工學碩士 學位論文

MRAC를 적용한 속도추정기를 가지는 유도전동기의 센서리스 벡터제어에 관한 연구

A Study on the Sensorless Vector Control of Induction Motor with Speed Estimator Using MRAC

指導教授 李 成 根

2001年 2月

韓國海洋大學校 大學院

電氣工學科 電氣工學專攻

崔 承 鉉

本 論文을

崔承鉉의 工學碩士 學位論文으로 認准함

委員長 : 工學博士 全 泰 寅 ㊞

委 員 : 工學博士 金 基 萬 ㊞

委 員 : 工學博士 李 成 根 ㊞

2 0 0 1 年 2 月

韓 國 海 洋 大 學 校 大 學 院

電 氣 工 學 科

崔 承 鉉

목 차

A Study on the Sensorless Vector Control of Induction Motor with Speed Estimator Using MRAC

by Choi Seung-Hyun

Department of Electric Engineering
The Graduate School of Korea Maritime University
Pusan, Republic of Korea

Abstract

The vector control of an induction motor has been applied in various industrial applications instead of a DC motor.

The rotor speed value is used this vector control and it is gained by speed sensor. However, it decreases the system reliability and increases systems cost. Therefore, various control algorithms have been proposed for the estimation of speed.

PI control is usually applied to speed control because it maintains the best state if PI gains are properly selected. However it doesn't maintain the best state if the parameter of an induction motor is changed.

This thesis proposes a new algorithm of sensorless speed control of induction motor. The proposed algorithm is based on a torque equation using the theory of MRAC(Model Reference Adaptive Control). It is robust for parameter variation and disturbance. The estimated speed is used as feedback in a vector control system.

To show the validity of the proposed algorithm, computer simulation and experimental test have been implemented. The result has been proven an excellent characteristic of the drive system.

기 호 설 명

$i_{1a}(t), i_{1b}(t), i_{1c}(t)$: Stator instantaneous phase current $[A]$

$v_{1a}(t), v_{1b}(t), v_{1c}(t)$: Stator instantaneous phase voltage $[V]$

$\boldsymbol{i}_{1a}, \boldsymbol{i}_{1b}, \boldsymbol{i}_{1c}$: Stator three phase current vector $[A]$

$\boldsymbol{v}_{ab}, \boldsymbol{v}_{bc}, \boldsymbol{v}_{ca}$: Line to line voltage vector $[V]$

T_1, T_2, T_0 : Adjacent basic vectors & zero vector switching time $[sec]$

$i_{1\alpha}, i_{1\beta}$: Stator current in $\alpha\beta$ stationary reference frame $[A]$

$i_{2\alpha}, i_{2\beta}$: Rotor current in $\alpha\beta$ stationary reference frame $[A]$

i_{1d}, i_{1q} : Stator current in dq synchronously reference frame $[A]$

i_{2d}, i_{2q} : Rotor current in dq synchronously reference frame $[A]$

$v_{1\alpha}, v_{1\beta}$: Stator voltage in $\alpha\beta$ stationary reference frame $[V]$

$v_{2\alpha}, v_{2\beta}$: Rotor voltage in $\alpha\beta$ stationary reference frame $[V]$

v_{1d}, v_{1q} : Stator voltage in dq synchronously reference frame $[V]$

v_{2d}, v_{2q} : Rotor voltage in dq synchronously reference frame $[V]$

$\phi_{1\alpha}, \phi_{1\beta}$: Stator flux in $\alpha\beta$ stationary reference frame $[V]$

$\phi_{2\alpha}, \phi_{2\beta}$: Rotor flux in $\alpha\beta$ stationary reference frame $[V]$

ϕ_{1d}, ϕ_{1q} : Stator flux in dq synchronously reference frame $[Wb]$

ϕ_{2d}, ϕ_{2q} : Rotor flux in dq synchronously reference frame $[Wb]$

$V_1 \quad V_6$: Effective space voltage vector $[V]$

V_0, V_7 : Zero voltage vector $[V]$

$i_{1\alpha}^*, i_{1\beta}^*$: Stator current command in dq synchronously reference frame

$v_{1\alpha}^*$, $v_{1\beta}^*$: Stator voltage command in $\alpha\beta$ stationary reference frame

v_{1d}^*, v_{1q}^* : Stator voltage commands in dq synchronously reference frame

B_1 : Flux density $[Wb/m^2]$

I_m : Maxim value of phase current $[A]$

i_T : Torque command current $[A]$

i_M : Magnetic command current $[A]$

$\boldsymbol{i}_1$: Composed stator current vector $[A]$

$\boldsymbol{i}_2$: Composed stator current vector $[A]$

$\boldsymbol{i}_{2r}$: Rotor current vector in rotor reference frame $[A]$

$\boldsymbol{v}_1$: Composed stator voltage vector $[V]$

$\boldsymbol{v}_2$: Composed rotor voltage vector $[V]$

$\boldsymbol{v}_{2r}$: Rotor voltage vector in rotor reference frame $[V]$

$\boldsymbol{v}^*$: Reference voltage vector $[V]$

$\boldsymbol{\phi}$: Flux vector $[Wb]$

$\boldsymbol{\phi}_1$: Composed stator flux vector $[Wb]$

$\boldsymbol{\phi}_2$: Composed rotor flux vector $[Wb]$

$\boldsymbol{\phi}_{2r}$: Rotor flux vector in rotor reference frame $[Wb]$

w_e : Synchronous angular speed $[rad/\sec]$

w_s : Slip angular speed $[rad/\sec]$

$\widehat{w_r}$: Rotor angular estimated speed $[rad/\sec]$

w_r : Rotor angular speed $[rad/\sec]$

w_r^* : Rotor angular speed command $[rad/\sec]$

θ_e : Flux angle $[rad]$

θ_r : Rotor axis angle $[rad]$

θ_s : Slip angle $[rad]$

L_1 : Stator inductance $[H]$

L_2 : Rotor inductance $[H]$

$l_{1,}\ l_2$: Leakage inductance $[H]$

M_1 : Mutual inductance $[H]$

R_1 : Stator resistance $[\Omega]$

R_2 : Rotor resistance $[\Omega]$

R_{2r} : Rotor resistance in rotor reference frame $[R]$

σ : Leakage factor

p : Differential factor

J_M : Moment of inertia $[\text{kg} \cdot m]$

B : Braking factor

w_m : Mechanical rotor angular speed $[rad/\sec]$

T_l : Load torque $[N \cdot m]$

x_p : process state vector

y_p : process output

x_m : reference model state vector

e : process-model state error

e_1 : process-model output error

ε : error signal as used in the adaptation

v : error-augmenting signal

u : process input

r : reference input

ω : signal vector

θ : controller parameter vector

θ^* : correct parameter vector

ϕ : parameter error vector(θ - θ^*)

Γ : adaptation gain matrix

ASG : auxiliary signal generator

SPR : strictly positive real

1. 서 론

산업용으로 사용되고 있는 서보시스템은 DC서보와 AC서보 2가지로 나눌 수 있으며 직류전동기가 대부분 사용되어 왔다. 그러나 직류전동기는 정류자 및 브러시를 가지는 구조적 특징에 의한 본질적인 단점을 가지고 있어 직류전동기를 대신해 AC서보시스템을 사용하려는 연구가 활발히 진행되었다[1-2].

AC서보시스템은 전원의 주파수 및 전압의 변화를 필요로 하는 제어상의 어려움 때문에 적어도 1960년대 이전까지는 정격속도 부근에서의 정격운전용으로 사용되었다. 1970대 초에 벡터제어(Vector Control)가 발표되었고 1970년대 후반 이후에 마이크로 프로세서 및 전력용 반도체가 급속히 발전됨으로써 유도 전동기의 제어기술은 비약적으로 향상되어 압연기 등과 같은 수 kW에서 수십 kW이상의 중·대용량 가변속 제어용에도 많이 사용하게 되었다[3-4].

벡터제어는 유도전동기의 1차, 2차 측의 전류, 자속 등의 전동기내의 신호를 벡터 신호로 취해 1차전류인 전기자 전류를 자속발생 성분과 토크발생성분으로 분리하고 이를 독립적으로 제어하여 타여자 직류 전동기의 수준으로 순시토크를 발생시키는 제어방식이다. 그러나 이와 같은 벡터제어를 수행하기 위해서는 고성능의 전류제어가 필요하며, 이에 따라 실시간의 빠른 부동소수점 연산이 필수적으로 요구되는데 이는 DSP와 같은 고속의 마이크로프로세서의 등장과 대용량이면서도 스위칭주파수가 수십 kHz에 이르는 파워 MOSFET나 IGBT등이 상용화되어 제작이 가능하게 되었다.

벡터제어는 자속각을 얻어내는 방법에 따라 직접벡터제어와 간접벡터

제어로 나눌 수 있다. 직접벡터제어는 일명 자속피드백제어라고하며 전동기내의 공극자속을 홀소자나 검색코일등으로 직접 측정하거나 고정자의 전류, 전압 등을 측정하여 자속을 추정하는 방법이다. 따라서 직접자속을 측정할 경우는 전동기의 회전속도를 정확하게 검출할 수 있는 속도센서가 필요하게 된다. 간접벡터제어는 고정자전류와 전동기 파라미터를 이용하여 슬립각속도를 계산하고, 여기에 전동기 회전속도를 더하여 간접적으로 자속각속도를 구하는 방식으로, 피드포워드 방식이라 하며 정확도가 높고 구현이 용이하며, 저속에서 고속영역까지 사용이 가능할 뿐만 아니라, 무엇보다도 일반적인 전동기를 그대로 사용할 수 있다는 장점이 있다. 이와 같은 벡터제어에서는 회전자의 속도를 필요로 하며, 이를 검출하기 위해 tacho-generator, digital shaft-position encoder와 같은 속도센서를 사용한다. 그러나 속도센서를 부착하는 것은 여러 가지 면에서 단점을 가진다. 우선 전동기의 온도를 고려해 센서를 선택하여야 하며 센서를 기계적으로 취부하여야 하는 어려움과 취부에 따른 유도전동기가 가지는 본래의 강인함을 약화시키고, 추가적인 센서의 부착으로 가격이 상승한다. 하지만 무엇보다도 속도센서에 제어의 신뢰도가 달려있기 때문에 전체적인 시스템의 신뢰도는 센서가 없는 경우보다 크게 떨어지게 된다. 따라서 속도센서를 제어 시스템에서 제거한 센서리스 벡터제어 이론이 연구되어, 지난 십여년간 여러 가지의 속도추정 방법이 제시되었다. 이러한 센서리스 벡터제어는 제어 장비의 가격을 저하시킬뿐 아니라 많은 산업분야에서 제어 성능을 향상시킨다[4-9].

유도전동기의 제어시스템 구성에 속도제어부로서 PI제어기를 사용하는 방식이 널리 채택되고 있다. 그 이유는 PI제어기의 제어 이득 값이 적절하게 선정될 경우 최적의 운전상태를 유지할 수 있는 우수한 특성을 갖고 있으면서도 제어 알고리즘이 간단하여 손쉽게 구현할 수 있기 때문이

다[10-11].

PI제어를 기본으로 하는 속도제어계는 그 제어 파라미터 조절에 의해서 속도와 토크를 제어하나 시스템의 기계정수인 관성, 마찰계수 등의 파라미터가 운전 중에 변하는 경우에는 최적의 운전 상태를 유지하기 위해서 제어 상수 값들을 적절하게 변화시켜야 한다. 또한 PI제어기는 부하외란이나 파라미터 변동에 대해 매우 민감하기 때문에 속도 및 토크응답의 극점을 PI제어의 파라미터 범위 안에서 설계하는 것은 한계가 있으며, 따라서 운전범위에 제한을 받게 된다[12-13].

본 논문에서는 센서리스 간접벡터제어를 위해 MRAC를 이용하여 속도추정기를 설계한다. 이는 유도전동기의 토크발생식에 기초해 속도추정식을 유도하여 파라메트릭(시변인) 모델식을 세우고, 이에 선형의 프로세스식이 모델식과 perfect 모델-matching될 수 있게, primary controller를 구성하는 것이다. 제안된 상태정보인 입력과 출력정보로부터 perfect model-matching을 구현하기 위해 Monopoli가 제안한 ‘augmented error method’를 사용하였다. Adaptive Law를 유도하기 위해, Lyapunov 안정도이론을 적용하였고, 이에 따라 primary controller내의 파라미터 값을 변화시킨다.

시뮬레이션 및 실험을 통해 PI제어와 제시된 방법의 특성을 비교, 분석하여 제안된 제어시스템의 유용성을 확인하였다.

2장에서는 유도전동기 벡터제어에 대해 설명하고 3장에서는 공간전압벡터 펄스폭변조방식(SVPWM) 인버터의 원리에 대해서 기술한다. 4장에서는 MRAC개념 및 이를 적용한 속도추정기를 설계할 것이다. 5장에서는 2, 3, 4장의 이론을 바탕으로 하여 컴퓨터시뮬레이션을 행해 타당성을 확인하고 마이크로 프로세서, 인터페이스회로, 전력변환회로등을 구현해 실제 실험을 행하였고 6장에서는 본 논문의 결론을 맺는다.

2 유도전동기 벡터제어

2.1 벡터제어

교류기의 전기자에는 통상 3상권선이 연결되어 있고, 3상권선에 흐르는 전류와 이것에 의해 발생하는 자속을 벡터로 취급하는 것에는 3상권선을 등가의 2상권선으로 표현하는 것이 편리하다.

Fig. 2-1, Fig. 2-2는 3상권선 및 2상권선의 각 상 전류벡터, 자속벡터를 나타내고 있다. 전류 및 자속벡터의 방향은 각 권선 전류를 함께 유입했을 때의 극성을 나타내고 있다. 3상권선에서 발생하는 각 상 벡터전류 i_{1a}, i_{1b}, i_{1c}로부터 합성전류벡터는 $\boldsymbol{i}_1$이 되고, $\boldsymbol{i}_1$과 같은 합성전류벡터를 만드는 2상권선 전류 $i_{1\alpha}, i_{1\beta}$는 Fig. 2-2에 나타내었다.

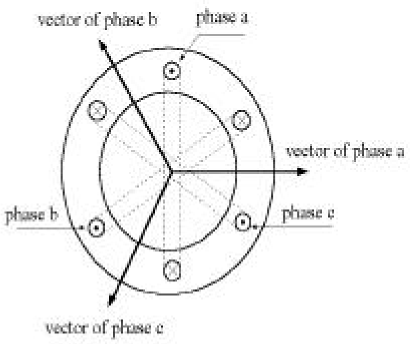


Fig. 2-1 Three phase winding

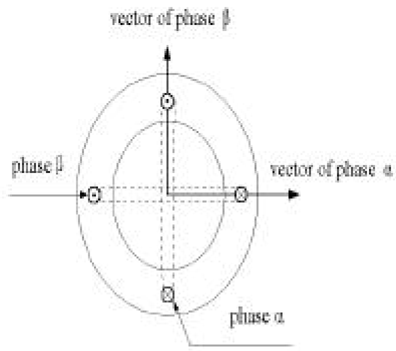


Fig. 2-2 Equivalent two phase winding

편의상 유도전동기는 각상이 1회권으로 구성되는 2극기라는 가정에서

관계식을 도출한다. 복소평면상에 표시된 3상의 공간전류벡터는 다음과 같이 복소수로 표현될 수 있다[14].

$$\boldsymbol{i_{1a}} = i_{1a}(t)$$
$$\boldsymbol{i_{1b}} = a\ i_{1b}(t) \qquad (2\text{-}1)$$
$$\boldsymbol{i_{1c}} = a^2 i_{1c}(t)$$

단, $a = e^{j\frac{2}{3}\pi} = \cos\frac{2\pi}{3} + j\sin\frac{2\pi}{3}$

$a^2 = e^{j\frac{4\pi}{3}} = \cos\frac{4\pi}{3} + j\sin\frac{4\pi}{3}$

그러면 이들 3상 공간전류벡터의 합성전류벡터는 다음과 같게 된다.

$$\boldsymbol{i_1} = \boldsymbol{i_{1a}} + \boldsymbol{i_{1b}} + \boldsymbol{i_{1c}}$$

$$= i_{1a}(t) + a i_{1b}(t) + a^2 i_{1c}(t)$$

$$= \frac{3}{2} I_m (\cos\omega t + j\sin\omega t)$$

$$= \frac{3}{2} I_m e^{j\omega_e t} \qquad (\text{단}, \gamma = 0) \qquad (2\text{-}2)$$

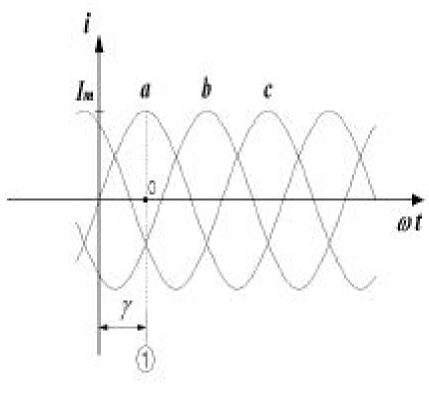


Fig. 2-3 Position of current vector

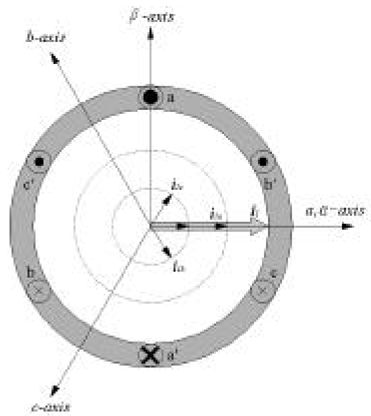


Fig. 2-4 Space current vector

Fig. 2-4는 Fig. 2-3의 ①위치에서의 공간전류벡터를 나타낸 것이며 $\gamma = \frac{\pi}{2}$이고, 벡터 회전 속도 ω_e와 회전각 θ_e는 "0"인 상태이다.

따라서, 합성공간전류벡터는 회전자계와 마찬가지로 크기가 전류 최대치의 1.5배로 일정하고 동기각속도 ω_e로 회전한다는 것을 알 수 있다.

그러나 좌표변환의 편의를 위해서 3상 공간전류벡터의 합성공간전류벡터는 식(2-2)을 사용하지 않고 다음과 같이 정의하기로 한다.

$$\boldsymbol{i}_1 = \frac{2}{3}(\boldsymbol{i}_{1a} + \boldsymbol{i}_{1b} + \boldsymbol{i}_{1c})$$

$$= \frac{2}{3}[\, i_{1a}(t) + a i_{1b}(t) + a^2 i_{1c}(t)\,] = I_m e^{j\omega_e t} \quad (2\text{-}3)$$

마찬가지 방법으로 고정자의 공간전압벡터와 공간 쇄교자속벡터는 각각 다음과 같이 표현될 수 있다.

$$\boldsymbol{v}_1 = \frac{2}{3}[v_{1a}(t) + a v_{1b}(t) + a^2 v_{1c}(t)] \quad (2\text{-}4)$$

$$\boldsymbol{\phi}_1 = \frac{2}{3}[\phi_{1a}(t) + a\phi_{1b}(t) + a^2\phi_{1c}(t)] \quad (2\text{-}5)$$

2.2 좌표 변환

전장에서 보인 3상 고정좌표계로 표시한 a, b, c 축의 변수들을 2축(α, β축) 고정좌표계로 변환하면 독립변수의 수가 감소하여 유도전동기의 해석을 용이하게 한다.

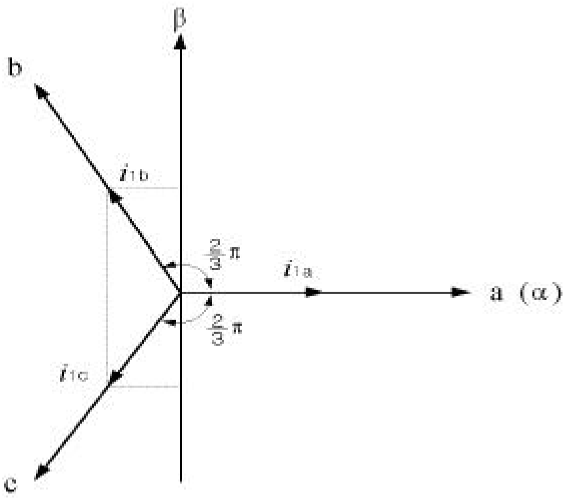


Fig. 2-5 Change abc axis to $\alpha\beta$ axis

예를 들어 전류의 경우에는 다음과 같이 나타낼 수 있으며,

$$
\begin{aligned}
\boldsymbol{i}_1 &= \frac{2}{3}[\boldsymbol{i}_{1a} + \boldsymbol{i}_{1b} + \boldsymbol{i}_{1c}] \\
&= \frac{2}{3}\left[\left(i_{1a}(t) - \frac{1}{2}i_{1b}(t) - \frac{1}{2}i_{1c}(t)\right) + j\left(\frac{\sqrt{3}}{2}i_{1b}(t) - \frac{\sqrt{3}}{2}i_{1c}(t)\right)\right] \\
&= i_{1\alpha} + j\,i_{1\beta} \qquad (2\text{-}6)
\end{aligned}
$$

이를 행렬 형태로 표현하면 식(2-7)과 같다.

$$
\begin{bmatrix} i_{1\alpha} \\ i_{1\beta} \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} i_{1a}(t) \\ i_{1b}(t) \\ i_{1c}(t) \end{bmatrix} \qquad (2\text{-}7)
$$

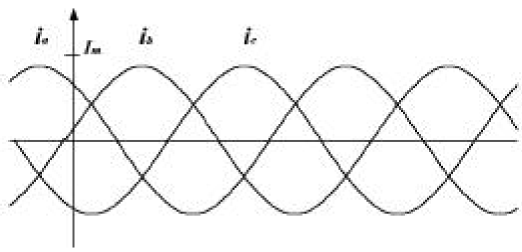
이 좌표변환(3상-2상 변환)은 전류의 순시치에 대해서도 항상 성립하며, 3상전류는 반드시 정현파가 아니어도 좋다.

Fig. 2-6은 앞의 변환결과를 종합한 것이다. 여기서 α, β권선은 어디까지나 계산의 형편상 가정한 것이고, 이것은 어디에 위치해도 좋지만,

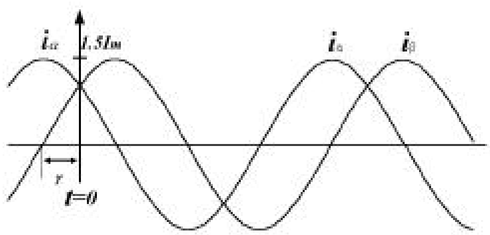
α권선을 a상권선의 위치와 일치시켜 생각하는 것이 편리하다. 또한 2상으로부터 3상으로 돌아갈 때의 변환, 2상-3상 변환은 식(2-7)을 역으로 계산하면 좋고, 이 관계는 다음 식으로 표현된다.

$$\begin{bmatrix} i_{1a}(t) \\ i_{1b}(t) \\ i_{1c}(t) \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & 0 \\ -\frac{1}{3} & \frac{1}{\sqrt{3}} \\ -\frac{1}{3} & -\frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} i_{1\alpha} \\ i_{1\beta} \end{bmatrix} \tag{2-8}$$

이상의 3상-2상 변환 및 2상-3상 변환은 전류에 한하지 않고 자속 및 전압에 있어서도 성립한다.



(a) Three phase of sine wave current



(b) Two phase of sine wave current

Fig. 2-6 Phase transformation of sine wave current

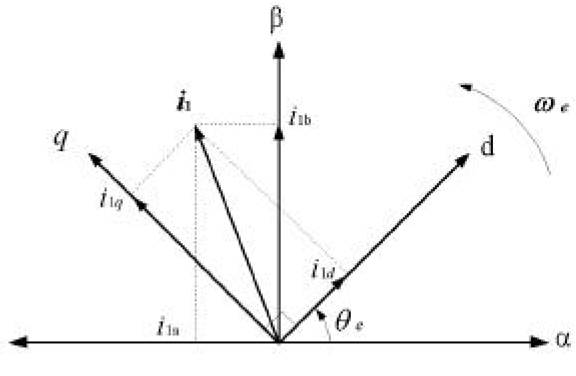


Fig. 2-7 Relation between stationary and synchronously reference frame

2축 고정좌표계로 변환된 $\alpha\beta$축 전류를 Fig. 2-7과 같이 다시 동기회전좌표계로 변환하여 나타내면 식(2-9)이 되고,

$$
\begin{aligned}
\boldsymbol{i_{1dq}} &= (i_{1\alpha} + j\, i_{1\beta})[\cos(-\theta_e) + j\sin(-\theta_e)] \\
&= i_{1\alpha}\cos\theta_e + i_{1\beta}\sin\theta_e + j(-i_{1\alpha}\sin\theta_e + i_{1\beta}\cos\theta_e) \\
&= i_{1d} + j\, i_{1q} \qquad (2\text{-}9)
\end{aligned}
$$

이를 행렬 형태로 표현하면 식(2-10)가 된다.

$$\begin{bmatrix} i_{1d} \\ i_{1q} \end{bmatrix} = \begin{bmatrix} \cos\theta_e & \sin\theta_e \\ -\sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} i_{1\alpha} \\ i_{1\beta} \end{bmatrix} \qquad (2\text{-}10)$$

식(2-10)는 변환행렬의 역변환에 의해 쉽게 dq축을 $\alpha\beta$축으로 변환이 가능하며 좌표변환식을 나타내면 식(2-11)과 같다.

$$\begin{bmatrix} i_{1\alpha} \\ i_{1\beta} \end{bmatrix} = \begin{bmatrix} \cos\theta_e & -\sin\theta_e \\ \sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} i_{1d} \\ i_{1q} \end{bmatrix} \qquad (2\text{-}11)$$

2.3 간접벡터제어에서의 슬립주파수

2.3.1 고정좌표계에서의 유도전동기 모델링

공간벡터를 사용한 3상 고정좌표계의 고정자 전압방정식과 3상 회전좌표계의 회전자 전압방정식은 식(2-12), 식(2-13)으로 나타낼 수 있다.

$$\boldsymbol{v}_1 = R_1 \boldsymbol{i}_1 + \frac{d}{dt}\boldsymbol{\phi}_1 \tag{2-12}$$

$$\boldsymbol{v}_{2r} = R_{2r} \boldsymbol{i}_{2r} + \frac{d}{dt}\boldsymbol{\phi}_{2r} \tag{2-13}$$

회전자의 회전좌표계에서의 값들을 고정좌표계로 변환하여 나타내면 다음과 같다[20].

$$\boldsymbol{v}_2 = ne^{j\theta_r} \cdot \boldsymbol{v}_{2r}\ ,\ \boldsymbol{i}_2 = \frac{e^{j\theta_r}}{n} \cdot \boldsymbol{i}_{2r}\ ,\ \boldsymbol{\phi}_2 = ne^{j\theta_r} \cdot \boldsymbol{\phi}_{2r}\ ,\ R_{2r} = \frac{1}{n^2}R_2 \tag{2-14}$$

단, $n = \frac{N_1}{N_2}$ (N_1 : 고정자측의 상당권선수

N_2 : 회전자측의 등가권선수)

식(2-13) 첫째 항은 다음과 같고,

$$\boldsymbol{R}_{2r}\boldsymbol{i}_{2r} = \frac{1}{n^2}\boldsymbol{R}_2 \cdot \frac{n}{e^{j\theta_r}}\boldsymbol{i}_2 = \frac{e^{-j\theta_r}}{n}R_2\boldsymbol{i}_2 \tag{2-15}$$

또 식(2-13) 우변 둘째 항은 다음과 같이 된다.

$$\frac{d}{dt}\boldsymbol{\phi}_{2r} = \frac{d}{dt}\left(\frac{e^{-j\theta_r}}{n}\boldsymbol{\phi}_2\right)$$

$$= \frac{e^{-j\theta_r}}{n}\left(\frac{d}{dt}\boldsymbol{\phi}_2 - j\omega_r\boldsymbol{\phi}_2\right) \tag{2-16}$$

따라서, 식(2-14), (2-15), (2-16)을 식(2-13)에 대입하면 고정좌표계로

변환하여 표현한 회전자 전압방정식(2-17)**을 얻게된다**.

$$\boldsymbol{v}_2 = R_2 \boldsymbol{i}_2 + \frac{d}{dt}\boldsymbol{\phi}_2 - j\omega_r \boldsymbol{\phi}_2 \tag{2-17}$$

여기서 전류 및 인덕터로 표현된 쇄교자속벡터는

$$\begin{bmatrix} \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_2 \end{bmatrix} = \begin{bmatrix} L_1 & M \\ M & L_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{i}_1 \\ \boldsymbol{i}_2 \end{bmatrix} \tag{2-18}$$

즉, $\boldsymbol{\phi}_1 = L_1 \boldsymbol{i}_1 + M \boldsymbol{i}_2$ (2-19)

$$\boldsymbol{\phi}_2 = M \boldsymbol{i}_1 + L_2 \boldsymbol{i}_2 \tag{2-20}$$

단, $L_1 = l_1 + M$, $L_2 = l_2 + M$

L_1, L_2 : 1, 2**차 인덕턴스**, M : **상호인덕턴스**

l_1, l_2 : 1, 2**차 누설인덕턴스**

로 나타낼 수 있고 식(2-12), **식**(2-17)**은 식**(2-21)**과 같은 행렬형태로 나타낼 수 있다**.

$$\begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{bmatrix} = \begin{bmatrix} R_1 + pL_1 & pM \\ (p - jw_r)M & R_2 + (p - jw_r)L_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{i}_1 \\ \boldsymbol{i}_2 \end{bmatrix} \tag{2-21}$$

식(2-21)**로 표현된 고정자 및 회전자의 전압방정식을 각각 분리하여** 2**축 고정좌표계로 표현하면 식**(2-22), **식**(2-23), **식**(2-24), **식**(2-25)**로 된다**.

$$v_{1\alpha} = (R_1 + pL_1)\, i_{1\alpha} + pM i_{2\alpha} \tag{2-22}$$

$$v_{1\beta} = (R_1 + pL_1)\, i_{1\beta} + pM i_{2\beta} \tag{2-23}$$

$$v_{2\alpha} = pM i_{1\alpha} + w_r M i_{1\beta} + (R_2 + pL_2) i_{2\alpha} + w_r L_2 i_{2\beta} \tag{2-24}$$

$$v_{2\beta} = -w_r M i_{1\alpha} + pM i_{1\beta} - w_r L_2 i_{2\alpha} + (R_2 + pL_2)\, i_{2\beta} \tag{2-25}$$

그러나, **농형 유도전동기에서** 2**차측 전압**($\boldsymbol{v}_2 = v_{2\alpha} + jv_{2\beta}$)**은 단락 되어 있으므로** “0”**이 된다**.

2.3.2 회전좌표계에서의 모델링

식(2-12)를 동기회전좌표계로 변환하면 식(2-26)이 되고,

$$\boldsymbol{v}_1 e^{j\theta_e} = R_1 \boldsymbol{i}_1 e^{j\theta_e} + p(\boldsymbol{\phi}_1 e^{j\theta_e})$$

$$\therefore \ \boldsymbol{v}_1 = R_1 \boldsymbol{i}_1 + (p + jw_e)\boldsymbol{\phi}_1 \tag{2-26}$$

식(2-17)을 동기회전좌표계로 변환하면 식(2-26)이 된다.

$$\boldsymbol{v}_2 e^{j\theta_e} = R_2 \boldsymbol{i}_2 e^{j\theta_e} + \boldsymbol{\phi}_2 e^{j\theta_e} - jw_r \boldsymbol{\phi}_2 e^{j\theta_e}$$

$$\therefore \ \boldsymbol{v}_2 = R_2 \boldsymbol{i}_2 + (p + jw_s)\boldsymbol{\phi}_2 \tag{2-27}$$

식(2-26), (2-27)을 행렬형태로 표현하면 식(2-28) 또는 식(2-29)와 같이 동기회전좌표계에서의 유도전동기 전압방정식이 얻어진다.

$$\begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{bmatrix} = \begin{bmatrix} R_1 + (p + jw_e)L_1 & (p + jw_e)M \\ (p + jw_s)M & R_2 + (p + jw_s)L_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{i}_1 \\ \boldsymbol{i}_2 \end{bmatrix} \tag{2-28}$$

또는,

$$\begin{bmatrix} v_{1d} \\ v_{1q} \\ v_{2d} \\ v_{2q} \end{bmatrix} = \begin{bmatrix} R_1 + pL_1 & -w_e L_1 & pM & -w_e M \\ w_e L_1 & R_1 + pL_1 & w_e M & pM \\ pM & -W_s M & R_2 + pL_2 & -w_s L_2 \\ w_s M & pM & w_s L_2 & R_2 + pM_2 \end{bmatrix} \begin{bmatrix} i_{1d} \\ i_{1q} \\ i_{2d} \\ i_{2q} \end{bmatrix} \tag{2-29}$$

식(2-29)를 다시 전개하여 정리하면 다음과 같다.

$$v_{1d} = R_1 i_{1d} + p(L_1 i_{1d} + M i_{2d}) - w_e(L_1 i_{1q} + M i_{2q})$$

$$= R_1 i_{1d} + p\phi_{1d} - w_e \phi_{1q} \tag{2-30}$$

$$v_{1q} = R_1 i_{1q} + p\phi_{1q} - w_e \phi_{1d}$$

$$v_{2d} = R_2 i_{2d} + p\phi_{2d} - w_s \phi_{2q} = 0$$

$$v_{2q} = R_2 i_{2q} + p\phi_{2q} + w_s\phi_{2d} = 0 \qquad (2\text{-}31)$$

농형 유도전동기에서는 회전자 전압($v_2 = v_{2d} + jv_{2q}$)이 “0”이 된다. 각각의 쇄교자속은 식(2-32), 식(2-33)과 같이 되며,

$$\phi_{1d} = L_1 i_{1d} + M i_{2d} = l_1 i_{1d} + M(i_{1d} + i_{2d})$$

$$\phi_{1q} = L_1 i_{1q} + M i_{2q} = l_1 i_{1q} + M(i_{1q} + i_{2q}) \qquad (2\text{-}32)$$

$$\phi_{2d} = L_2 i_{2d} + M i_{1d} = l_2 i_{2d} + M(i_{1d} + i_{2d})$$

$$\phi_{2q} = L_2 i_{2q} + M i_{1q} = l_2 i_{2q} + M(i_{1q} + i_{2q}) \qquad (2\text{-}33)$$

단, $L_1 = l_1 + M, \quad L_2 = l_2 + M$

위의 식(2-33)의 세 번째, 네 번째 식으로부터 식(2-34)를 얻을 수 있으며,

$$i_{2d} = \frac{\phi_{2d}}{L_2} - \frac{M}{L_2} i_{1d}$$

$$i_{2q} = \frac{\phi_{2q}}{L_2} - \frac{M}{L_2} i_{1q} \qquad (2\text{-}34)$$

식(2-34)를 유도전동기 회전자 전압방정식인 식(2-31)에 대입하면 식(2-35)와 같이 된다.

$$\frac{d}{dt}\phi_{2d} + \frac{R_2}{L_2}\phi_{2d} - \frac{R_2}{L_2} M i_{1d} - w_s\phi_{2q} = 0$$

$$\frac{d}{dt}\phi_{2q} + \frac{R_2}{L_2}\phi_{2q} - \frac{R_2}{L_2} M i_{1q} + w_s\phi_{2d} = 0 \qquad (2\text{-}35)$$

타여자 직류전동기와 같은 제어특성을 얻기 위해서는, 회전자 쇄교자속을 d축과 일치시켜 $\phi_{2q} = \frac{d}{dt}\phi_{2q} = 0$과 같이 제어하는 것이 바람직하므로 식(2-35)는 다음 식과 같이 된다.

$$\frac{d}{dt}\phi_{2d} + \frac{R_2}{L_2}\phi_{2d} - \frac{R_2}{L_2}Mi_{1d} = 0$$

$$-\frac{R_2}{L_2}Mi_{1q} + w_s\phi_{2d} = 0 \quad (2\text{-}36)$$

$$\frac{R_2}{L_2}\frac{d}{dt}\phi_{2d} + \phi_{2d} = Mi_{1d}$$

$$w_s = \frac{R_2}{L_2}M\frac{i_{1q}}{\phi_{2d}} \text{ (일반적인 경우)} \quad (2\text{-}37)$$

정상상태에서 회전자자속이 일정하게 되므로 $\phi_{2d}\ const. == \phi_2, \frac{d}{dt}\phi_{2d} = 0$ 이 되어 식(2-37)의 위 식은 $\phi_{2d} = Mi_{1d}$ 가 되어 다음과 같은 정상상태 식으로 된다.

$$w_s = \frac{1}{\tau}\frac{i_{1q}}{i_{1d}} \text{ (정상상태)} \quad (2\text{-}38)$$

단, τ : 2차시정수 $\left(\frac{L_2}{R_2}\right)$

3. 공간전압벡터 PWM 인버터

일반적인 3상 전압형 인버터의 구성도는 Fig. 3-1과 같이 나타낼 수 있다. 전력용 소자들의 ON-OFF 스위칭 과정에 의해서 직류가 원하는 주파수의 교류로 바뀌게 된다. 스위칭 과정에서는 다음과 같은 2가지 조건을 만족시키지 않으면 안 된다.

(1) 6개의 스위칭 소자 중 3개는 항상 ON 되어 있어야 하고, 3개는 항상 OFF 되어 있어야 한다.

(2) 같은 선상의 직렬로 연결된 상, 하 두 소자는 동시에 도통되는 일이 없도록 서로 상반된 신호에 의해서 동작되어야 하며, 과도시 도통되어 있는 시간이 겹치지 않도록 주의하여야 한다.

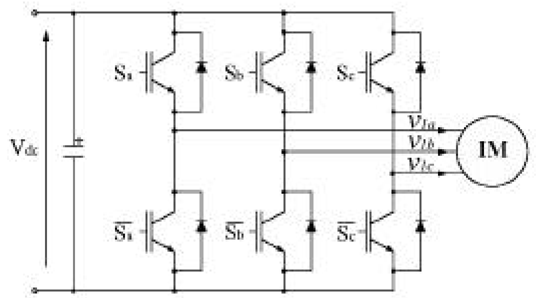


Fig. 3-1 Three-phase voltage source inverter and AC motor

조건을 고려하여, Fig. 3-1의 전력용 반도체 소자를 Fig. 3-2와 같이 스위치로 대치해서 생각하기로 한다.

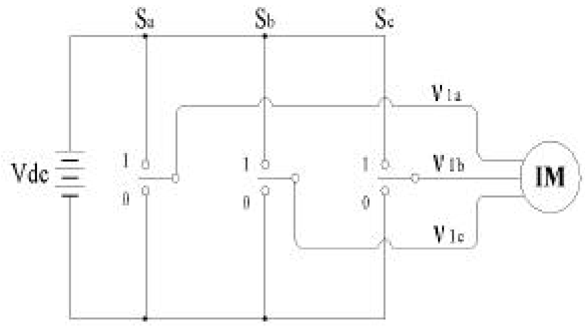
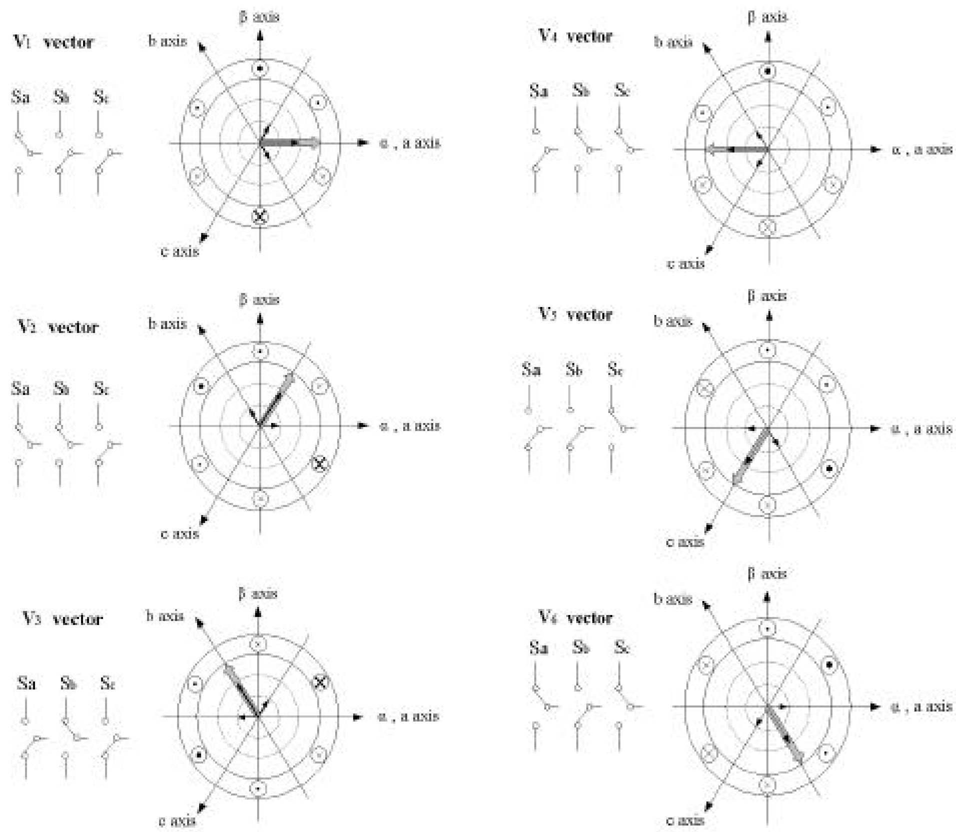


Fig. 3-2 Basic scheme of 3-phase inverter and AC motor

3개의 스위치 Sa, Sb, Sc의 ON-OFF스위칭 동작에 의해서 만들어질 수 있는 합성 출력 전압벡터는 8가지가 있으며, 이중 3개의 스위치가 모두 ON이거나 모두 OFF인 경우는 전압을 발생시킬 수 없는 무효벡터 (V_0, V_7)가 되고, 유효벡터는 Fig. 3-3과 같이 V_1~ V_6벡터로 나타내 진다. 여기서 스위치가 "1"로 표시된 것은 ON을 뜻하며 상하 반도체소자 중 위쪽의 소자가 ON되고 아래쪽 소자가 OFF된 것을 의미한다.

Fig. 3-3은 각각의 스위칭에 의해 생성될 수 있는 6가지 공간전압벡터를 위치와 크기로 나타낸 것으로 크기는 직류입력전압 V_{dc}크기의 2/3배로 일정하고 60° 의 위치의 차를 가지고 회전하고 있음을 보여주고 있다.



(a) Basic Vector V_1, V_2, V_3 (b) Basic vector V_4, V_5, V_6

Fig. 3-3 Complementary switching and space voltage vectors

Fig. 3-4는 스위칭에 의해 형성된 6개 벡터의 위치와 유도전동기 1차측(고정자 측)으로 공급되는 3상전압의 관계를 나타내고 있다. Fig. 3-3으로 표현된 6개의 공간전압 벡터는 원점을 중심으로 각각 60˚ 의 위치차를 가지고 있으므로 이것을 한곳에 모아 보면 Fig. 3-5와 같이 되어 각각의 상태를 이해하는데 편리하다.

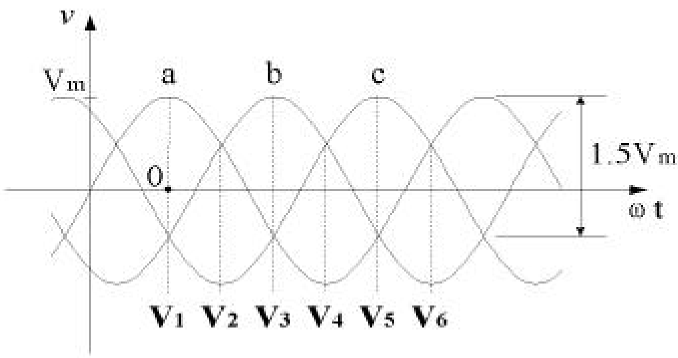


Fig. 3-4 Position of 6 basic vectors

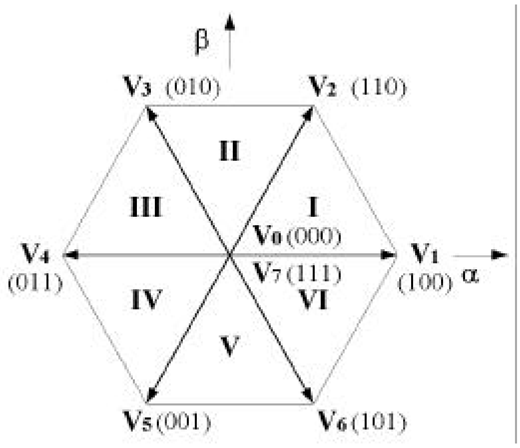


Fig. 3-5 SVPWM, vectors and sectors

공간벡터 PWM **스위칭 기법에 의해** AC**전동기에 원하는 상전압을 공급하기 위해서는 다음과 같은** 3**가지 중요한 단계를 거쳐야 한다**.

(1) **기준전압 형성** : Fig. 3-6**에서 나타낸** $\alpha\beta$**평면 내에 정확한 기준전압 벡터의 위치를 요한다**. **이것은 회전속도와 기준벡터**(v^*)**크기의 정확한 제어를 암시한다**.

(2) $\alpha\beta$평면 내에 존재하는 v^*의 위치를 적당한 스위칭 변수 S_a, S_b, S_c의 패턴으로 변환되어야 한다.

(3) 해당패턴의 스위칭에 의한 PWM을 발생시킨다.

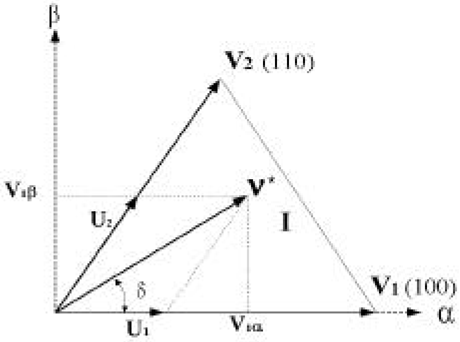


Fig. 3-6 Reference vector as a combination of adjacent vectors

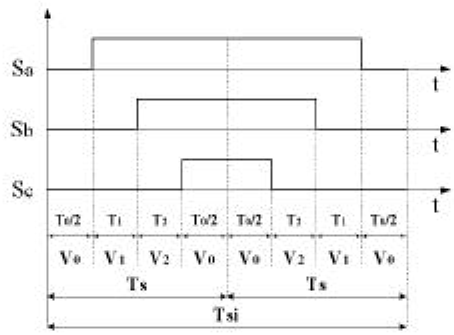


Fig. 3-7 Switching pattern of SVPWM in the sector I

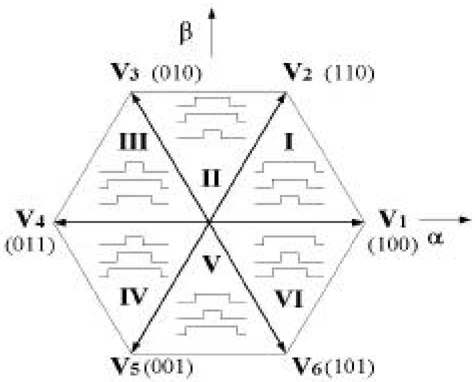


Fig. 3-8 Hexagon of SVPWM pattern

3.1 스위칭패턴과 기본벡터

스위칭변수벡터 $[S_a\ S_b\ S_c]^T$와 선간전압벡터 $[v_{ab}\ v_{bc}\ v_{ca}]^T$는 식(3-2)로 다음과 같이 주어진다.

$$\begin{bmatrix} v_{ab} \\ v_{bc} \\ v_{ca} \end{bmatrix} = V_{dc}\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}\begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix} \qquad (3\text{-}2)$$

또한, 상전압벡터 $[v_a\ v_b\ v_c]^T$는 식(3-3)과 같이 되고 여기서 V_{dc}는 DC 공급전압이다.

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \frac{1}{3}V_{dc}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}\begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix} \qquad (3\text{-}3)$$

스위칭의 조합에 의한 8가지 상태와 식(3-2)과 식(3-3)에 따라 Vdc에 의해 생성된 각각의 선간 전압, 상전압을 나타내면 Table 3-1과 같다.

α,β를 고정좌표축의 횡축, 종축이라 했을 때 식(3-4)로 주어진 $\alpha\beta$변환에 의해 3상전압벡터는 $\alpha\beta$축상의 벡터로 변환이 가능하여 이 $\alpha\beta$축상의 벡터는 3상전압의 공간벡터의 합을 나타낸다. 8개의 스위칭패턴에 해당하는 상전압은 Fig. 3-6에 나타낸 것과 같이 $\alpha\beta$변환에 의해 $\alpha\beta$평면에 대응시킬 수 있다.

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} \qquad (3\text{-}4)$$

8개의 스위칭에 의해 생성된 벡터는 결과적으로 6개의 유효벡터와 2개의 영벡터로 나타난다. 6개의 영이 아닌 벡터는 Fig. 3-5와 같이 6개의 축을 형성하고, 서로 인접한 벡터 사이의 각도는 60° 이다. 이 8개의 벡

터를 기본벡터라 부르고, $V_1, V_2, V_3 \cdots V_7$로 나타낸다.

SVPWM 기술의 목적은 8개의 스위칭패턴의 조합에 의해 기준전압벡터 $\boldsymbol{v}^*$의 값을 구하는 것이다. 기본벡터 사이에 존재하는 기준벡터($\boldsymbol{v}^*$)를 구하는 가장 간단한 수단 중 하나는 똑같은 주기 내에서 (짧은 주기 T_s내에서) 인버터 평균 출력을 구하는 것이다. 이 방법을 그림으로 도시하면 Fig 3-6과 같다. 본 논문에서는 기준 공간벡터를 구하기 위해서 3상 대칭변조법을 사용하였으며, 대칭PWM의 파형은 샘플링주기 내에 유효벡터를 중간에 위치시키고 무효벡터인 영벡터를 양옆에 반반씩 나누어 인가시킴으로써 얻을 수 있다. T_1, T_2는 샘플링주기동안에 인접한 두 기본벡터에 해당하는 스위칭 패턴에 인가된 시간이다. 이 시간에 의해서 인접한 두 기본벡터 축에 형성된 벡터는 각각 U_1, U_2가 되고 U_1, U_2벡터의 합성에 의해 기준벡터 $\boldsymbol{v}^*$가 해당 영역에 형성된다.

Table 3-1 Switching patterns and output voltage vectors

Voltage vector	S_a S_b S_c	$\boldsymbol{v}^*$	v_{ab} v_{bc} v_{ca}	v_a v_b v_c
$\mathbf{V}_0$	0 0 0	$0\angle 0°$	0 0 0	0 0 0
$\mathbf{V}_1$	1 0 0	$\frac{2}{3}V_{dc}\angle 0°$	V_{dc} 0 $-V_{dc}$	$\frac{2}{3}V_{dc}$ $-\frac{1}{3}V_{dc}$ $-\frac{1}{3}V_{dc}$
$\mathbf{V}_2$	1 1 0	$\frac{2}{3}V_{dc}\angle 60°$	0 V_{dc} $-V_{dc}$	$\frac{1}{3}V_{dc}$ $\frac{1}{3}V_{dc}$ $-\frac{2}{3}V_{dc}$
$\mathbf{V}_3$	0 1 0	$\frac{2}{3}V_{dc}\angle 120°$	$-V_{dc}$ V_{dc} 0	$-\frac{1}{3}V_{dc}$ $\frac{2}{3}V_{dc}$ $-\frac{1}{3}V_{dc}$
$\mathbf{V}_4$	0 1 1	$\frac{2}{3}V_{dc}\angle 180°$	$-V_{dc}$ 0 V_{dc}	$-\frac{2}{3}V_{dc}$ $\frac{1}{3}V_{dc}$ $\frac{1}{3}V_{dc}$
$\mathbf{V}_5$	0 0 1	$\frac{2}{3}V_{dc}\angle 240°$	0 $-V_{dc}$ V_{dc}	$-\frac{1}{3}V_{dc}$ $-\frac{1}{3}V_{dc}$ $\frac{2}{3}V_{dc}$
$\mathbf{V}_6$	1 0 1	$\frac{2}{3}V_{dc}\angle 300°$	V_{dc} $-V_{dc}$ 0	$\frac{1}{3}V_{dc}$ $-\frac{2}{3}V_{dc}$ $\frac{1}{3}V_{dc}$
$\mathbf{V}_7$	1 1 1	$0\angle 0°$	0 0 0	0 0 0

이 기준전압벡터 $\boldsymbol{v}^*$를 발생하기 위하여 인가하는 전압벡터 원리를 수식으로 표현하면 식(3-5)와 같다. 두 인접한 기본벡터에 인가한 시간 (T_1, T_2)의 합이 주기 T_s보다 작을 경우에는 나머지 시간만큼 영벡터 (000, 111)를 인가한다. T_0는 영벡터 인가 시간이다.

$$\int_0^{T_s} \boldsymbol{v}^* = \int_0^{T_1} V_1 + \int_{T_1}^{T_1+T_2} V_2 + \int_{T_1+T_2}^{T_s} V_7 \qquad (3\text{-}5)$$

단, $T_s = T_1 + T_2 + T_0$

식(3-5)에서 PWM주기가 작고 $\boldsymbol{v}^*$의 변화가 비교적 느리다고 가정하면 식(3-5)로부터 식(3-6)을 얻을 수 있다.

$$T_s \boldsymbol{v}^* = T_1 V_1 + T_2 V_2 + T_0 (V_0 \text{ or } V_7) \qquad (3\text{-}6)$$

$$\boldsymbol{v}^* = \frac{T_1}{T_s} V_1 + \frac{T_2}{T_s} V_2 = U_1 + U_2 \qquad (3\text{-}7)$$

식(3-6)의 오른편 3번째 항은 왼편의 벡터합성에 영향을 주지 않는다.

U_1, U_2 및 영벡터를 인가할 시간 T_1, T_2 및 T_0은 다음과 같이 계산된다. U_1, U_2, 및 $\boldsymbol{v}^*$를 $\alpha\beta$축 값으로 나타내면 다음과 같다.

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = T_1 \frac{2}{3} V_{dc} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + T_2 \frac{2}{3} V_{dc} \begin{bmatrix} \cos 60° \\ \sin 60° \end{bmatrix} = T_s \frac{2}{3} V_{dc}\, a \begin{bmatrix} \cos\delta \\ \sin\delta \end{bmatrix}$$

$\boldsymbol{v}^*$의 α축의 값 : $T_1 \frac{2}{3} V_{dc} + T_2 \frac{2}{3} V_{dc} \cos 60° = T_s \frac{2}{3} V_{dc}\, a \cos\delta$

$\boldsymbol{v}^*$의 β축의 값 : $T_2 \frac{2}{3} V_{dc} \sin 60° = T_s \frac{2}{3} V_{dc}\, a \sin\delta$

$$\therefore T_2 = T_s a \frac{\sin\delta}{\sin\frac{\pi}{3}}$$

$$T_1 = T_s a \left(\frac{\sin(\frac{\pi}{3} - \delta)}{\sin\frac{\pi}{3}} \right)$$

단, $0 \le \delta < 60°$, $a = \frac{|v^*|}{\frac{2}{3}} V_{dc}$

δ는 V_1과 v^*가 이루는 각을 나타낸다.

$$T_1 = T_s \frac{|v^*|}{\frac{2}{3} V_{dc}} \frac{\sin(\frac{\pi}{3} - \delta)}{\sin\frac{\pi}{3}} = \sqrt{3}\, T_s \frac{|v^*|}{V_{dc}} \sin(\frac{\pi}{3} - \delta) \qquad (3\text{-}8)$$

$$T_2 = T_s \frac{|v^*|}{\frac{2}{3} V_{dc}} \frac{\sin\delta}{\sin\frac{\pi}{3}} = \sqrt{3}\, T_s \frac{|v^*|}{V_{dc}} \sin\delta \qquad (3\text{-}9)$$

$$T_0 = T_s - (T_1 + T_2)$$

지령전압벡터 즉 기준벡터가 Fig. 3-5와 같은 정육각형의 내부에 존재하게 되면 위의 스위칭시간이 유효하지만 이를 벗어나는 큰 전압일 경우에는 $T_1 + T_2 > T_s$가 되어 원하는 출력전압을 정해진 주기 내에 얻을 수 없게 된다. 이러한 경우를 인버터가 포화되었다고 하며, 이러한 과변조의 경우에는 방향은 그대로 유지하고 크기를 주기 내에서 낼 수 있는 최대전압인 정육각형 상에 존재하도록 스위칭 시간을 조절한다. 이러한 경우에는 $T_1 + T_2 = T_s$, $T_0 = 0$ 이 되어 유효벡터만 인가되고 영벡터는 인가되지 않는다.

공간전압벡터 PWM 기법은 두 전압벡터를 시간적인 평균으로 합성하여 지령전압에 해당하는 출력전압을 발생하며, 이를 수행하는 과정에서

인버터의 각 스위치를 한 샘플링주기에 한번씩만 ON되거나 OFF되도록 하는 3상 대칭변조법을 사용하는 경우에 영역Ⅰ Ⅵ에 대한 스위칭 패턴을 도시하면 Fig. 3-9와 같다. 그런데, 여기서 영역Ⅱ,Ⅳ,Ⅵ의 경우에는 그림에서와 같이 스위칭 동작을 수행하면 한 주기에 여러 번 ON, OFF가 이루어지게 되므로 이 영역에서는 두 전압벡터의 인가순서를 바꾸어 영역 Ⅰ,Ⅲ,Ⅴ에서처럼 한 주기에 한번씩만 ON, OFF가 이루어지도록 한다.

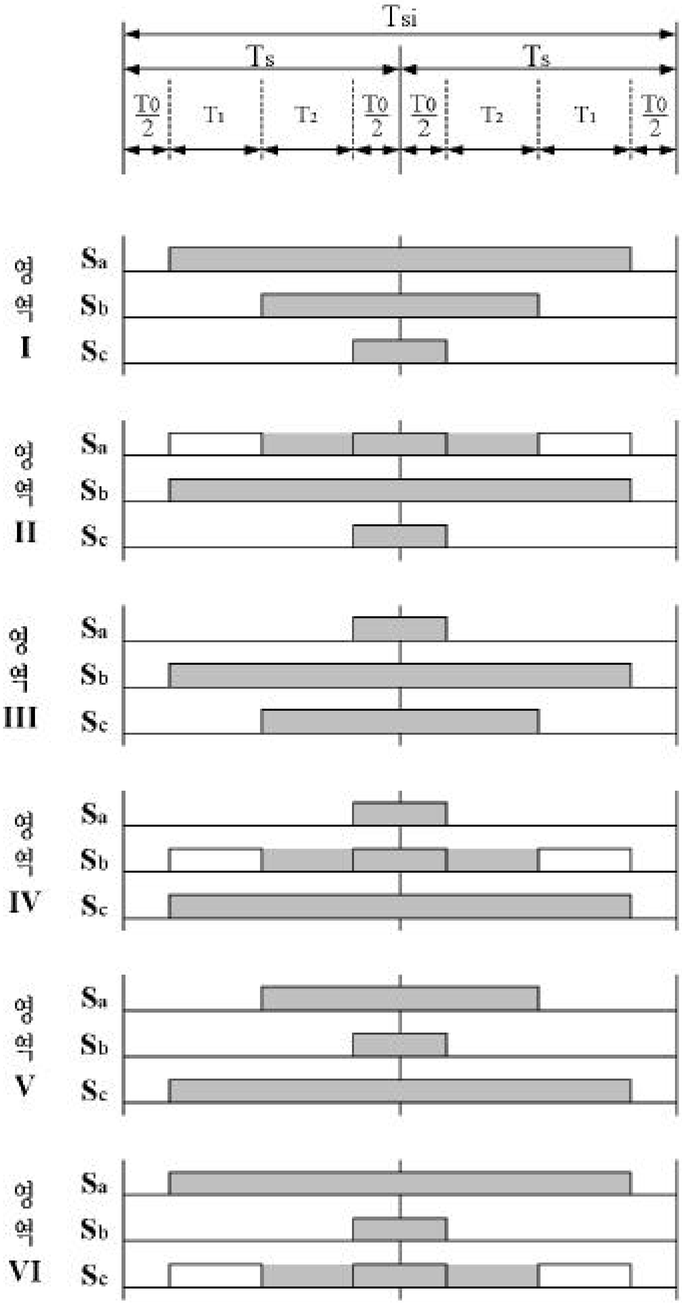


Fig. 3-9 Switching sequence at each sector

4. MRAC를 적용한 속도 추정기

4.1 MRAC의 개념

Fig. 4-1은 MRAC의 가장 일반적인 형태를 보이며, 1958년 Whitacker에 의해 처음으로 소개되었다. 이 방법에서, primary controller는 non-adaptive제어에서처럼 안정한 폐루프 응답을 얻기 위해 사용한다. 지령신호에 대한 원하는 프로세스응답은 시변으로 정의된 reference 모델에 의해서 결정되기 때문에, 폐루프 동작의 경우처럼 모든 환경내에서 쓰일 수 있는 primary controller의 고정 파라미터를 얻을 수 없다.

Adaptation mechanism은 프로세스출력 y_p가 모델출력y_m를 추종하도록 하고, 이들 출력사이의 오차를 영으로 만드는 안정한 파라미터를 설정한다. 만약 프로세스출력(y_p)과 프로세스상태(x_p) 그리고 프로세스입력(u)나 reference신호(r)를 이용할 수 있다면, adaptation mechanism 내에서 사용할 수 있다.

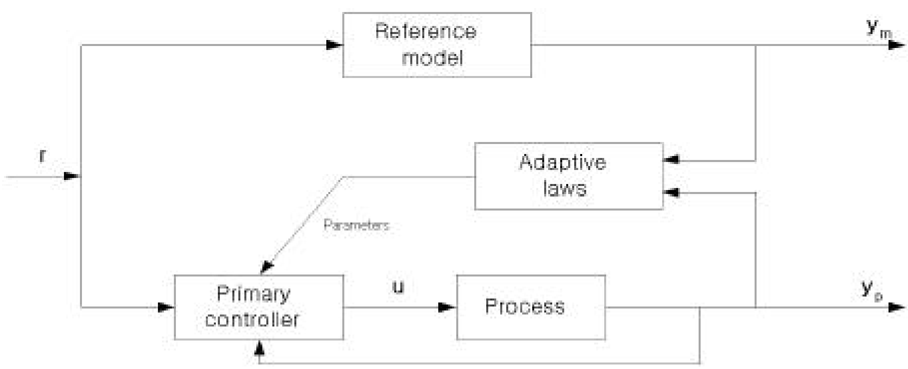


Fig. 4-1 General parallel MRAC scheme

Fig. 4-1은 두 개의 피드백루프(primary controller를 포함하는 내부루프와 adaptation mechanism을 포함하는 외부루프)로 구성되었고, 프로세스에 병렬로 reference 모델이 연결된 경우를 보인다. MRAC방법이 항상 그림과 같지 않으며 직렬형태의 MRAC등의 많은 경우가 있다. 직렬의 경우, reference 모델은 프로세스의 앞부분에 위치하며, 궤도생성기처럼 간주된다. 폐루프가 보통 단계적으로 설정점을 변화시킬 수 없는 반면에, 더 원만한 reference 모델 출력에 의해 원하는 프로세스응답이 달성될 수 있다. 직렬구조는 프로세스 입력포화문제(van Amerongen 1982)에 대한 해와 primary 피드백루프의 게인을 높게 설정할 수 있는 것과 같은 몇몇 이점이 있으며, self-tuning과 predictive adaptive와 같은 다른 적응제어에서도 직렬구조는 나타나며, 보통 reference generator로 불린다. 직렬 MRAC에 병렬을 더한 구조를 소위 series-parallel이라 한다. 이 경우, reference 모델은 두부분(프로세스에 직렬인 부분과 병렬인 부분)으로 나뉜다. 이 경우는 제어기파라미터를 조정하는 보통의 경우 대신 모델이 조정되는 추정에 사용된다. Passenier는 추정에 사용될 때의 가변구조의 이점과 결점을 설명했다.

MRAC의 중요한 논점은 adaptative law의 설계이다. 먼저 adaptative law는 sensitivity 모델을 사용해 디자인하고, 후에 Lyapunov 안정도 이론과 Popov 초안정도(hyperstability)이론을 사용해 안정한 적응시스템을 증명한다[24][27][28].

Adaptive controller design

The process

프로세스는 이상적인 상태를 만족해야한다. 예를 들어, 적응제어이론을 사용하기 위해 프로세스는 선형이어야 하고 완전하게 외란에 자유로워야 하며, 프로세스전달함수는 최소상태여야 한다.

그러나, 실제 프로세스는 언급된 요구조건을 모두 만족시킬순없다. 차수는 정확히 알 수 없으며, 선형이거나 외란에 자유로울 수도 없다. 프로세스의 DC 이득의 부호는 대체로 알 수 있다. 그렇지만, 프로세스에 대한 요구조건을 결코 만족할 수 없다하더라도 적응제어기는 실행될 수 있다. 즉, 적응제어기가 다소 간략화된 프로세스모델을 사용해 설계된다는 것을 의미한다. 그러므로 적응제어기가 불완전한 상태, 즉 실제 프로세스에서, 어떻게 동작하는지를 아는 것이 중요하다.

The primary controller

primary controller는 선형제어기 디자인에서 잘 알려진 구조를 가질 수 있지만, 다음과 같은 몇몇 중요한 요구조건을 만족해야한다.

첫째로, 폐루프 동작이 reference 모델응답과 동일하게 되도록 제어기 파라미터를 설정하는 perfect model-matching상태를 만족해야한다. 이는 primary controller의 구조와 차수가 프로세스와의 관계 속에서 결정하여야한다. 예를 들어, PID알고리즘의 경우 낮은 차수의 프로세스에 대한 요구조건을 만족해야 한다.

둘째로, direct adaptation을 적용하기 위해, 제어신호 u는 파라미터의 선형함수여야한다. primary controller는 MRAC에서 사용하며, θ는 제어

기 파라미터 벡터, 제어신호 u는 $u = \theta^T \omega$의 형태이다. 만약 프로세스상태벡터 x_p를 완벽하게 알 수 있다면, reference신호 r을 가지고 합성된 x_p는 보통 벡터 ω의 형태이고, 신호생성기부분(보조신호)은 불필요하다. 그러나 프로세스의 출력만이 측정된다면, 신호벡터(signal vector)의 생성을 위해서 보조신호생성기가 필요하다. 페루프 동작을 위한 제어기 파라미터 θ의 값은 θ^*로 표시한 reference 모델을 따라간다.

The reference model

원하는 프로세스동작을 결정하는 reference 모델은 보통 비선형시변형태로 주어지고, 제어계산기내에서 수행된다. 만약 상태피드백을 사용한다면, reference 모델은 항상 완전한 상태벡터를 만들지만, 출력피드백의 경우에는 입-출력관계에 의해 동작이 결정된다. 위에 언급한 perfect model-matching상태의 요구조건(모델의 관계정도(the pole excess)는 프로세스의 관계정도와 동일하게 존재해야한다)을 reference 모델에 적용한다. 안정성, 제어성 그리고 최소상태(phase)를 추가적으로 요구한다. 이는 단지 이론적인 요구이다. 그러나, 실제적으로 몇몇 이론적인 요구조건이 만족될 수 없다는 것에 주의하여 reference 모델은 분별 있게 선택해야한다(실제적으로 프로세스 출력이 reference 모델 출력을 따라갈 수 있다는 관점에서). 예를 들어, 만약 응답이 매우빠르게 reference 모델을 선택한다면, 제어신호 u를 매우 크게 할 필요가 있고, 이는 적응을 방해하는 입력포화효과나 모델화되지않은 동역학을 야기할 수도 있다.

실제적인 프로세스동작이 알려지지 않거나 변하기 때문에 reference 모델의 선택은 항상 명백하지 않으며, 또 너무 주의 깊은 reference 모델의 선택은 필요한 응답보다 더 늦은 페루프 응답을 야기시킬수 있다.

reference 모델의 선택은 실제 프로세스수행에 관계되므로, 결과적으로 reference 모델을 조절함으로써 적응을 행할 수 있다.

Derivation of adaptive laws

adaptive law의 유도에는 여러 가지 방법이 있지만, Lyapunov 안정도에 의한 방법을 설명한다.

Lyapunov's method

MRAC의 비선형, 시변특성 때문에, 선형안정도분석은 적용할 수 없다. 그래서 적응시스템의 전체적인 안정도를 증명하기 위해 잘 알려진 방법인 Lyapunov's direct method(Parks,1966)을 사용한다. Lyapunov's method는 만약 Lyapunov 함수 $V(x)$가 아래조건을 만족한다면, 시스템이 일양점근안정평형 $x = 0$을 가짐을 설명한다.

$V(x) > 0$ for $x \neq 0$ (positive definite)

$\dot{V}(x) < 0$ for $x \neq 0$ (negative definite)

$V(x) \rightarrow \infty$ for $||x|| \rightarrow \infty$

$V(0) = 0$

그러므로, Lyapunov function V는 시간에 따라 감소해야한다. 적응시스템디자인내에서의 Lyapunov's method를 사용하는 것에 대한 안정도 요구조건은 아래의 방법에 따라 직접적으로 adaptive law로 전환된다.

· 첫째단계는 에러식($e = y_p - y_m$ 이나, $e = x_p - x_m$)의 유도이다. 에러식 유도에서 많은 자유도가 존재하지만 편리성을 위해서 비선형입력특성을

고려한 선형전달함수로 쓰여져야 한다.

· 둘째, Lyapunov function은 signal error와 parameter error의 관계를 사용해 함수를 선택한다. 즉 signal error vector $e = x_p - x_m$ 와 parameter error vector $\phi = \theta - \theta^*$ 로 구성된 시스템상태의 최소선택에 기초한다. 가장 간단한 Lyapunov function V는 아래와 같다.

$$V = e^T P e + \phi^T \Gamma^{-1} \phi$$

행렬 P와 Γ^{-1}는 양의 한정이어야 한다.

· 셋째, V의 시간도함수를 계산해야한다. $\dot{V}$이 음의 한정이면 일양점근안정이 증명된다. 일반적으로 $\dot{V}$의 형태는 아래와 같다.

$$\dot{V} = -e^T Qe + \{\text{some terms including } \phi\}$$

ϕ를 포함하는 식을 영으로 만듦으로써 Q가 양의 한정인 e를 고려해 $\dot{V}$이 음의 한정임을 증명한다. 양의 대칭행렬 Q는 아래의 식에 의해 양의 대칭행렬 P를 만든다.

$$A^T P + PA = -Q$$

에러식의 선형부분이 보통 reference 모델전달함수를 포함하기 때문에, 위식의 행렬 A는 점근 안정한 시스템을 나타내고, 양의 대칭핼렬 Q는 항상 양의 대칭행렬 P를 만든다.

$\dot{V}$의 음의 한정성에서 $e = 0$는 점근안정평형점을 의미하지만 $\phi = 0$는 단지 안정하다는 것을 의미하므로, $\dot{V}$은 ϕ가 아니라 e의 함수처럼 나타난다. 그러므로 θ의 θ^*로 수렴은 증명하지 않는다. 그러나 reference signal이 충분히 큰 주파수 성분을 가지는 경우, signal vector

가 끊임없이 입력된다면, $\phi = 0$ 는 점근안정 평형이다. 아래의 예를 고려해보면, 이는 명백하다. 만약 reference signal이 영이라면 모든 시스템 상태는 영이고 그래서 프로세스 상태는 모델의 상태와 동일하다. 적응은 일어나지 않고, 부정확한 파라미터 설정은 유지된다.

· 넷째, 여분의 표현을 영으로 놓음으로써 adaptive law를 유도하며, 일반적으로 아래의 형태이다.

$\dot{\theta} = -\Gamma\varepsilon\xi$

ε은 e에 직접 관계되고, ξ는 ω의 변형된 표현이다. 이런 변형은 안정한 적응동작을 달성하기 위한 것이다. ε은 보통 상태에러($e = x_p - x_m$)의 선형 합성 형태인 $p^T e$나 출력에러($e_1 = y_p - y_m$)의 증가된 형태 $e_1 + v$이다. ξ는 ω나 ω의 필터된 형태로 구성된다.

MRAC using output feed back

단지 프로세스 출력만 이용할 수 있다면, 상태 피드백은 더 이상 사용될 수 없고, 다른 primary controller가 perfect model-matching상태를 만족하기 위해 디자인되어야 한다. adaptive law의 유도 역시, vector e의 부재에 의한 에러보상기를 사용할 수 없기 때문에 다르다. 그러므로 에러식의 SPR(Strictly Positive Real)특징은 다른 방법으로 달성되어야 한다.

출력피드백을 가진 MRAC는 완전한 프로세스 상태벡터 x_p의 부재를 보상하기 위해 출력에러($e_1 = y_p - y_m$)에 증가 신호 v를 사용한다. 이 방법은 Monopoli에 의해 처음으로 소개되었고, ‘augmented error method'

로 불린다. 이 방법은 필터가 y_p와 u로부터 신호벡터를 생성하기 위해 사용된다는 관점에서 Landau의 방법과 유사하다. 그러나, SPR 에러식을 달성하는 방법이 다르고, 에러 증가의 수단으로 사용한다. 후에, Narendra 와 Valavani(1978)은 본질적으로 동일하지만 약간 다른 방법을 보인다.

증가된 에러방법은 특정한 primary 제어기와 adaptive law의 합성이다. 만약 프로세스가 low damping ratio을 가진 극들을 가진다면, 이 구조는 원하는 응답에 늦게 수렴함이 나타났다. ASG에 의해 만들어진 signal vector를 변형시키는 것은 시스템의 동작을 향상시킨다.

The augmented error method

augmented error method의 primary controller 구조는 Fig. 4-2(Narendra et al., 1980: Narendra and Annaswamy, 1989)에 보인다.

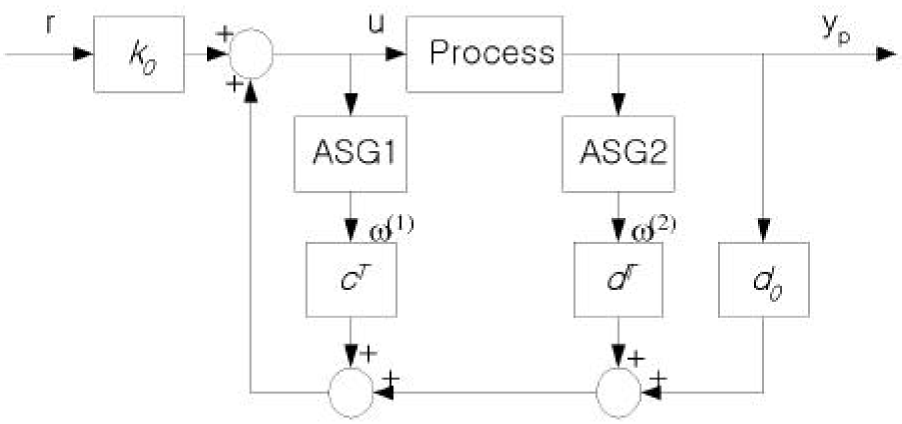


Fig. 4-2 Primary controller structure of augmented error method

$\omega^T = (r, \omega^{(1)T}, y_p, \omega^{(2)T})$: The signal vector

$\theta^T = (k_0, c^T, d_0, d^T)$: The controller parameter vector

θ^* : The correct parameter vector

$\phi = \theta - \theta^*$: The parameter error vector

4.2 Adaptive law의 유도

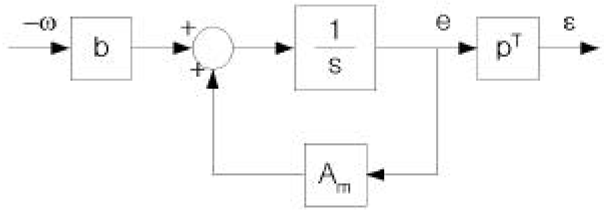
Adaptive law의 유도를 위한 Lyapunov 방법이나 Hyperstability 이론을 사용할 수 있으려면, 에러식의 선형부분이 SPR이어야 한다.

만약 에러벡터(e)를 사용할 수 있다면, SPR은 p (compensator vector)를 사용해 증명된다($\varepsilon = p^T e$ 사용). p는 $\frac{\varepsilon}{-\omega}$ 이 SPR(strict positive real)이 되게 에러전달 함수 내에 존재하는 영점의 위치를 바꾸거나 영점을 추가한다. 그러나 단지 $e_1 = y_p - y_m$ 만을 사용할 수 있다면, c(observation vector)가 고정되었기 때문에 이 계획은 더 이상 사용할 수 없다.

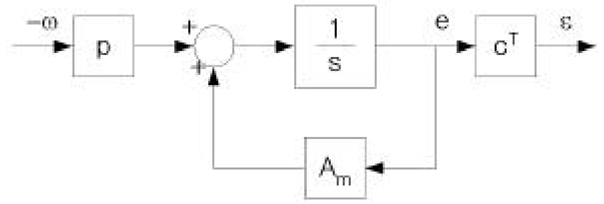
완전한 에러벡터(e)를 알 수 없으므로, 에러식의 입력을 바꾸어서 선형부분 SPR 만듦으로 이를 보상한다. 즉 Fig. 4-3(b)와 같이, 전달함수의 p를 입력벡터처럼 위치시킬 수 있다(K-Y lemma). p(보상벡터)의 위치 변동은 에러모델의 입력(이나 reference 모델입력)에 보조 신호(auxiliary signal)를 더하므로 가능해진다.

Adaptative law는 Lyapunov의 두 번째 방법을 사용해 유도할 것이며, Suzuki와 Dohimoto에 의해 설명된 hyperstability theory의 적용도 가능

할 것이다.



(a)



(b)

Fig. 4-3 Normal place(a) and alternative place(b) of compensator vector p

법칙유도에 있어서 첫 번째는 에러식의 유도이다. 이 방법에는 ASG가 폐루프에서 별도의 상태를 도입하기 때문에, 상태 피드백의 경우처럼 간단하지 않다. 폐루프 시스템은 모델보다 높은 차수이므로, 유도를 더 복잡하게 한다. Narendra, Valavani(1978)는 출력에러($e_1 = y_p - y_m$)를 아래와 같이 쓸 수 있음을 유도했다.

$$e_1 = \frac{k_p}{k_m} \omega_m (\phi^T \omega) \quad (4\text{-}1)$$

더나아가, k_p**의 부호를 알 수 있다고 가정하면** SPR**특성을 부과할 수 있다. 보편성의 결여가 없다면** k_p, k_m**은 동일하다고 가정한다.**

$$y_p = w_m (r + \phi^T w) \tag{4-2}$$

w_m**이** SPR**인 경우와** SPR**이 아닌 경우에 대해** adaptation law**가 각각 유도된다.**

<Case 1> : w_m , SPR(strictly positive real)**인 경우**

w_m**이** SPR**이면** e_1**의 보상기는 필요없고** e_1**은** adaptation law**에서 직접 사용될 수 있다.**

$$w_m = c_I^T (SI - A)^{-1} b_m \quad \left(c_I^T = (1,0,0,\cdots,0) \right) \tag{4-3}$$

Lyapunov function

$$V = e^T P e + \phi^T \Gamma^{-1} \phi, \quad \Gamma^T = \Gamma > 0, \; e = \begin{pmatrix} e_1 \\ \dot{e}_1 \\ \cdot \\ \cdot \\ e_1^{(n-1)} \end{pmatrix} \tag{4-4}$$

(미분) $\dot{V} = e^T (A_m P + P A_m) e + 2(\phi^T w) b_m^T P e + 2\phi^T \Gamma^{-1} \phi$ (4-5)

w_m**이** SPR**이기 때문에**(K - Y lemma**에 따라**) , **양의 한정 대칭행렬**

P, Q는 존재한다.

$$A_m^T P + PA_m = -Q$$

$$Pb_m = c_I \qquad c_I^T = (1,0,0,\cdots,0) \qquad (4\text{-}6)$$

$$\Rightarrow (Pb_m)^T = (c_I)^T \Rightarrow b_m^T P^T = c_I^T \Rightarrow b_m^T Pe = c_I^T e$$

$$\Rightarrow c_I^T = (1,0,0,\cdots,0) \Rightarrow e_1$$

$$\Rightarrow \dot{V} = -e^T Qe + 2(\phi^T w)e_1 + 2\phi^T \Gamma^{-1}\dot{\phi} \qquad (4\text{-}7)$$

$\dot{V}$이 음의 한 점이 되게 오른쪽의 두 표현을 0으로 하면 아래 식과 같다.

$$2(\phi^T w)e_1 = -2\phi^T \Gamma^{-1}\phi$$

$$\phi^T we_1 = -\phi^T \Gamma^{-1}\dot{\phi}$$

$$\dot{\theta} = \dot{\phi} = -\Gamma we_1 \leftarrow \text{adaptive law} \qquad (4\text{-}8)$$

<Case 2> : w_m이 SPR이 아닌 경우

w_m이 SPR이 아니면, 식(4-1)에 영점을 더할 필요가 있으며, 출력 에러 e_1에 별도의 신호 v를 더하여 달성한다.

$$\varepsilon = e_1 + v \qquad L^{-1}u$$

$$= e_1 + w_m L(\phi L^{-1} - L^{-1}\phi)^T w$$

$$= w_m(\phi^T w) + w_m L\phi^T L^{-1}w - w_m L L^{-1}\phi^T w$$

$$\varepsilon = w_m L(\phi^T L^{-1}w) \leftarrow \text{adaptation} \qquad (4\text{-}9)$$

결과적으로 증가된 에러신호(ε)가 adaptation에 사용되며 위 식에서 L은 $w_m L$이 SPR로 만들기 위해 선택된 S의 다항식이다. 여기서 L의 (n_L) 차수는 ($n-m-1 \le n_L \le n-m+1$)을 만족해야 한다. 다른 가능성은 L을 유리함수로 선택하는 것이다. 전달 1을 가진 에러식으로 되는 $L^{-1} = w_m$를 선택하는 것이 일반적이다. L은 $n_L = n-m-1$의 다항식으로 가정할 것이다. adaptive law는 출력에러(e_1)대신 증가된 에러 ε를 기초로해 유도되었다. ε에 대한 식은 에러 전달함수 내에 다항식 L을 포함하며, Lyapunov 함수는 $V = e^T P e + \phi^T \Gamma^{-1} \phi$와 같다.

여기서, e는 e_1대신 ε를 결합시킨 에러벡터이다.

$$e = \begin{pmatrix} \varepsilon \\ \varepsilon \\ . \\ . \\ \varepsilon^{n-1} \end{pmatrix}$$

식(4-9)에 따라 e에 대한 식은

$$e = A_m e + b_m' \phi^T (L^{-1} w) \qquad \varepsilon = (1, 0, \ldots, 0) e \qquad (4-10)$$

b_m'은 L이 새로운 에러모델(식 4-9)의 부분이기 때문에 b_m' 내에 포함된 L에 의해 나타나는 영점에 의해 b_m과는 다르다.

$$\dot{V} = e^T (A_m^T P + P A_m) e + 2 b_m'^T P e \phi^T (L^{-1} w) + 2 \phi^T \gamma^{-1} \dot{\phi} \qquad (4-11)$$

$w_m L$이 SPR이기 때문에 양의 한정 대칭 행렬은 아래 식을 만족해야

한다.

$$A_m^T P + PA_m = -Q$$

$$Pb_m^{'} = c_I \quad (4\text{-}12)$$

여기서, P가 대칭 행렬이기 때문에 $P = P^T$이다.

$$(Pb_m^{'})^T = c_I^T \Rightarrow b_m^{'T} P = c_I^T$$

$$\Rightarrow b_m^{'T} Pe = c_I^T e = \varepsilon$$

$$2\phi^T \Gamma^{-1} \phi = -2\varepsilon\phi^T L^{-1} w$$

$$\Rightarrow \dot{\theta} = \dot{\phi} = -\Gamma(L^{-1}w)\varepsilon \quad (4\text{-}13)$$

그러므로 보조신호를 더함으로써 ε에 대한 안정도 증명이 가능해진다. 그러나 보조신호 v는 파라미터 에러벡터 ϕ(물론 이용할 수 없다.)를 포함하므로 다음표현을 고려해 이 문제는 해결한다.

$$\phi L^{-1} - L^{-1}\phi$$

ϕ를 $\theta - \theta^*$로 간주되며, 이 표현은 다음과 같이 쓰여질 수 있다.

$$\phi L^{-1} - L^{-1}\phi = (\theta - \theta^*)L^{-1} - L^{-1}(\theta - \theta^*)$$

$$= \theta L^{-1} - L^{-1}\theta - (\theta^* L^{-1} - L^{-1}\theta^*)$$

(4-14)

θ^*가 상수이므로 괄호부분이 0이 된다. 그래서

$$\phi L^{-1} - L^{-1}\phi = \theta L^{-1} - L^{-1}\theta \quad (4\text{-}15)$$

그러므로

$$\varepsilon = e_1 + w_m L(\theta L^{-1} - L^{-1}\theta)^T w \quad (4\text{-}16)$$

이 식은 미분 연산자 없이 보조 신호가 수행될 수 있음을 의미한다.

$u = \theta^T w$의 표현을 다시 고려하면, 식(4-16)은 아래와 같이 표현된다.

$$\varepsilon = e_1 + w_m L(\theta^T L^{-1} - L^{-1}\theta^T)w$$

$$\varepsilon = e_1 + w_m L(\theta^T L^{-1} w - L^{-1} u) \qquad (4-17)$$

여기서 P는 안정한 adaptive law를 유도하기 위해 필요할 뿐, 실제 사용되지는 않는다.

4.3. 속도추정기 설계 및 전체시스템

벡터제어에서는 속도에 대한 정보가 필요하다. 센서리스 벡터제어구현을 위한 속도 즉, $\widehat{\omega_r}$(추정속도)를 추정하기 위한 제어기를 MRAC이론을 적용해 구성한다. 먼저 plant를 모델링하고, 원하는 입력지령 추적과 정상상태오차 제거의 목적으로 PI제어기를 설계하고 선형식으로 프로세스식을 유도한다. 여기에 다시 구조적인 변동분을 고려해 모델식을 세우고, 모델과 프로세스 식을 이용해 MRAC이론을 적용하여, 변동분을 보상하기 위한 primary controller를 구성한다. 이 primary controller가 프로세스의 출력이 안정한 영역으로 도달하도록 Lyapunov 안정도 이론을 바탕으로한 Adaptive Law를 구하고, 이를 적용한다.

유도전동기의 벡터제어시 회전자저항변동에 대한 보상이 이상적이라면 유도전동기의 응답특성은 직류전동기와 유사하게 되어 유도전동기의 토크발생은 직류전동기의 모델과 같이 단순화시킬 수 있다. 관성모멘트 J, 마찰계수 B, 전동기 토크 T_m, 부하토크 T_l인 유도전동기의 운동방

정식은 다음 식(4-20)과 같으며, Fig. 4-4와 같은 블록도로 나타낼 수 있다.

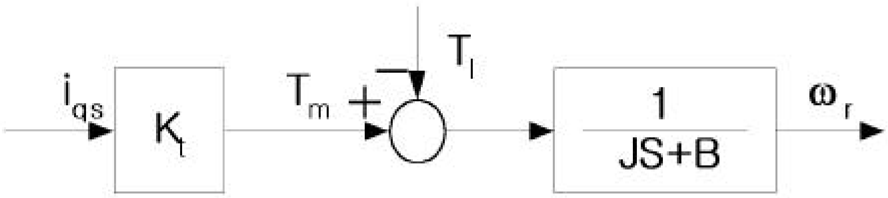


Fig. 4-4 Block diagram of plant

$$T_m - T_l = J\frac{d\omega_r}{dt} + B\omega_r \quad (4\text{-}19)$$

전동기 상수를 K_t 라 하면, 식(4-19)는 아래와 같이 다시 쓰여진다.

$$K_t i_{qs} - T_l = J\frac{d\omega_r}{dt} + B\omega_r \quad (4\text{-}20)$$

A. Derivation of the process

디자인 목적과 적응제어이론을 적용하기 위해, 외란에 자유롭다고 가정하여 프로세스를 선형으로 설계한다. 이렇게 해서 유도된 프로세스식은 아래와 같고 블록도는 Fig. 4-5와 같이 나타낼 수 있다. 프로세스식 내에서 사용되는 수식은 모델과 구분을 위해 아래첨자 p를 사용해 다시 표현하였다.

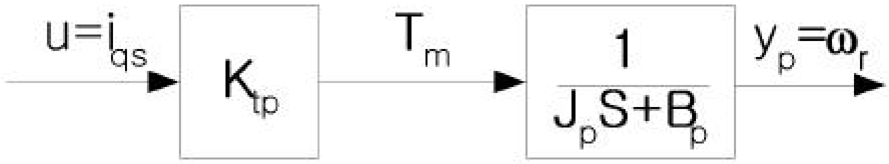


Fig. 4-5 Block diagram of process

프로세스식의 ω_r은 추정속도에 해당하므로 $\widehat{\omega_r}$와 같이 표현하고, i_{qs}에 대한 출력을 y_p라 하자.

입력($u = i_{qs}$)에 대한 출력 y_p의 전달함수를 구하면,

$$\frac{y_p}{u} = \frac{\widehat{\omega_r}}{i_{qs}} = \frac{K_{tp}}{J_pS + B_p}$$

$$y_p = \frac{K_{tp}}{J_pS + B_p} u \qquad (4\text{-}21)$$

식(4-21)을 시뮬레이션을 위한 상태방정식으로 유도하기 위해,

$\frac{u}{J_pS + B_p} = x_1$라 하자.

$$J_pSx_1 + B_px_1 = u$$

$$Sx_1 = -\frac{B_p}{J_p}x_1 + \frac{1}{J_p}u$$

여기서 $\frac{B_p}{J_p} = a_p$, $\frac{1}{J_p} = b_p$, $Sx_1 = \dot{x}_1$ 라 하면

$$\dot{x}_1 = -a_px_1 + b_pu$$

그러므로, $y_p = K_{tp}x_1$이다.

B. Derivation of the model

유도전동기는 벡터제어법으로 선형 제어될 수 있다. 그러나 실제의 경우, 회전자시정수오차가 존재하며, 이것은 비선형 특성의 토오크 응답을 일으켜 토오크상수의 구조적 변동분을 발생시키고, 전동기에 연결되는 기계적 부하는 운전중 변화되며 부하관성의 변화는 전달함수의 계수에 큰 영향을 미쳐 선형 시불변시스템으로 표현이 불가능해진다. 벡터제어 유도전동기의 전달함수는 변동분이 존재하는 비선형시변 전달함수로 표현되어야한다. Fig. 4-6은 모델의 블록도이고, J , K_t , B 를 구조적 변동분이 존재하는 J_m , K_{tm} , B_m 식으로 바꾸어 모델식을 유도하면 다음과 같다.

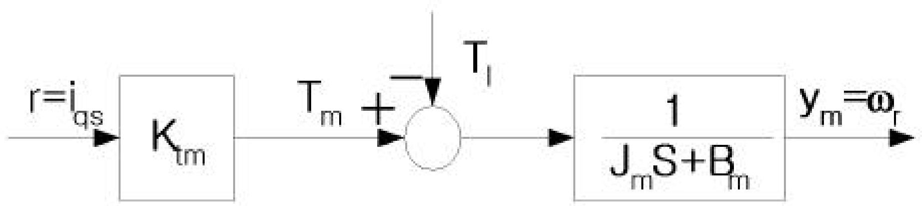


Fig. 4-6 Block diagram of model

입력에 대한 출력의 전달함수로 나타내면

i_{qs}에 대한 출력을 y_{m1}으로 두고, T_l에 대한 출력을 y_{m2}라 하면, $y_p = y_{m1} - y_{m2}$ 이다.

먼저 입력($r = i_{qs}$)에 대한 출력 y_{m1}의 전달함수를 구하면,

$$\frac{y_{m1}}{r} = \frac{\omega_r}{i_{qs}} = \frac{K_{tm}}{J_m S + B_m}$$

$$y_{m1} = \frac{K_{tm}}{J_m S + B_m} r \qquad (4-22)$$

식(4-22)를 시뮬레이션을 위한 상태방정식으로 유도하기 위해,

$\frac{r}{J_m S + B_m} = x_2$ 라 하자.

$$J_m S x_2 + B_m x_2 = r$$

$$S x_2 = -\frac{B_m}{J_m} x_2 + \frac{1}{J_m} r$$

여기서 $\frac{B_m}{J_m} = a_m$, $\frac{1}{J_m} = B_m$, $S x_2 = \dot{x}_2$ 라 하면

$$\dot{x}_2 = -a_m x_2 + b_m r$$

그러므로, $y_{m1} = K_{tm} x_2$ 이다.

입력(T_l)에 대한 출력 y_{m2}의 전달함수를 구하면,

$$\frac{y_{m2}}{T_l} = \frac{\omega_r}{T_l} = \frac{1}{J_m S + B_m}$$

$$y_{m2} = \frac{1}{J_m S + B_m} T_l \qquad (4-23)$$

식(4-23)을 시뮬레이션을 위한 상태방정식으로 유도하기 위해,

$\frac{T_l}{J_m S + B_m} = x_3$ 라 하자.

$$J_m S x_3 + B_m x_3 = T_l$$

$$S x_3 = -\frac{B_m}{J_m} x_3 + \frac{1}{J_m} T_l$$

여기서 $\frac{B_m}{J_m} = a_m$, $\frac{1}{J_m} = b_m$, $Sx_3 = \dot{x}_3$ 라 하면,

$\dot{x}_3 = - a_m x_3 + b_m T_l$

그러므로, $y_{m2} = x_3$ 이다.

결론적인 출력 $y_m = y_{m1} - y_{m2}$ 는 아래와 같다.

$$y_m = K_{tm} x_2 - x_3 \qquad (4\text{-}24)$$

여기서, 전동기 상수 K_{tm} 는 간접벡터시 전류와 자속제어가 완전하게 이루어진다고 가정하면, 전동기 토크 T_m 이 다음과 같으므로, 그 값을 구할 수 있다.

$$T_m = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\lambda_{dr} i_{qs}$$

$T_m = K_t i_{qs}$ 이므로, K_{tm} 는 $\frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\lambda_{dr}$ 이다.

C. Primary controller design

실제 모델의 변동분을 보상하는 제어기를 설계한다. $u = \phi^T w$. 여기서 우리가 이용할 수 있는 양은 입력과 출력이므로, 완전한 프로세스 상태벡터를 알 수 없다. 이에 따라 우리가 실제 사용할 수 있는 출력에러 ($e = y_p - y_m$)에 대한 보상이 필요하게 된다. 그러므로, Monopoli가 제시한 augmented error method에 의해 시그널을 확장한다.

$w^T = (r, w_1, y_p, w_2) \Rightarrow$ signal vector

$\theta = \phi^T = (k_0, c^T, d_0, d^T)$

$\Rightarrow$ controller parameter vector

$\theta^* \Rightarrow$ correct parameter vector

$\phi = \theta - \theta^* \Rightarrow$ parameter error vector

확장된 시그널과 파라미터에 대한 제어기를 구성하면 아래와 같이 된다.

$$\therefore u = k_0 r + c w_1 + d_0 y_p + d w_2 \qquad (4\text{-}25)$$

D. Derivation of the adaptive law using Lyapunov's stability theory[26][27][28]

MRAC의 비선형, 시변특성 때문에, 선형안정도분석은 적용할 수 없다. 적응시스템의 전체적인 안정도를 증명하기 위해 잘 알려진 방법인 Lyapunov's direct method(Parks,1966)을 사용한다.

Lyapunov's method는 만약 Lyapunov함수 $V(x)$가 아래조건을 만족한다면, 시스템이 일양점근안정평형 $x = 0$을 가짐을 설명한다.

$V(x) > 0$ for $x \neq 0$ (positive definite)

$\dot{V}(x) < 0$ for $x \neq 0$ (negative definite)

Lyapunov 함수는 $V = e^T P e + \phi^T \Gamma^{-1} \phi$와 같고, 도함수 $\dot{V}(x)$를 구하면 식(4-26)과 같다.

$$\begin{aligned} \dot{V} = \; & e^T (A_m^T P + P A_m) e \\ & + 2 b_m^{'T} P e \phi^T (L^{-1} w) + 2 \phi^T \gamma^{-1} \dot{\phi} \end{aligned} \qquad (4\text{-}26)$$

식(4-26)에서 오른쪽의 두식을 영으로 놓으면 식(4-27)과 같은 Adaptive law를 구할 수 있다.

$$\dot{\theta} = \dot{\phi} = -\Gamma(L^{-1}w)\varepsilon \tag{4-27}$$

Lyapunov's stability theory을 적용해 구한 adaptive law를 이용해 primary controller내의 파라미터 값들을 변경한다.

$$\dot{\theta} = \dot{\phi} = \begin{pmatrix} \dot{k}_0 \\ \dot{c} \\ \dot{d}_0 \\ \dot{d} \end{pmatrix}$$

$$= -\begin{pmatrix} \gamma_0 & 0 & 0 & 0 \\ 0 & \gamma_1 & 0 & 0 \\ 0 & 0 & \gamma_2 & 0 \\ 0 & 0 & 0 & \gamma_3 \end{pmatrix} \begin{pmatrix} L^{-1}r \\ L^{-1}w_1 \\ L^{-1}y_p \\ L^{-1}w_2 \end{pmatrix} \varepsilon$$

$$\therefore \dot{k}_0 = -\gamma_0 L^{-1} r\varepsilon$$

$$\dot{c} = -\gamma_1 L^{-1} w_1 \varepsilon$$

$$\dot{d}_0 = -\gamma_2 L^{-1} y_p \varepsilon$$

$$\dot{d} = -\gamma_3 L^{-1} w_2 \varepsilon \tag{4-28}$$

E. Sensorless vector control

속도 추정치를 벡터제어계로 피드백 시켜 센서리스 벡터제어시스템을 구현한다. 원하는 입력지령추적과 정상상태오차제거의 목적으로 PI제어기를 설계한다. PI제어 방식에 있어 무엇보다 중요한 것은 각항의 계수의 선택이다. 이를 위해 잘 알려진 계수 튜닝방법으로 스텝응답법과 한계감도법(Ziegler Nichols)있다. 여기서는 한계감도법을 사용해서 값을 선정한 뒤 계속적인 튜닝으로 최적의 계수를 찾아내기 위해 미세조정을 행하였다.

한계감도법의 특징은 비례동작만으로 값을 구한다는 것이다. 구하는 방법으로는 제어루프가 한계진동이 되도록 비례계수를 늘려간다. 그때의 최대값과 진동주파수 ω_m 을 계측해서 Ziegler Nichols가 만든 산술표를 적용해 구한다. 그 값은 아래와 같다.

$$K_p = 0.6K_{pm}$$
$$K_i = K_p \frac{\omega_m}{\pi}$$
$$K_d = \frac{K_p \pi}{4} \omega_m$$

MRAC를 이용한 속도제어계는 아래 그림과 같다.

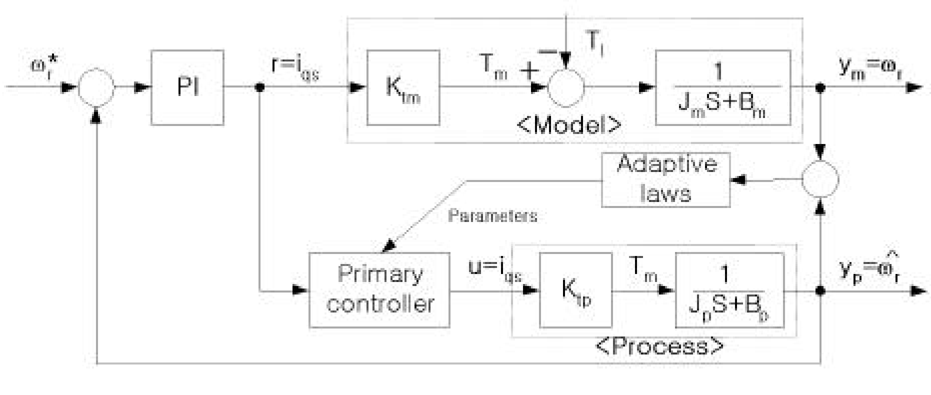


Fig. 4-7 Speed control system using MRAC

본 논문에서 제안한 MRAC를 이용한 속도추정기를 부가시킨 유도전동기의 간접벡터구동 시스템의 블록도를 Fig. 4-8에 나타내었다. 추정속도를 계산하기 위한 MRAC부분, 전류제어, 비간섭제어부분으로 구성되

어 있다.

기준속도와 추정속도를 비교한 속도오차는 PI제어기를 통하여 고정자에 인가할 토크발생을 위한 q축 전류를 발생시키며, 이 전류를 다시 MRAC를 이용한 속도추정기를 통해서 추정속도를 연산하여 속도제어계로 피드백 된다. 그리고 d축, q축 지령전류는 전동기의 고정자에서 측정한 전류 ids, iqs와 비교하여 고정자 dq축 지령전류를 얻고 이에 다시 비간섭제어를 통해 상호간섭분을 제거하여 최종 고정자 dq축 지령전류를 얻는다. 이를 다시 상변환(2상/2상)을 통해 고정좌표계에 대한 지령치를 얻고, 2상/3상 변환하여 공간전압벡터스위칭을 결정할 고정자3상전압을 얻는다.

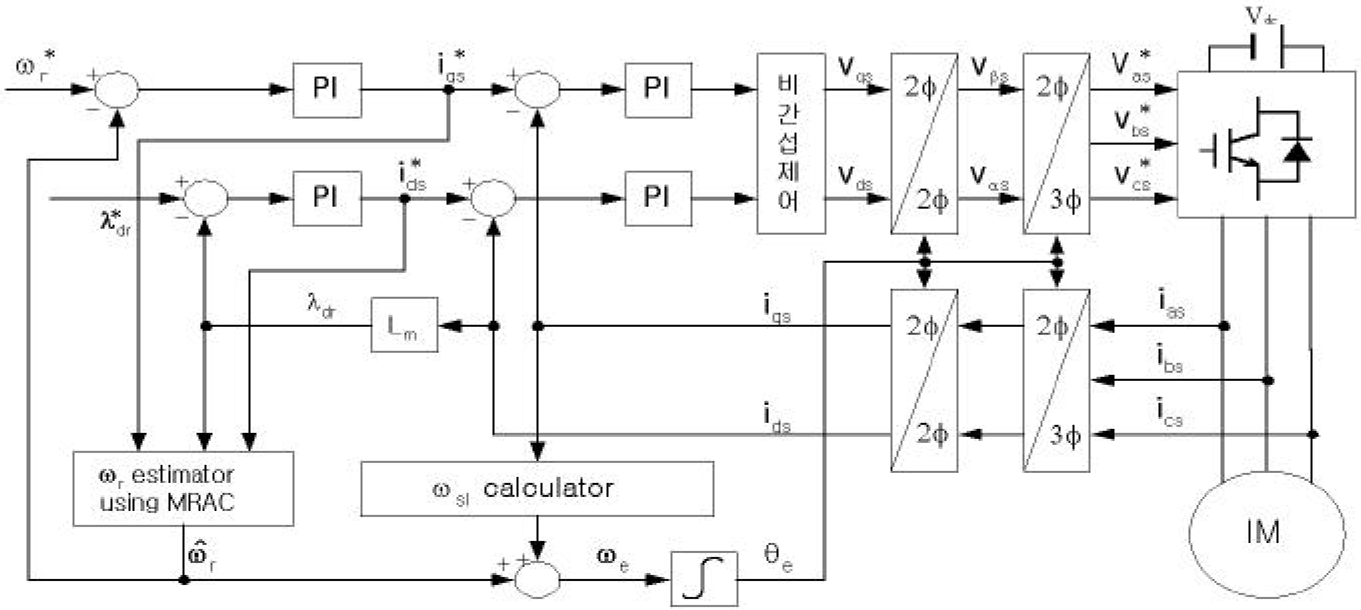


Fig. 4-8 Schematic diagram of indirect vector control system

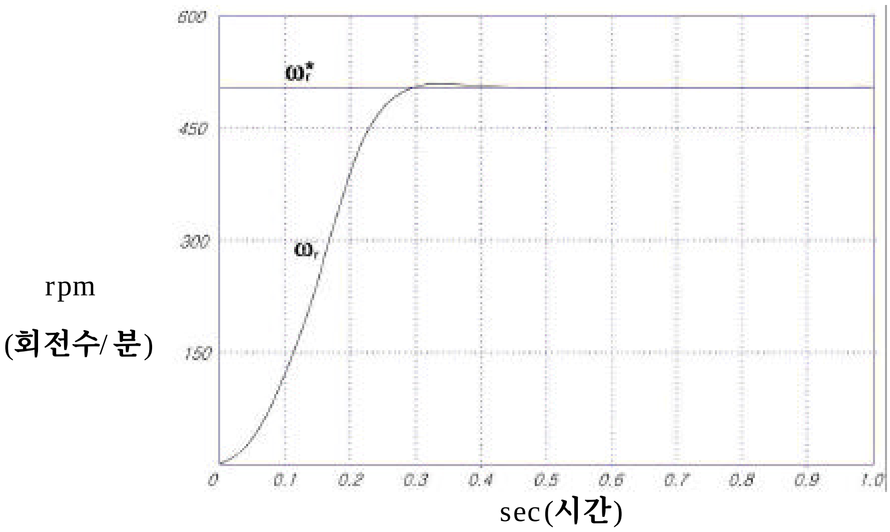
5. 시뮬레이션 및 시스템 구현

5.1 시뮬레이션

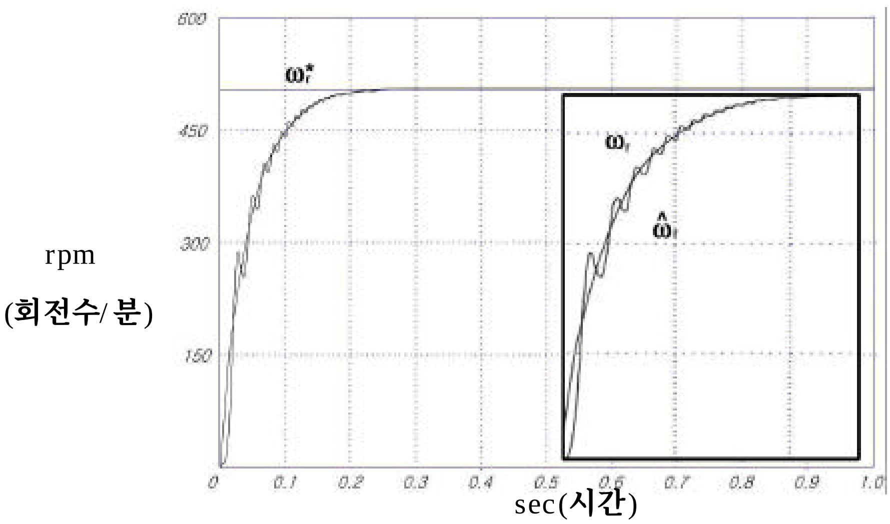
본 장에서는 제안한 방법의 타당성을 검증하기 위해 시뮬레이션을 수행하였다. 시뮬레이션은 Turbo C 언어를 사용하여 IBM PC 상에서 행하였고, 제어기내의 속도제어주기는 2 [ms], 전류제어주기는 200 [μs]로 하여, PI제어기만을 사용하는 벡터제어와 MRAC를 적용한 속도추정기를 사용하는 벡터제어의 특성을 비교 분석하였다. 다음 Table 5.1은 시뮬레이션 및 실험에 사용된 유도전동기의 정격과 파라미터 값이다.

Table 5.1 Parameters and ratings of sample induction motor

Parameters & ratings		values
Rated output		2.2 [kw] (3HP)
Rated voltage		220 [V]
Rated load current		8.6 [A]
Rated speed		1740 [rpm]
Pole	P	4
Stator resistance	R_1	0.9210 [Ω]
Rotor resistance	R_2	0.5830 [Ω]
Stator inductance	L_1	67.1 [mH]
Rotor inductance	L_2	67.1 [mH]
Mutal inductance	M	65.0 [mH]
Moment of inertia	J	0.0418 [kg・m]

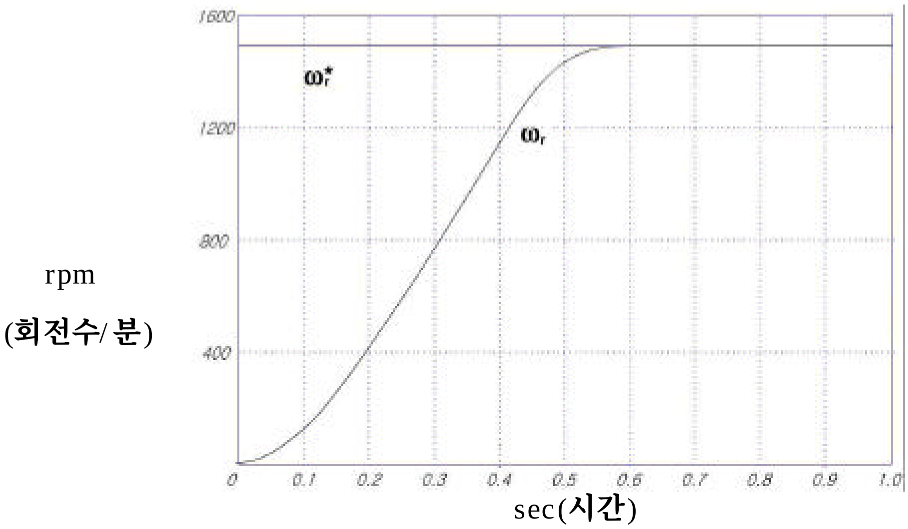


(a) PI Control

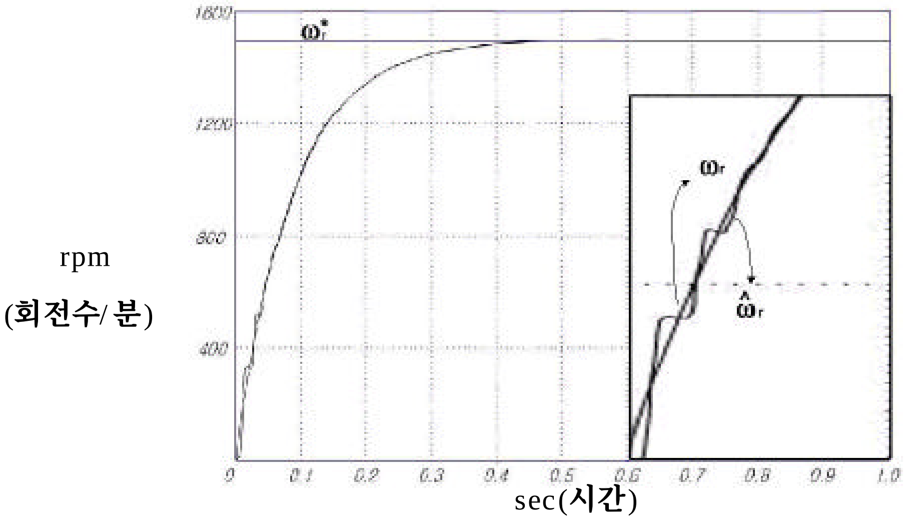


(b) MRAC

Fig. 5-1 Reference speed(ω_r^*) 500[rpm]



(a) PI Control



(b) MRAC

Fig. 5-2 Reference speed(ω_r^*) 1500[rpm]

Fig. 5-1은 지령속도 500[rpm]을 주었을 때 속도파형을 시뮬레이션한 파형으로 (a)는 PI제어만을 행했을 때의 파형이며, (b)는 본 논문에서 제안한 MRAC속도 추정기를 사용했을 때의 파형이다. 그림에서 보면 알 수 있듯이 PI제어만을 행했을 때는 정상상태 도달까지 300[ms]정도의 시간이 걸린다. 이에 비해 MRAC를 수행했을 때는 약 250[ms]정도로 저속에서 PI제어보다 빠른 제어가 수행됨을 알 수 있다. Fig. 5-2는 속도 1500[rpm]을 지령치로 주었을 때의 파형이다. (a)의 경우는 550[ms]이며 (b)의 경우는 450[ms]로 역시 빠른 제어 성능을 확인할 수 있었다.

5.2 시스템 구현

본 논문에서 제안한 알고리즘을 적용한 시스템을 구현하였다. 하드웨어부분은 마이크로프로세서, SVPWM 인버터 그리고 이 두 부분을 연결하는 인터페이스회로를 주축으로 프로그림의 다운로딩을 위한 퍼스널컴퓨터(PC), 신호 입・출력부 및 전원장치로 구성되어져 있다. 제어프로그램의 모든 소프트웨어는 속도를 고려하고 하드웨어의 제어가 용이하도록 TMS320C31의 어셈블리언어를 사용하여 작성하였으며 속도제어주기는 2[ms]이고, 전류제어주기는 200[μs]로 설정하였다. 하드웨어 사진은 Fig. 5-3에 나타내었다.

마이크로프로세서

제어회로의 핵심으로 모든 수학적 연산처리가 행해지고 제어시스템 전체의 제어가 이곳으로부터 이루지게 되므로 고속의 부동소수점연산이 가능한 Texas Instrument사의 32비트 DSP인 TMS320C31을 사용하였다.

유도전동기의 벡터제어실현을 위해서는 많은 고속의 연산을 실시간으로 할 필요가 있고, 연산의 정밀도가 제어기의 성능을 좌우하게 된다.

인터페이스

외부로부터 센싱된 값이나 지령치를 CPU로 전달하거나 현재의 제어 상태를 인터페이스를 통해 외부에 표시한다. 가능한 모든 기능을 디지털 소자인 Altera사의 EPLD(Erasable Programmable Logic Device)내에 집적시켰으며, 정밀측정을 요하는 전류센싱이나 직류전압 검출부에 사용되는 저항은 1%정밀저항을 사용하였다.

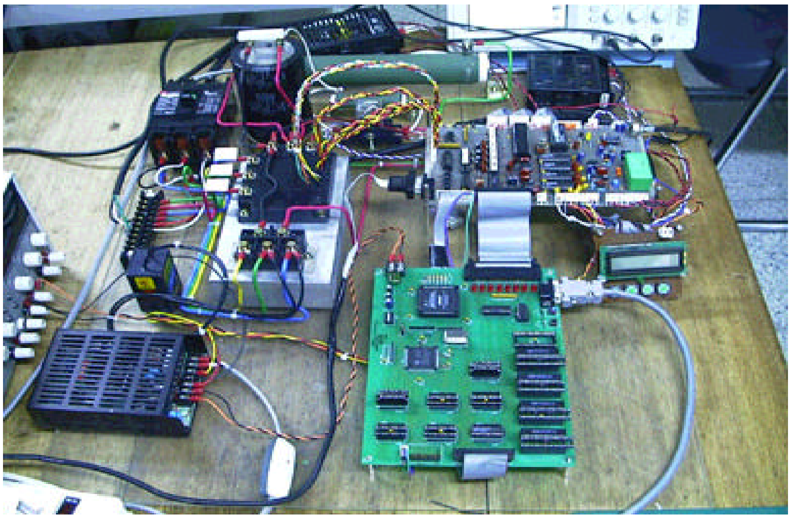


Fig. 5-3 Photograph of hardware

CPU는 리셋회로, 클럭발생회로와 프로그램 다운로딩을 위한 PC가 연결되어 있으며, 이곳으로부터는 어드레스신호, 데이터신호, 제어신호 및 EPLD신호의 입·출력이 행해지고 있다. DC 링크단의 전압, 전동기 상전

류 및 외부 속도지령의 아날로그신호는 A/D 컨버터(MAXIM사 MAX122B)를 통하여 EPLD로 전달된다. 또한 SVPWM 인버터 스위칭을 위한 게이팅 신호도 EPLD내에서 시간으로 정량화 되어 TLP550의 포토커플러를 통해 IPM으로 전달된다.

전력변환부

실제적인 유도전동기의 가변속구동이 가능하게 되는 부분으로 크게 컨버터, 인버터 그리고 DC 링크단으로 구성하였으며, 그 개략도를 나타내면 Fig. 5-4와 같다. 컨버터와 인버터부분은 부피 및 내 노이즈성을 고려하여 컴팩트한 모듈을 사용하였으며, 컨버터에 사용된 브릿지 다이오드 모듈은 SanRex사의 800[V], 40[A]용량의 DF40BA80을 사용하였다. 3상 인버터용 IGBT(Insulated Gate Bipolar Transistor) 모듈은 자체적으로 게이트 드라이브 회로와 보호회로, 그리고 발전제동용 IGBT까지 내장하고 있는 IPM(Intelligent Power Module)으로 Mitsubishi 사의 600[V], 75[A] 용량의 PM75RSA060을 사용하여 매우 간단한 구조로 하였다. 컨버터용 다이오드 모듈에서는 단순히 교류를 직류로 바꾸는 역할을 하므로 DC링크단의 전압 평활을 위해서 평활용 콘덴서를 사용하였다. 평활용 콘덴서는 450[WV], 2400[μF], 85℃의 전해콘덴서를 사용하였다. 이와 같이, 정전용량이 큰 콘덴서를 사용할 경우에는 전원투입시 큰 돌입전류가 흐르게 되므로 다이오드 모듈이 파괴되고 만다. 따라서, 컨버터와 평활용 콘덴서 사이에 30[W], 10[Ω]의 저항을 직렬로 삽입하여 전원투입시 이 저항을 통해서 콘덴서에 충전되는 전류를 제한하였다. 콘덴서 충전이 완료되면 더 이상의 전류제한은 필요없으며, 바이패스를 위해서 충분한 용량의 릴레이를 사용하였다. 전원이 투입되고 직렬저항을 통하여 콘덴서가 충전되고 나면 릴레이가 ON되고 운전 중에는 OFF되지

않도록 EPLD내에서 처리를 하였다.

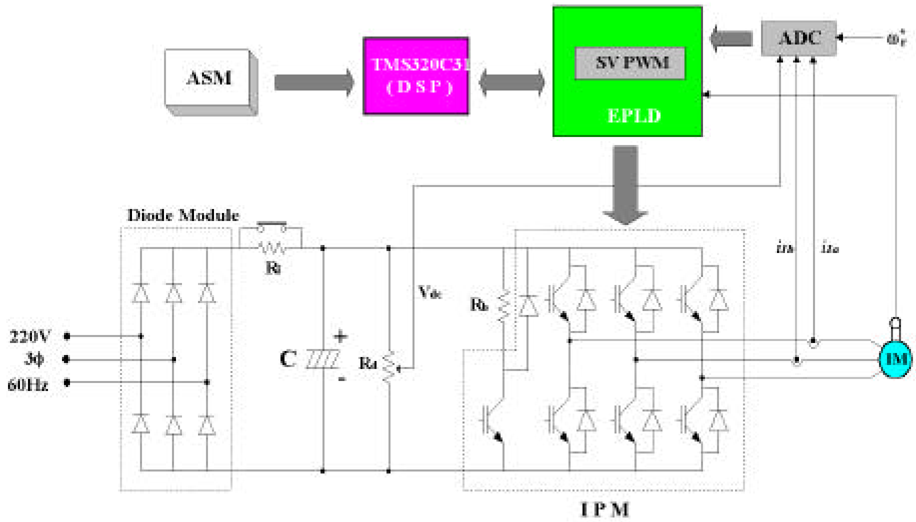


Fig. 5-4 Schematic diagram of IM motor servo system

프로그램

프로그램의 길이 및 실행속도를 고려하여 본 시스템에서는 모든 제어프로그램을 TI사의 TMS320C31 어셈블리언어를 사용하여 작성하였다. 프로그램의 개발과정중 수정을 고려하여 퍼스널 컴퓨터에서 프로그램을 작성하고 시리얼 부팅으로 CPU내의 내부 RAM 이나 CPU외부에 부착되어있는 외부 RAM으로부터 프로그램을 실행하도록 하였다. Fig. 5-5는 주프로그램의 흐름도를 나타내고 있다.

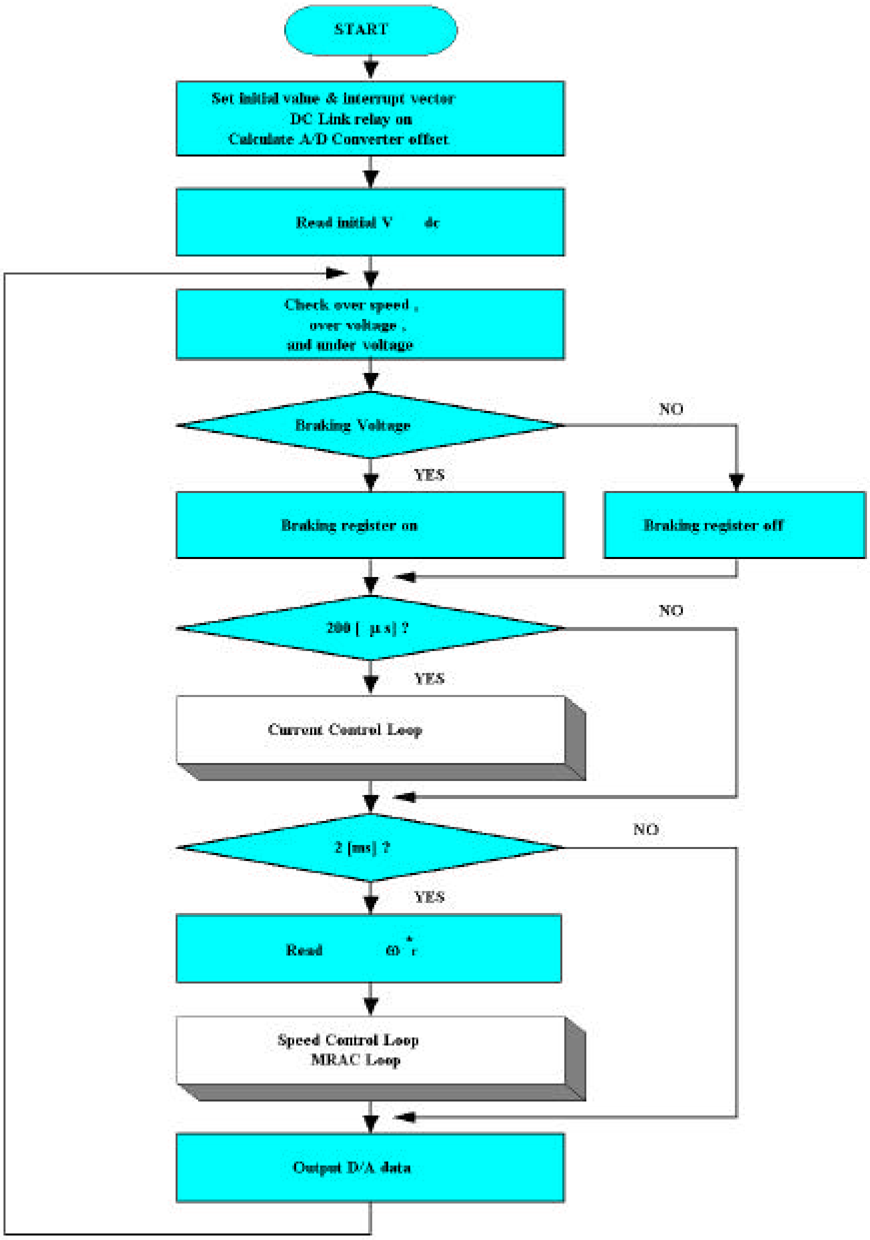
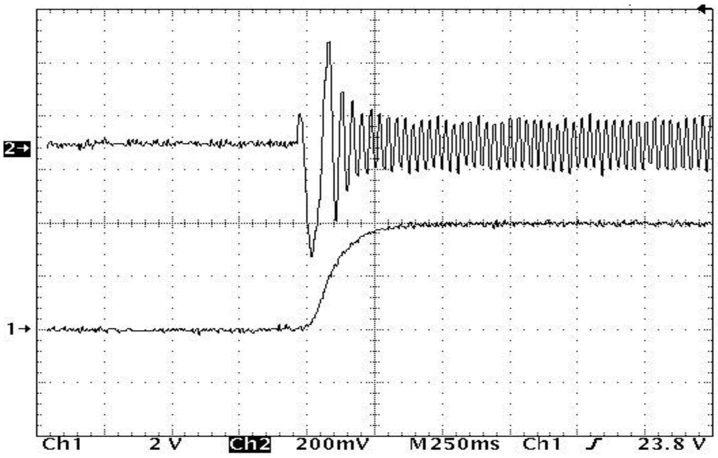
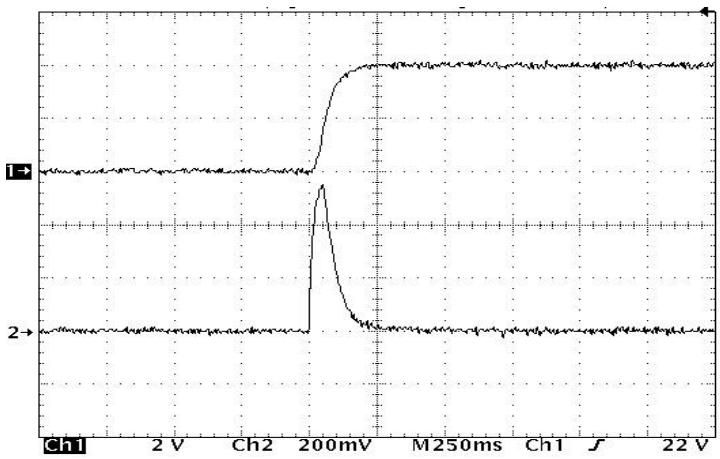


Fig. 5-5 Flow chart of main loop



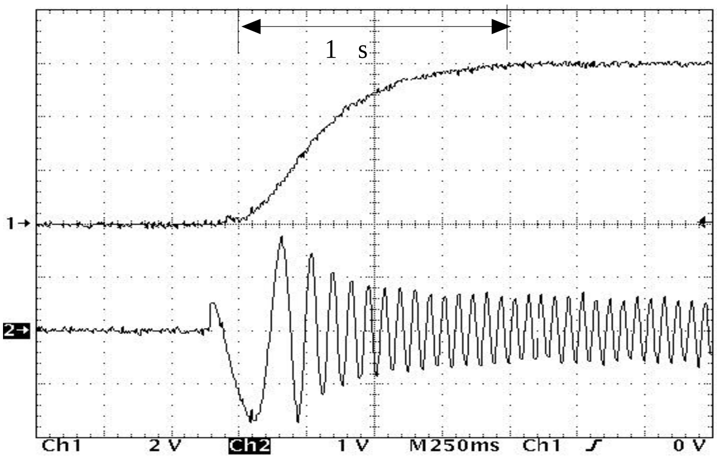
(a) PI Control (채널1: 속도, 채널2: 상전류)



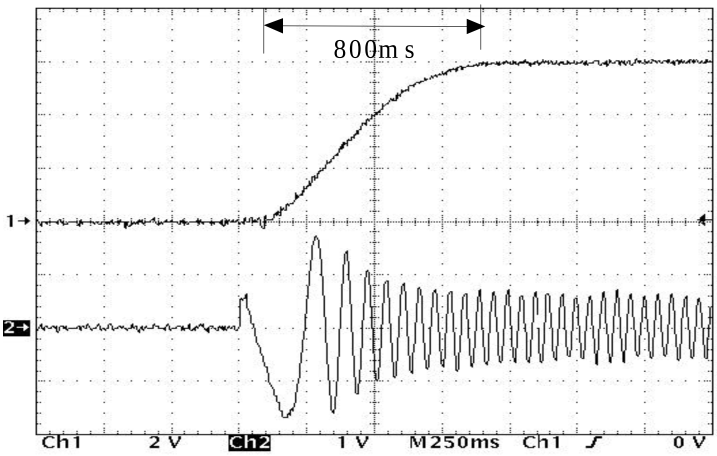
(b) MRAC (채널1: 속도, 채널2: q축전류)

Fig. 5-6 Reference speed(ω_r^*) 500[rpm]

(250rpm/div, 250ms/div)



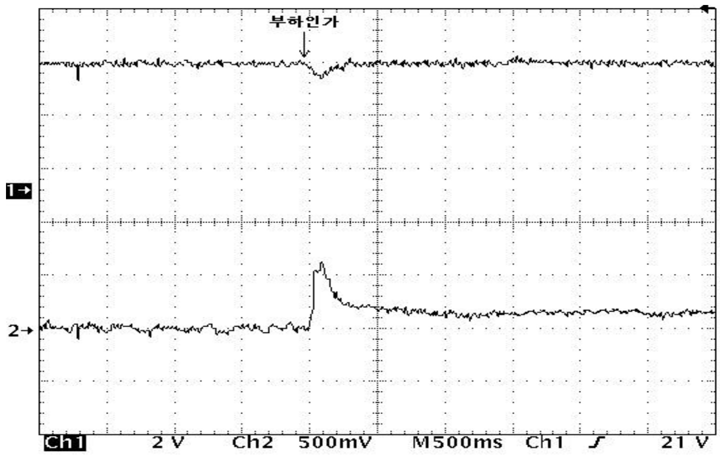
(a) PI Control (채널1: 속도, 채널2: 상전류)



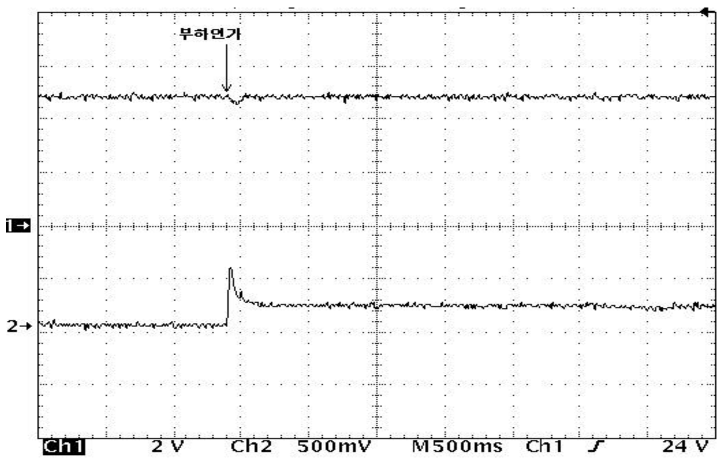
(b) MRAC (채널1: 속도, 채널2: 상전류)

Fig. 5-7 Reference speed(ω_r^*) 1500[rpm]

(250rpm/div, 250ms/div)



(a) PI Control (채널1: 속도, 채널2: q축전류)



(b) MRAC (채널1: 속도, 채널2: q축전류)

Fig. 5-8 load disturbance

(250rpm/div, 250ms/div,0-> 50%)

Fig. 5-6에서 Fig. 5-8은 실험을 통한 파형을 보인다. 시뮬레이션 결과와 비교하기 위해 지령속도 500[rpm] 및 1500[rpm]을 주어 파형을 관찰하였으며, 외란에 대한 강인성을 확인하기 위해 부하토크를 인가해보았다. 그림에서 (a)는 역시 PI제어만을 수행했을 때의 파형이며, (b)는 MRAC를 적용한 파형이다. 그림에서 수직축은 속도를 나타나며, 250rpm/div이고, 수평축은 시간축이며 250ms/div이다.

Fig. 5-6을 보면, 무부하시의 속도응답특성에 대한 파형으로서 PI제어만을 수행했을 때는 지령속도까지 도달시간이 약 400[ms]정도의 시간이 걸리고, MRAC수행 시는 약 250[ms]정도의 시간이 소요되므로 시뮬레이션의 결과와는 다소 차이가 있으나 MRAC의 성능이 우수함을 확인할 수 있다.

Fig. 5-7은 1500[rpm]의 지령속도를 주었을 때의 파형이다. (a)는 1[s], (b)는 800[ms]정도로 MRAC는 고속영역에서도 빠른 제어 성능을 보였다.

Fig. 5-8은 부하토크 인가시 속도 파형을 관찰해 보았다. PI제어시는 부하인가시 속도회복시간이 약 300ms이고, MRAC제어시는 약 120ms로 MRAC제어시의 외란억제특성이 매우 우수함을 알 수 있다.

6. 결 론

본 논문에서는 센서리스 벡터제어 구현을 위해 토크식을 사용해 속도를 추정하였으며, 단순 PI제어시의 문제점을 개선 및 보다 빠른 응답특성을 얻기 위해 MRAC를 적용한 속도추정기를 설계하여 제어 특성 및 응답특성을 향상시키고자 하였다. 제안한 방법을 적용해 시뮬레이션과 실험을 행하여 다음과 같은 결론을 얻을 수 있었다.

1. 기존의 속도센서의 사용에 따른 문제점을 속도센서를 사용하지 않고 계산을 통해 속도를 계산함으로써 속도센서를 통해 첨가될 수 있는 잡음에 대한 문제점제거 및 비용부담을 줄일 수 있었다.

2. 단순 PI제어의 문제점을 극복하기 위해, MRAC를 적용하여 보다 빠른 응답특성과 부하외란등에 강인한 특성을 확인할 수 있었으며, PI제어가 가지는 문제점을 보상할 수 있었다.

3. MRAC알고리즘 및 전류, 속도 제어등을 모두 소프트웨어로 처리하였고, 소프트웨어 보호기법과 하드웨어 보호장치등을 사용하여 시스템의 신뢰성을 향상시킬 수 있었으며, 디지털 회로의 EPLD내에 집적을 통해 부피의 소형화를 꾀할 수 있었다.

4. 시뮬레이션과 실험시 저속영역에서는 속도편차가 없으나, 고속영역에서는 속도편차가 약 200ms가 발생한 것은 시뮬레이션시 저항변동분의 보상이 이상적이라고 가정했기때문으로 분석되며, MRAC의 제어 성능이 우수하다고 사료된다.

본 논문에서, 변동분에 대한 보상이 이상적이라고 가정한 저항을 고려하여 보다 정확한 제어알고리즘에 대한 연구가 진행되어야 할것으로 사료된다.

참고 문헌

[1] **新中新二, "誘導形　サボモータの適応ベクトル制御"**, *IEE Japan*, Vol. 117-D. No. 8, pp. 1024-1032, 1997.

[2] **新中新二, 榊原則夫, 深澤英樹, "誘導機ベクトル制御のための統一的ベクトル　解析"**, *IEE Japan*, Vol. 117-D. No. 8, pp. 1024-1032, 1993.

[3] K Ohishi et al, "Robust Control of a DC Servo Motor Based on Linear Adaptive System", *JIEE Trans.*, Vol. 108-D. No. 1, pp. 39-45, 1988.

[4] Phyama.k, Asher.G.M. and Sumner.M., "Comparative experimental assessment for high-performance sesorless induction motor drivers", *IEEE IE*, Vol. 1, pp. 384-391, 1999.

[5] Djemai. M, boukhobza. T, Barbot. J. P, Thomas. J. l and Poullain. S, "Rotor speed and flux nonlinear observer for speed sensorless induction motors", *IEEE International conference*, Vol. 2, pp. 848-857, 1998.

[6] Faa-jeng Jeng Lin, Rong-Jong Wai and Pao_chuan Lin, "Robust speed sensorless induction motor drive", *IEEE AES*, Vol. 35, pp. 566-578, April 1999.

[7] Colin Schauder, "Adaptive speed identification for vector Control of Induction motors without rotational Transducers", *IEEE*, pp. 1054-1061, 1992.

[8] B. D. Youn et al, "Robust Speed Control of Induction Motor Using Sliding Mode Torque Observer", *IPEC Proc.*, pp. 87-92, 1995.

[9] Sangwonwanich. S, Suwankawin. S, “A speed-sensorless IM drive with modified decoupling control”, *PCC-nagaoka 1997*, Vol. 1, pp. 85-90, 1997.

[10] Hurst. K. D, Habetler. T. G, “A simple, Tacho-less, IM drive with diret torque control down to zero speed”, *IECON 97*, Vol. 2, pp. 563-568, 1997.

[11] Ilas. C, Griva. G, Profumo. F, “Wide range speed sensorless induction motor drives with rotor resistance adptation”, *PE, Drives and energy Systems for Industrial Growth*, Vol. 1, pp. 211-215, 1995.

[12] Kyung-Seo Kim, "**속도센서 없는 유도 전동기 자속 기준제어를 위한 새로운 자속 연산기**“, ***전력전자학회***, Vol. 31, pp. 125-130, 1998.

[13] Young-Soo Seo, Young_bae Lim, young-Chun Kim, Dae_Young Seong, “**순시 속도 관측기를 이용한 유도 전동기의 극저속운전**”, *KIEE*, 47-2-6, 1998.

[14] Andrzej M. Trzynadlowski, University of Nevada. Reno, "The Field Orientation Principle in Control of Induction motors", Kluwer Academic Publishers, 1994

[15] S. Sastry, M. Bodon, Adaptive Control: Stability, Convergence, and Robustness, Prentice-Hall, 1989.

[16] Attaianese. C, Fusco. G, Marongiu. I, Perfetto. A, “Parameter sensitivity of speed estimation in speed sensorless induction motor driver”, *AMC'96-MIE*, Vol 1, pp. 162-167, 1996.

[17] Blasco-Gimenez. R, Asher. G. M, Sumner. M, Bradley. K. j, “Dynamic performance limitations for MRAS based sensorless

induction motor drivers.1. Stablity analysis for the closed loop drive", *IEE EPA*, Vol. 143 2 pp. 113-122, march 1996.

[18] Shinaka. S, "Servo-performance hybrid vector contril for sensorless induction motor drive", *IEEE ISIE'99*, Vol. 1, pp. 380-385, 1999.

[19] Buja.G, Menis.R, "Accuracy of the speed estimation in the sensorless induction motor drives based on the MRAS technique", *OPTIM '98*, Vol. 2, pp. 407-414, 1998.

[20] Suwankawin. S, Sangwongwanich. S, "Stability analysis and design guidelines for a speed-sensorless induction motor drive", *PCC-pagaoka 1997*, Vol. 2, pp. 549-552, 1997.

[21] Phaderm Nangsue, Pragasen pillay, Susan E.Conry, "Evolutionary Algorithms for induction motor Parameter Determination", *TEC IEEE*, Vol. 14, No. 3, pp. 447-453, September 1999.

[22] Kanokvate Tungpimolrut, Fang_Zheng peng, Tadasi Fukao, "Pobust Vector Control of Induction Motor without Using Stator and Rotor Circuit Time Constants", *IEEE*, 1994.

[23] Tsugutoshi Ohtani, Member, Noriyuki Takada, Koji Tanaka, "Vector Control of Induction Motor without Shaft Encoder", *IEEE*, 1992.

[24] Fang-Zheng Peng, Tadashi Fukao, "Pobust speed identification for speed-sensorless vector control of induction motor", *IEEE*, 1994.

[25] I. D. Landau, M. Tomizuka, **適応制御システムの理論と實際**, **オーム社**, 1981.

[26] R. Isermann, K. H. Lachmann, D. Matko, Adaptive Control Systems, Prentice-Hall, 1992.

[27] K. S. Narendra, A. M. Annaswamy, Stable Adaptive Systems, Prentice-Hall, 1989.

[28] P. A. Ioannou, Jing Sun, Robust Adaptive Control, Prentice-Hall, 1996.

[29] **윤덕용**, TMS320C31 **마스터**, Ohm, 1998