도우상에서 호출하고, 프로그램 수행파일은 DSP에 직렬 전송된 hex파일을 실행한다.

#### • 업로딩

필요한 경우 VC++로 작성된 응용파일들 중 데이터 전송 파일인 receive.exe을 시뮬링크 윈도우 상에서 호출하고, 데이터 전송파일은 DSP로부터 PC로 실험데이터를 직렬전송시켜 실험데이터를 분석 가능하게 하거나, 데이터 파일의 생성, 저장, 가공이 가능하게 하였다.

## 3. Simulink를 이용한 유도전동기 벡터제어 시뮬레이션

### 3.1 시뮬레이션 전체 계략도<sup>[1]</sup>

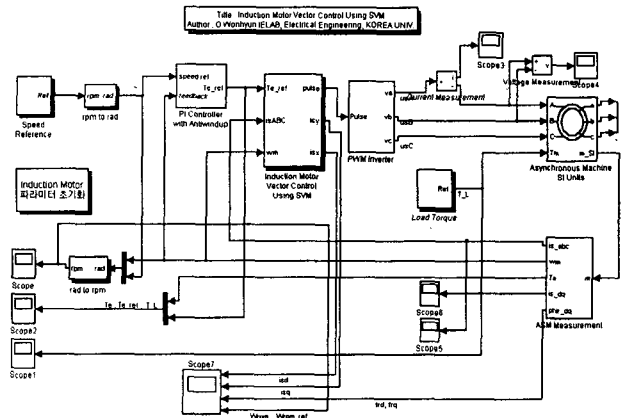


그림 6 시뮬레이션 전체 블록도  
Fig. 6 Simulation Total Block Diagram

### 3.2 시뮬레이션 결과

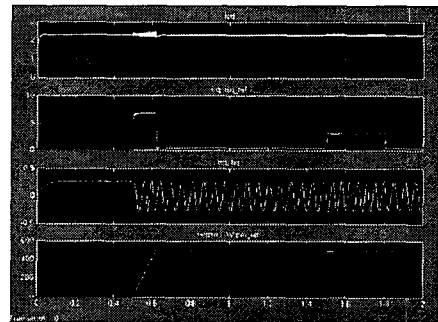


그림 7 스텝부하 인가시 500rpm에서의 속도 및 전류제어 특성  
Fig. 7 Speed and current control characteristics with step load at 500rpm

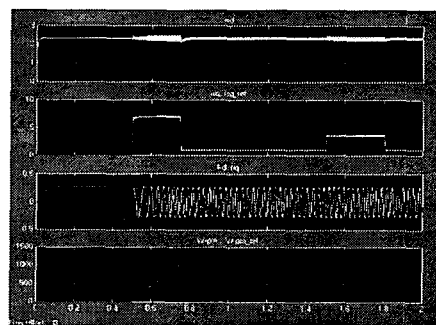


그림 8 스텝부하 인가시 1000rpm에서의 속도 및 전류제어 특성  
Fig. 8 Speed and current control characteristics with step load at 1000rpm

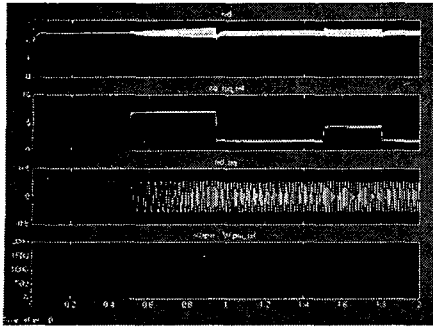


그림 9 스텝부하 인가시 1690rpm에서의 속도 및 전류제어 특성  
 Fig. 9 Speed and current control characteristics with step load at 1690rpm

## 4. 실험 결과

### 4.1 실험 장치

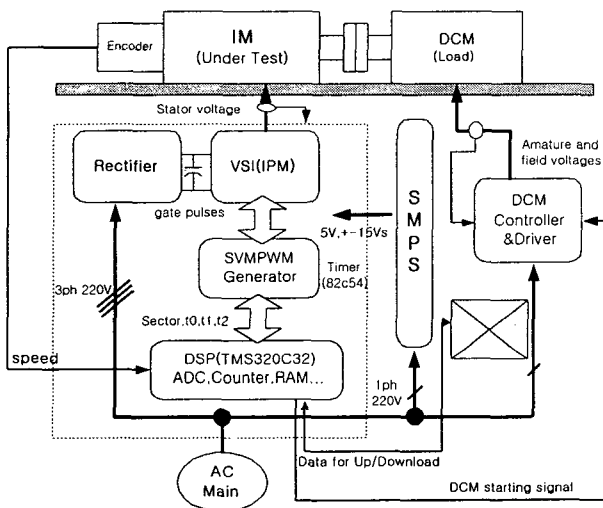


그림 10 하드웨어 실험장치 전체 구성도  
 Fig. 10 Total Hardware System Configuration

본 연구에서 구성된 하드웨어 시스템은 부하변동에 대한 견실한 속도, 위치 및 전류제어 특성을 얻을 수 있는 유도전동기의 드라이브로 전체 시스템을 완전 디지털로 구성하여 소프트웨어를 실행할 수 있도록 하였다.

본 논문에서 사용되는 전체 제어 알고리즘은 많은 양의 복잡한 계산을 실시간 처리해야 하기 때문에 필수적으로 고속의 프로세서를 요구한다. 따라서 본 연구에서는 고속 및 다량의 연산에 있어서 적합한 프로세서인 DSP TMS320C32를 하드웨어 시스템의 주연산 및 제어기로 사용하였고, 이와 함께 전력 변환부의 고직접화를 통한 인버터의 신뢰성 향상을 실현하기 위해 IPM(Intelligent Power

Module)을 이용하여 전력 변환부를 구성하였다. 특히, 본 연구에서 사용된 IPM은 모듈 내부에 자체적인 고장 진단회로가 포함되어 있고, 전력회생에 필요한 별도의 IGBT를 갖추고 있기 때문에 인버터 회로의 집적화 및 신뢰성을 극대화 할 수 있다.

표 2 유도전동기 내부 파라미터  
 Table 2 Induction Motor Parameter

3상 4극 농형 유도전동기	750[W] 1690[rpm] 60[Hz]
$R_s$	3.633 [ $\Omega$ ]
$R_r$	3.87 [ $\Omega$ ]
$L_s$	0.104 [H]
$L_r$	0.103 [H]
$L_m$	0.1 [H]
$J$	0.012 [ $Kg \cdot m^2$ ]

표 3 제어기 이득값  
 Table 3 Controller Gain

		무부하	부하
전류 제어기	P 이득값	3.47	2.61
	I 이득값	1829.46	914.73
속도 제어기	P 이득값	0.31	0.61
	I 이득값	15.1	1.51

### 4.2 무부하시 실험결과

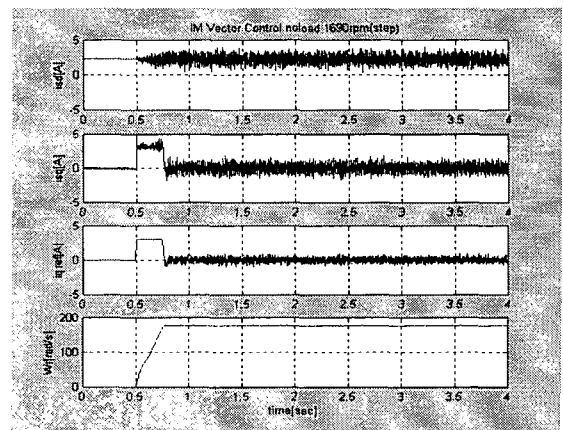


그림 11 무부하시 1690rpm에서의 속도 및 전류제어 특성  
 Fig. 11 Speed and current control characteristics with no load at 1690rpm

### 4.3 정격부하 인가시 실험결과

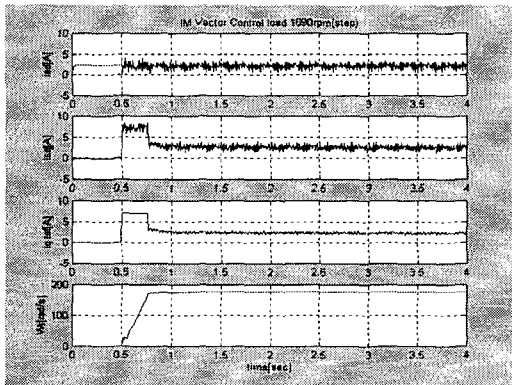


그림 12 부하 인가시 1690rpm에서의 속도 및 전류제어 특성  
 Fig. 12 S speed and current control characteristics with load at 16900rpm

### 4.4 가변속시 실험결과

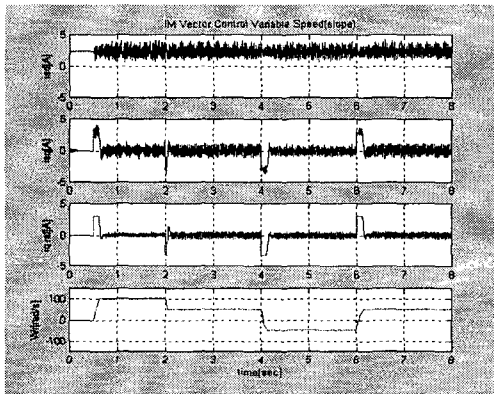


그림 13 가변속시(step지령) 속도 및 전류제어 특성  
 Fig. 13 Speed and current control characteristics with variable speed (step ref. load input)

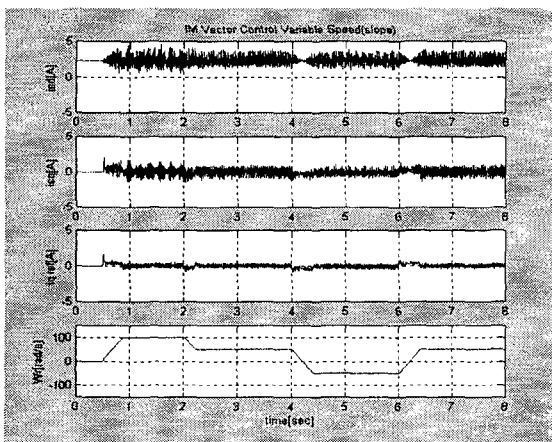


그림 14 가변속시(slope지령) 속도 및 전류제어 특성  
 Fig. 14 Speed and current control characteristics with variable speed (slope ref. load input)

## 5. 결 론

본 연구는 Matlab/Simulink/RTW 기반을 이용한 실시간 시뮬레이터 환경을 구축하여 사용자가 Simulink상에서 설계한 제어기 모델을 보다 빠르고 정확하게 타겟 시스템에서 적용해 볼 수 있게 하였다.

## 참 고 문 헌

- [1] Simulink, Dynamic System Simulation for MATLAB, Using Simulink, the Math Works Inc., 2000
- [2] Real-Time Workshop, for Use with Simulink, User's Guide, the Math Works Inc., 2001
- [3] 김남정, 유지윤, "공간 벡터 변조기법을 이용한 유도전동기의 완전 디지털 벡터제어", 대한 전기학회 논문지. 44권 11호, 1995. 11
- [4] W. Novotny and R.D.Lorenz, "Principle of Vector Control and Field Orientation", in Tutorial Course Record, "Introduction to Field Orientation and High Performance AC Drives", as part of 1985 Annu. Meet IEEE-IAS, pp. 6~1 to 6~17
- [5] D. Wuest and F. Jenni, "Space Vector-based Current Control Schemes for Voltage Source Inverters", IEEE PESC, pp 986-992, 1993
- [6] TMS320C3X/4X Optimizing C Compiler User's Guide, the Texas Instruments Inc., 1998