

영구자석형 동기전동기의 적응제어에 의한 속도제어

Speed Control of Permanent Magnet Synchronous Motor by Adaptive Control

柳 正 雄* · 禹 廣 俊**
(Jeong - Woong Yu · Kwang-Joon Woo)

요 약

이 논문은 영구자석형 동기전동기의 속도제어에 사용되는 제어법중 하나인 기준모델 적응제어법의 알고리즘을 자동회귀이동평균모델로 유도한 것이다. 그리고 유도된 알고리즘을 토대로 PI 제어기를 가진 영구자석형 동기전동기에 마이크로프로세서를 사용하여 기준속도입력에 대한 속도응답이 기준모델에 잘 추종함을 입증하였다. 또한 제어대상인 전동기의 파라미터의 변동에도 오버슈트가 없고 속응성이 있는 속도응답을 얻을 수 있음을 실험을 통하여 확인하였다.

Abstract- The model reference adaptive control algorithm (MRAC), which is one of the methods for controlling the speed of a permanent magnet synchronous motor (PMSM), has been developed using the autoregressive (ARMAX) method. Applying this algorithm to a microprocessor which is used in driving PMSM with PI controller, it has been proved that the response speed of the reference input follows closely that of the reference model. It has also been proved by experiments that the quick speed response without over-shoot could be obtained for the motor system with variable parameters.

1. 서 론

폐환제어(Feed-back control)는 주로 외란입력에 대처하는 것이 주목적인 반면에 적응제어(Adaptive control)는 제어대상의 동특성의 변화, 제어시스템의 파라미터 변동에 대처하는 것이 주목적이다. 그런데 전동기의 속도제어에서는 부하의 관성 및 마찰의 변화나 파라미터 중 온도상승에 의한 저항의 변화, 전력증폭기의 이득변화, 자기포화등의 변동에 직접적으로 전동기의 회전수를 변화시킨다.

이런 이유로 공작기계, 로봇트, 추적망원경 등을

폐환제어법에 응용하기에는 많은 문제점이 발생한다! 따라서 이러한 특성변화에 적응할 수 있는 적응제어방식의 적용에 관한 연구가 요청되고 있다.

여기서 사용된 기준모델적응제어(Model reference adaptive control : MRAC)^{2,3}의 알고리즘은 선형모델추종제어(Linear model following control : LMFC)를 포함한 것으로 기준모델(Reference model)은 동작지수(Index of performance)의 희망값을 나타내고 제어입력은 기준모델의 응답을 추적하는 제어시스템을 구동하기 위한 것이다.

그런데 대부분의 적응제어 알고리즘은 복잡하여 전동기의 가변속시스템을 마이크로프로세서로 실현하기엔 많은 어려움이 있다.

따라서 2개 이상의 전달함수를 갖는 전동기 속도제어시스템을 1계로 근사화시켜 플랜트의 부하

*正 會 員 : 忠北大 工大 電氣工學科 副教授 · 工博

**正 會 員 : 嶺國大 工大 電氣工學科 副教授 · 工博

接 受 日 字 : 1988年 4月 19日

1 次 修 正 : 1989年 2月 27日

상태에 따라 시스템에 접근된 기준모델 적응제어 알고리즘을 유도하였다.

이를 토대로 마이크로프로세서를 사용하여 앞에서 언급된 전동기의 가변속구동시스템을 실현하는 제어회로를 구성하고 그 하드웨어와 소프트웨어를 적용하여 그 타당성을 입증하였다.

2. 수학적 모델링 및 MRAC의 알고리즘

본 영구자석형 동기전동기(Permanent magnet synchronous motor : PMSM)의 구동시스템에 대한 블록선도는 쇠교자속이 일정하고, 진동이 없는

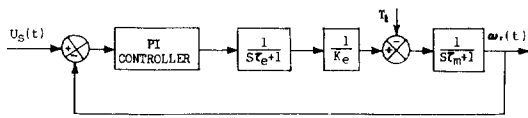


그림 1 PMSM 구동시스템의 블록도

Fig. 1 Block diagram of PMSM drive system.

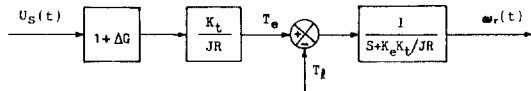


그림 2 PMSM 구동시스템의 근사 블록도

Fig. 2 Approximate block diagram of PMSM drive system.

상태하에서 2계로 근사화시키면 그림 1과 같이 된다.

그림 1에서 입력전압 $U_s(s)$ 에 대한 출력속도는

$$\frac{\omega_r(s)}{U_s(s)} = \frac{1/K_e}{(s\tau_e+1)(s\tau_m+1)} \quad (1)$$

이고, 여기서 K_e 는 역기전력상수, $\tau_e=L/R$, $\tau_m=J R/K_e K_t$ 는 전기적 시정수와 기계적 시정수이다.

위의 식에서 인덕턴스 L 가 권선저항 R 에 비하여 무시할수 있다면 2계인 식 (1)은 1계인 식 (2)로 근사화 시킬 수 있다.

$$\frac{\omega_r(s)}{U_s(s)} = \frac{K_t/JR}{s+(K_e K_t/JR)} \quad (2)$$

그리고 식 (1)를 식 (2)에 의하여 저차화모델

(Reduced modeling)로 변환하고 그 이득 1을 중심으로 섭동하는 시변이득변수로 표시할 때 그림 1을 그림 2와 같이 1계의 전달함수를 갖는 것으로 등가변환할 수 있다.

그림 2에 대한 이산모형을 자동회귀이동평균모델(Autoregressive moving average model with auxiliary input : ARMAX)로 표시하면

$$A_p(q^{-1})y(t) = q^{-1}B_p(q^{-1})u(t) + \xi(t) \quad (3)$$

이고, PMSM구동시스템을 1계로 근사화 하였으므로 식 (3)의 $A_p(p)$ 와 $B_p(q^{-1})$ 는 아래와 같이 표현된다.

$$A_p(q^{-1}) = 1 + a_p q^{-1} \quad (4)$$

$$B_p(q^{-1}) = b_p \quad (5)$$

여기서 q^{-1} 는 후향전이연산자(Backward shift operators)이고, $y(t)$, $u(t)$ 는 시각 t 에서의 샘플치로 회전자속도 ω_r 과 제어기준입력 U_s 를 나타낸다. 또 $\xi(t)$ 는 부하도오크의 변화와 같은 외란을 나타내는 샘플치이며 일반적으로 D.C 옴셋(Off set)을 포함하는 유색잡음(Coloured noise)이나 여기서는 간단히 하기 위하여 비상관잡음(Uncorrelated noise)으로 간주한다^{4), 5)}. d 는 플렌트의 지연시간을 샘플치로 나눈 정수이며, 한 샘플시간 만큼 지연된다고 가정하여 $d=1$ 로 한다.

또한 플렌트에 대한 식 (3)의 기준모델 적응제어 방식은 플렌트의 제어동특성에 따라 기준모델을 아래와 같이 표현할 수 있다.

$$A_M(q^{-1})y^*(t) = q^{-1}B_M(q^{-1})r(t) \quad (6)$$

$$\text{여기서, } A_M(q^{-1}) = 1 + a_M q^{-1} \quad (7)$$

$$B_M(q^{-1}) = b_M \quad (8)$$

이며, $y^*(t)$ 는 출력으로 원하는 속도의 샘플치이고, $r(t)$ 는 원하는 속도의 기준입력이다.

또한 $A_M(q^{-1})$ 는 안정하고 기준모델의 지연시간은 플렌트의 지연시간과 같다고 가정한다. 식 (6)으로 표시된 기준모델에 대한 MRAC의 기본구조는 그림 3과 같이 구성된다.

이 MRAC의 구조는 기준모델의 출력 $y^*(t)$ 와 플랜

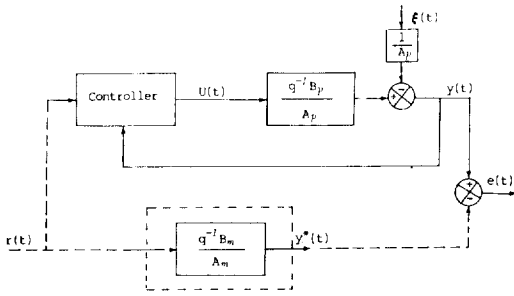


그림 3 MRAC 시스템의 구조
Fig. 3 Scheme of MRAC system.

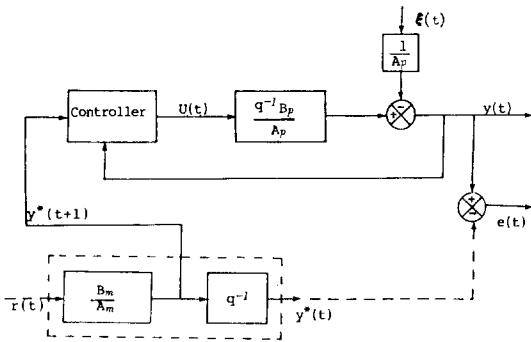


그림 4 그림 3의 변형구조
Fig. 4 Metamorphosed scheme of Fig. 3.

트의 출력 $y(t)$ 와의 오차 $e(t)$ 가 영이 되도록 제어기를 구성하는 방식이다. 그리고 기준모델의 식 (6)에서 식 (7), (8)이 주어지므로 $y(t)$ 의 예측값을 알 수 있다. 따라서 이 예측값을 $r(t)$ 대신 이용하면 그림 3은 그림 4와 같이 재구성할 수 있다.

또한 식 (3)은 다음 예측자(Predictor) 형태로 표현된다.

$$A_M(q^{-1})y(t+1|t) = \alpha(q^{-1})y(t) + \beta(q^{-1})u(t) \quad (9)$$

여기서 $\alpha(q^{-1}) = G(q^{-1})$ (10)

$$\beta(q^{-1}) = F(q^{-1})B_p(q^{-1}) \quad (11)$$

이고, $G(q^{-1})$ 와 $F(q^{-1})$ 는 다음 항등식을 만족하는 다항식이다.

$$A_M(q^{-1}) = F(q^{-1})A_p(q^{-1}) + q^{-1}G(q^{-1}) \quad (12)$$

$$F(q^{-1}) = 1 + f_1q^{-1} + \dots + f_{\alpha-1}q^{-\alpha+1} \quad (13)$$

$$G(q^{-1}) = g_0 + g_1q^{-1} + \dots + q_{d-1}q^{-d-1} \quad (14)$$

여기서 $d=1$ 이고, 식 (13), (14)는 1계모델이므로 $F(q^{-1}) = 1, G(q^{-1}) = g_0$ 이다.

식 (4), (7), (12)로 부터

$$g_0 = a_M - a_P \quad (15)$$

이 되고, 제어입력은 플레트의 예측값 $y = (t+1|t)$ 가 $\hat{y}(t+1)$ 이 되도록 결정한다.

따라서 식 (9)은 식 (16)과 같이된다.

$$\alpha(q^{-1})y(t) + \beta(q^{-1})u(t) = A_M(q^{-1})\hat{y}(t+1) \quad (16)$$

식 (6), (10), (16)으로 부터

$$\begin{aligned} u(t) &= (1/b_p)[b_M r(t) - g_0 y(t)] \\ &= (1/b_p)[b_M r(t) - (a_M - a_P) y(t)] \end{aligned} \quad (17)$$

되나, 식 (17)에서 b_p, a_P, g_0 가 미지수이므로 바로 적용할 수 없다. 따라서 이 값들을 실시간 계수추정 알고리즘을 이용하여 식별한 값으로 대체한다.

이 때 추정계수값을 \hat{a}_P, \hat{b}_P 라 하면 식 (17)은

$$\bar{u}(t) = (1/\hat{b}_P)[\hat{b}_M r(t) - (a_M - \hat{a}_P)y(t)] \quad (18)$$

으로 되고, 플레트모델계수를 실시간으로 추정하기 위한 최소자승 알고리즘은 다음과 같다.⁵⁾

먼저 식 (3)을 벡터형태로 다시쓰면

$$y(t) = \phi(t-1)^T \theta + \xi(t) \quad (19)$$

여기서, $\phi(t-1)^T = [-y(t-1), u(t-1)]$ (20)

$$\theta^T = [a_P, b_P] \quad (21)$$

이고, θ 가 미지수이므로 θ 의 추정치 $\hat{\theta}$ 와 유사회귀 벡터(Pseudo regression vector)로 나타내면 아래와 같이 표현된다.

$$y(t) = \hat{\phi}(t-1)^T \hat{\theta}(t) \quad (22)$$

이 때 순환최소자승추정자(Recursive least square estimator : RLSE)를 이용한 적응알고리즘은

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t-1)\phi(t-1)[y(t)$$

$$-\phi^T(t-1)\hat{\theta}(t-1)] \quad (23)$$

$$P(t-1) = \frac{1}{\lambda} [P(t-2) \frac{p(t-2)\phi(t-1)\phi^T(t-2)p(t-2)}{\lambda + \phi^T(t-1)p(t-2)\phi(t-1)}] \quad (24)$$

여기서, $p(-1) = p_0 = \epsilon I$, ($\epsilon \gg 0$), ($\lambda \leq 1$)

λ 는 망각계수(Forgetting factor)로 계수의 시변성 및 잡음의 크기에 따라 대체로 0.97~0.99로 정한다.^{2), 3)}

적용제어 알고리즘의 전체흐름은 다음과 같다.

- 스텝 1 : 초기치 설정 ($\theta(0)$, $\phi(0)$, $p(0)$, λ)
- 스텝 2 : 출력(전동기 속도) 측정
- 스텝 3 : 식 (23) (24)에 의한 $\hat{\theta}(t)$ 계산
- 스텝 4 : 식 (18)에 의한 $\bar{u}(t)$ 계산
- 스텝 5 : 스텝 2로 돌아감.

위에서 $\hat{\theta}(t) = \theta_0$ 로 수렴하려면 $\bar{y}^*(t)$ 의 지속여기(Persistently exciting)조건이 만족되어야, 하나 대부분의 제어방식에서 계수추정의 수렴을 보장하기 위해 제어입력은 기본적으로 여기조건을 만족시킨다고 가정한다. 이 지속여기조건의 수렴성 증명은 Goodwin 등을 참조한다.⁴⁾

또 망각계수 λ 은 t 시각 이전의 데이터에 대하여 지속적으로 하중치를 곱하여 오랜 정보일수록 현재의 계수추정에 영향을 갖지 않도록 시변 계수추정을 가능하게 한다. 이 때 사용된 λ 의 값은 0.97로 하였다.⁵⁾

이상과 같이 적응제어 알고리즘을 포함한 전동기 속도제어의 전체블럭선도는 그림 5와 같다.⁶⁾

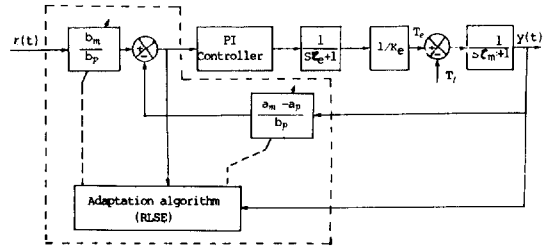


그림 5 MRAC에 의한 속도제어의 블럭도

Fig. 5 Block diagram of speed control by MRAC.

되는 구조로 되어있다.

3. 속도제어 시스템

실험에 사용된 PMSM의 PI제어기의 블럭도는 그림 6과 같다. 이 전동기는 4극 3상, 정격전압 $V=200[V]$, 정격전류 $I=4[A]$, 정격출력 $500[W]$, 정격속도 $1200[rpm]$ 이며 각 파라미터의 값은 표 1과 같다.

그리고 회전자의 위치각 검출은 레졸버(Resolver)를 사용하였으며, 검출된 아나로그값을 R/D변환기에 의하여 디지털값으로 변환하여 사용하였다. 마이크로프로세서에 의한 제어장치는 전체 루우프의 시스템구성을 간단하게 하고, 연산처리 과정에 소요되는 실행시간을 감소시키기 위하여 그림 7과 같이 인텔사의 16-비트 8086을 기초로 하여 구성하였다.

CPU의 클럭주파수는 12MHz로 구성하여 처리속도의 향상을 기하고, 데이터 샘플링주기 설정과 기타 제어동작 처리를 위한 인터럽터 기능을 위하여

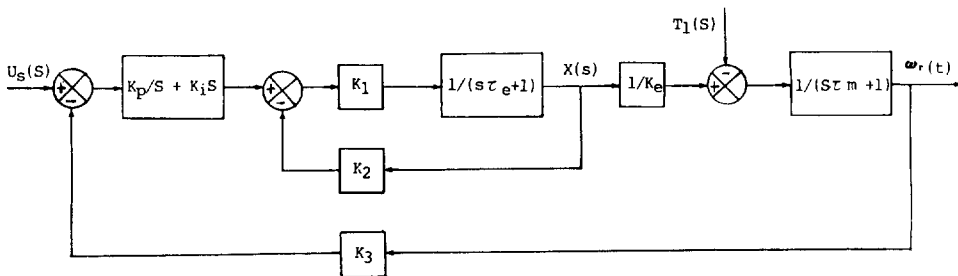


그림 6 PI제어기의 블럭도

Fig. 6 Block diagram of PI controller.

그림 5에서 적응제어알고리즘을 수행하기 전에 앞절에서 언급된 속도에 대한 PI제어기가 외란입력에 대처하고, 다시 적응알고리즘에 의하여 제어

CPU 외부에 타이머를 설치하였다. 총 48K-바이트 중 프로그램모니터에 4K-바이트, 프로그램제어에 4K-바이트 그리고 RAM에 40K-바이트를 할당하여

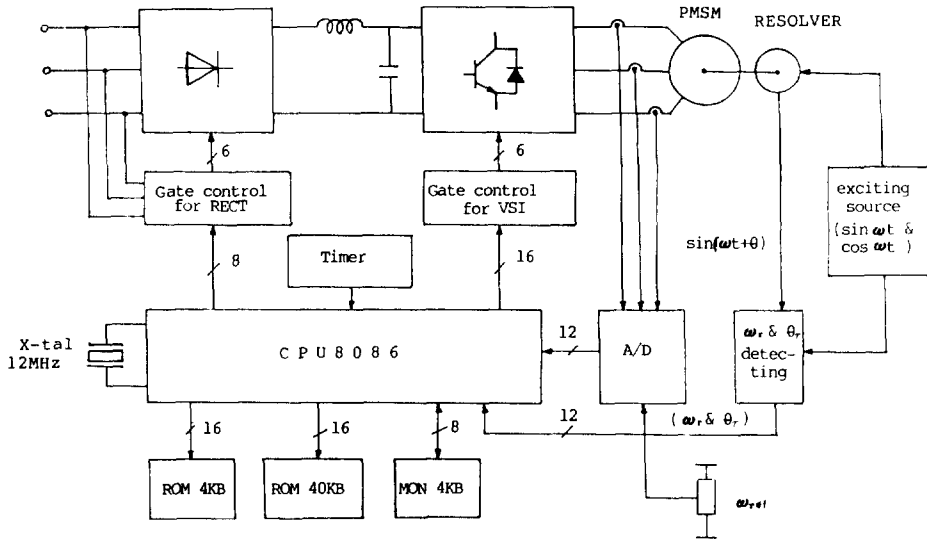


그림 7 MRAC를 위한 마이크로프로세서의 블럭도
 Fig. 7 Block diagram of microprocessor for MRAC.

표 1 PMSM의 파라미터

Table 1 Parameter of PMSM.

$R=4.05[\Omega]$	$J=1.1[\text{kg m sec}^2]$
$L=37.0[\text{mH}]$	$K_1=310$
$K_p=0.2353$	$K_2=1[\text{V/A}]$
$k_i=3$	$k_3=0.4[\text{V/rps}]$
$k_1=742[\text{kg m/A}]$	

사용하였다.

이 제어회로에 사용된 플랜트 즉 전동기 시정수의 계수는 1 계로 근사화된 식 (2)에 대한 이산형 모델이 식 (3)이므로 플랜트의 계수 a_p, b_p 는 다음과 같은 식으로 주어진다. [7, 8, 9]

$$a_p = \exp(-T_s/\tau_{cp}) \tag{25}$$

$$b_p = K(1-a_p), \quad (K: \text{이득}) \tag{26}$$

여기서 기준모델도 플랜트와 가능한 한 같게 표시하여야 하므로 기준모델의 계수도 위의 식과 같이 식 (27) 및 식 (28)로 표현할 수 있다.

$$a_M = \exp(-T_s/\tau_{cM}) \tag{27}$$

$$b_M = K(1-a_M) \tag{28}$$

위의 식에서 T_s 는 샘플링주기로 $2[\text{msec}]$ 이고, τ_{cp} 와 τ_{cM} 은 플랜트 및 기준모델의 시정수이다. 그런데 기준모델의 시정수가 실제 전동기의 시정수보다 크게 하여 ($\tau_{cM} \geq \tau_{cp}$) 기준모델의 계수 a_M, b_M 을 결정한다.

4. 실험 결과

실험에 사용된 전동기의 출력속도에 대한 기준입력(설정속도)을 1 계로 근사화한 시정수 τ_{cp} 및 K 를 그림 6의 블럭도와 표 1에 표시된 전동기의 각종 파라미터를 토대로 구하여 보면 $\tau_{cp} = 100$ 이고, $K = 25$ 이므로 기준모델의 계수는 $\tau_{cM} \geq 100$ 및 $K = 25$ 로 결정할 수 있다.

초기값 $\hat{\theta}(0)$ 선정은 위에서 결정한 τ_{cM} 및 K 값과 식 (27), (28)에 의해 구해진 값으로 하였다. 또 $\phi(0)$ 는 출력 및 입력 데이터에 따라 주어지고, $P(0)$ 는 $\theta(0)$ 의 값이 참값에 근사한 값으로 추정되므로 1로 하였다. 여기서 사용된 P 는 2×2 매트릭스이고, θ 및 ϕ 는 벡터이다.

전동기 속도제어에 MRAC를 적용한 플로우차트는 그림 8과 같으며 전동기의 설정속도를 $950[\text{rpm}]$ 에서 $100[\text{rpm}]$ 까지 증가했을 때 기준모델은 그림 9와 같다.

그리고 전동기의 계단상 속도변화에 MRAC 방법

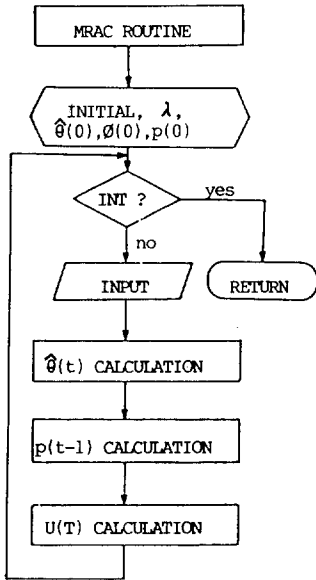


그림 8 MRAC의 플로우차트
Fig. 8 Flow-chart for MRAC.

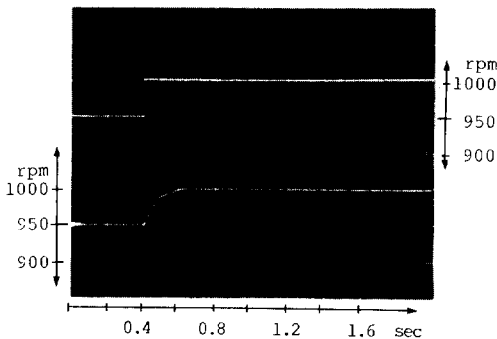


그림 9 속도의 계단변화에 대한 기준모델
(부하 : 0.2pu, 950rpm→1000rpm)
Fig. 9 Reference model at speed step change
(load : 0.2pu, 950rpm→1000rpm).

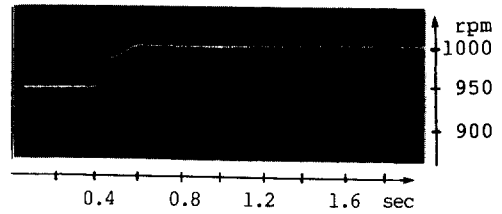


그림 10 MRAC에 의한 속도응답(JR)(950rpm→1000 rpm)
Fig. 10 Speed response at speed reference change by MRAC algorithm (JR) (950rpm→1000 rpm).

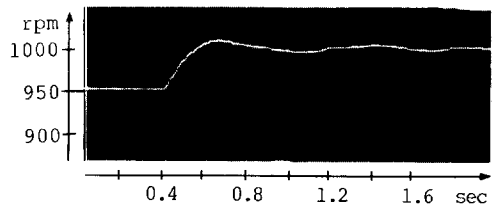


그림 11 PI 제어에 의한 속도응답(JR) (950rpm→1000rpm)
Fig. 11 Speed response at speed reference change by PI Controller (JR) (950rpm→1000 rpm).

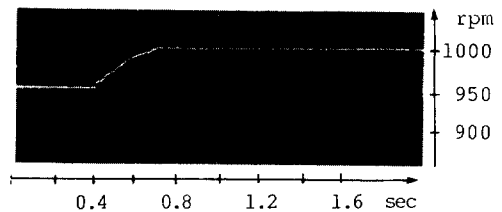


그림 12 MRAC에 의한 속도응답(2JR) (950rpm→1000rpm)
Fig. 12 Speed response at speed reference change by MRAC algorithm (2JR) (950rpm→1000rpm).

을 적용했을 때 속도응답이 그림10과 같이 되어 그림 9의 기준모델에 잘 추종됨을 보여주고 있다.

한편 위와 같은 상태에서 PI 제어만의 계단상 속도변화에 대한 속도응답을 구해보면 그림11과 같은 응답곡선을 얻을 수 있는데 두 경우를 비교하여 보면 정정시간(settling time)은 대략 200[msec]로 거

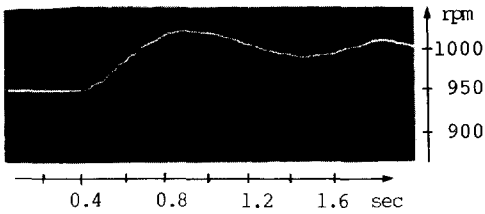


그림 13 PI 제어에 의한 속도응답 (2JR) (950rpm→1000rpm)

Fig. 13 Speed response at speed reference change by PI controller (2JR) (950rpm→1000rpm).

의 같으나 PI 제어기 약간의 오버-슈트(Over-shoot)가 있었음을 보여주고 있다.

또다른 실험으로 부하 0.2[pu], 속도 950[rpm]의 상태하에서 회전자의 위치검출기인 레졸버의 위치를 변경하여 2 배의 고정자권선전류가 흐르도록 고정 배치하고 계단상 속도변화에 대한 속도응답을 측정할 결과 그림 12와 같이 정정시간이 2 배로 증가하였다.

정정시간의 증가는 그림 2의 구동시스템의 근사 블록도에서 $KeKi/JR$ 의 값이 1/2 배로 감소되었음을 의미한다. 이는 Ke, Ki 의 값이 일정하다면 (JR)의 값이 2 배로 증가하였다고 볼 수 있으므로 전동기의 파라미터 변화로 간주할 수 있다. 즉 그림 10, 11은 전동기의 파라미터가 (JR)일때의 속도응답이고, 그림 12, 13은 파라미터가 (2JR)일때의 값이다.

그림 13은 전동기의 파라미터가 (2JR)일때 PI 제어만의 속도응답곡선으로 정정시간이 550(msec)가 되었고, 오버-슈트도 상당히 크게 되었다. 이상으로 두 종류의 제어방식을 비교하여 볼 때 MRAC에 의한 속도제어가 제어시스템의 파라미터 변화에 적응 보상 효과를 가지고 있음을 보여준다.

5. 결 론

본 논문은 영구자석형 동기전동기의 가변속 운전에 전동기의 부정확한 파라미터의 선정과 부하의

관성모멘트의 변화에 MRAC 방식을 적용하여 동특성을 개선할 수 있는 이론과 기법을 제시하였다.

이는 전동기에 대한 전달함수를 1 계로 근사화시켜 이산모형인 ARMAX 모델로 표현하여 최소자승 추정차(RLSF)로 기준모델 적응제어의 알고리즘을 유도하여 16비트 마이크로프로세서로 실현시켰다.

이의 동특성을 PI 제어법과 비교하면 PI 제어기보다 적응제어법이 속응성이 우수하며 오버-슈트가 없는 동특성을 얻을 수 있음을 입증하였다. 그러나 본 연구를 더욱 유용하게 응용하기 위해서는 제어의 정밀성과 강인성 향상 및 고계적응제어 알고리즘의 감략화가 개발 추진되어야 할 것으로 생각된다.

(본 연구는 산학협동재단의 지원에 의하여 수행되었음)

참 고 문 헌

- 1) I.D.Landau, Theory & Practice of Adaptive Control Systems, 1981.
- 2) Y.D.Landau, Adaptive Control-The Model Reference Approach, Marcel Dekker, 1979.
- 3) G.C.Goodwin and K.S.Sin, Adaptive Filtering, Prediction and Control, Prentice Hall, 1984.
- 4) G.C.Goodwin, Dynamic system Identification : Experimental Design and Data Analysis, Academy Press, 1977.
- 5) Young, P, Recursive Estimation and Time-Series Analysis, Springer-Verlag, 1984.
- 6) Susumu Tadakuma, "Microprocessor-Based Adjustable Speed DC Motor Drives using Model Reference Adaptive Control," Proc. of IEEE IECON, 85, pp. 524-528, 1985.
- 7) Haruo Naitoh, "Model Reference Adaptive Control Based DC Motor Speed Controller," Proc. of IEEE IECON, 84, pp. 474-479, 1984.
- 8) W.R.Pearson, "Brushless DC Motor Propulsion using Synchronous Motor for Transient Systems," IEEE Trans. Ind.Electns, vol. IE-31, no.4, Nov.1984.
- 9) Tian-Hua Liu, "Microprocessor-based motion control of a Permanent Synchronous Motor," Proc. of IEEE IECON, 86, pp.563-568 vol.2, 1986.