

# 直流電動機의 速度制御에 對한 適應制御理論의 適用에 關한 研究

(A Study on Application of Adaptive Control Theory to D. C. Motor Speed Control)

金 成 國\*, 金 道 鉉\*\*, 崔 桂 根\*\*\*

(Kim, Sung Kuk Kim, Do Hyun and Choi, Keh-Kun)

## 要 約

最近에 活潑히 研究되고 있는 model reference adaptive control 理論을 microprocessor를 사용하여 直流電動機의 速度制御에 適用할 수 있음을 보였다. 本 論文에서는 一次로 近似시킨 reference model을 設定하고 model의 output과 motor의 output의 差異가 零에 수렴하도록 하였다. 먼저 computer simulation에 의해 reference 入力과 負荷의 變動에 따른 出力을 살펴본 후 microprocessor M 6800을 利用하여 直流電動機를 制御하였는데 input 및 負荷의 급격한 變動에 잘 適用하는 特性을 보여 주었다.

## Abstract

In this paper, the application of model reference adaptive control theory to the D.C motor speed control using the microprocessor is studied.

It is shown that with the use of an adaptive control algorithm the error between output of the motor and the reference model, which is approximated to first order, can be converged to zero.

By computer simulation and the practical implementation with the microprocessor M 6800, can be concluded that the adaptive control system adapts well to the rapid change of the load and reference inputs.

## 1. 서 론

確率制御 理論이 制御機(controller)를 構成하는데 有用하다는 것이 잘 알려졌지만 실제로 적용하는 면에서는 制御對象인 工程(plant)의 動特性(dynamics)과 disturbance를 확실히 알 수 없기 때문에 制御機의 係數(parameter)를 정하는 데 어려움이 있다.<sup>[1]</sup>

\*準會員, 서울大學校 大學院 電子工學科 卒業 軍服務中 (in active service)

\*\*正會員, 明知大學校 工科大學 電子工學科. (Dept. of Electronics Eng. Myong Ji UNiv.)

\*\*\*正會員, 서울大學校 工科大學 電子工學科 (Dept. of Electronics Eng. Seoul National Univ.)  
接受日字: 1981年 3月 6日

<sup>[2]</sup> 특히 plant의 傳達函數의 係數가 변하는 경우에는 最適 制御機(optimal controller) sensitivity가 아주 크기 때문에 그 性能이 最適의 狀態인 것을 보장할 수 없다.<sup>[2]</sup>

이와 같은 문제 때문에 工程의 傳達函數의 係數를 모르는 경우나 係數가 변하는 경우에는 그 係數를 on-line으로 推定(identify) 하면서 制御할 필요가 있다. 이러한 目的으로 最近 몇 년간 適應制御(adaptive control)에 대한 理論이 많이 발표되었다. 그것들의 거의 대부분이 parameter adaptive control로서 工程의 傳達函數의 次數(order), 時間遲延(time delay) 등은 알고 있고 그 係數(parameter)만 모르는 常數라고 가정하였다.<sup>[1][2][6][10]</sup> 그 중에서 특히

MRAC(model reference adaptive control)가 관심을 끌고 있는데 MRAC는 원하는 制御目的을 어떤 理想的인 model로 표시하고 工程(plant)의 출력이 model의 출력과 같게 되도록 制御機를 구성한다. 즉 制御機와 工程으로 구성된 全體 system의 傳達函數가 model의 傳達函數와 같아지게 하는 것이다.<sup>[3][9][11]</sup>

본 논문에서는 適應制御(MRAC)理論을 直流電動機의 速度制御에 適用하여 보았다. 適用制御 algorithm을 구하여 먼저 mincomputer로 simulation을 해 보고 microprocessor를 使用하여 直流電動機의 速度를 制御하였다. 實驗으로 microprocessor를 使用하여 適應制御를 수행할 수 있음을 보였다.

II. 適應制御 理論의 概要

制御하려는 工程(plant)은 다음과 같은 difference eq.을 만족한다고 가정한다.<sup>[5][6][9]</sup>

$$\begin{aligned}
 A(q^{-1})y(k) &= q^{-d} B(q^{-1})u(k) + w(k) \\
 A(q^{-1}) &= 1 + a_1 q^{-1} + \dots + a_n q^{-n} \\
 B(q^{-1}) &= b_0 + b_1 q^{-1} + \dots + b_m q^{-m} \dots \dots \dots (2.1)
 \end{aligned}$$

여기서  $y(k)$ 는 工程의 出力  $u(k)$ 는 入力이다.  $q^{-1}$ 은 back ward shift operator 이고  $d$ 는 罫 또는 陽의 定數로 工程의 時間遲延을 나타낸다.  $w(k)$ 는 測定 不可能한 disturbance 이다.

Plant에 대하여 다음과 같은 가정을 한다.

- ① Polynomial  $A(q^{-1}), B(q^{-1})$ 의 次數( $n, m$ )와 遲延( $d$ )을 알고 있다.
- ② Plant는 minimum phase 이다. 즉  $B(q^{-1})$ 는 stable polynomial 이다.
- ③  $b_0$ 는 罫이 아니다.

Reference model은 다음과 같은 difference eq.으로 표시된다고 가정한다.

$$\begin{aligned}
 A^M(q^{-1})y^M(k) &= q^{-d} B^M(q^{-1})r(k) \\
 A^M(q^{-1}) &= 1 + a_1^M q^{-1} + \dots + a_n^M q^{-n} \\
 B^M(q^{-1}) &= b_0^M + b_1^M q^{-1} + \dots + b_m^M q^{-m} \dots (2.2)
 \end{aligned}$$

여기서  $r(k)$ 는 reference input 또는 command input을 나타내고  $y^M(k)$ 는 reference model의 output을 나타낸다.

$A^M(q^{-1})$ 은 stable polynomial 이며  $A^M(q^{-1}), B^M(q^{-1})$ 의 係數는 罫이 될 수도 있다. 制御機를 구성하는 目的은 plant의 出力  $y(k)$ 와 reference model의 出力  $y^M(k)$ 의 差異를 가능한한 작게 하는 것이다.

여기서는 disturbance를 고려하지 않는 deterministic design 경우만 생각한다.

$w(k)=0$ 이라 가정하면 다음과 같이 하여 plant의 出力  $y(k)$ 가 model의 出力  $y^M(k)$ 와 같게 할 수 있다.

먼저(asymptotically) stable polynomial  $T(q^{-1})$ 을 택한다.

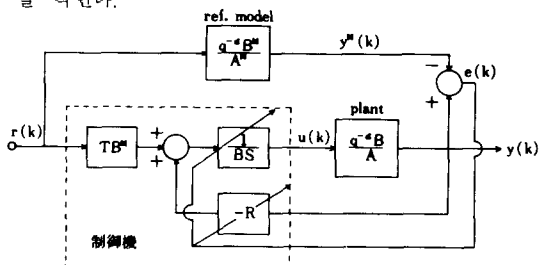


Fig. 1. Control system configuration.

$T(q^{-1}) = 1 + t_1 q^{-1} + \dots + t_{n_T} q^{-n_T}$   
 그러면 다음 polynomial eq.을 만족시키는 유일한 解  $S(q^{-1})$ 과  $R(q^{-1})$ 이 존재한다.

$$\begin{aligned}
 T(q^{-1})A^M(q^{-1}) &= S(q^{-1})A(q^{-1}) + q^{-d}R(q^{-1}) \\
 R(q^{-1}) &= r_0 + r_1 q^{-1} + \dots + r_{n_R} q^{-n_R} \\
 S(q^{-1}) &= 1 + S_1 q^{-1} + \dots + S_{d-1} q^{-d+1} \\
 n_R &= m_{ax}(n_T + n - d, n - 1) \dots \dots \dots (2.3)
 \end{aligned}$$

등식(2.3)이 성립하고 plant의 入力  $u(k)$ 를 식(2.4)이 성립되도록 인가하면 그림 1에서 controller와 plant로 구성된 全體 system의 傳達函數가 model의 傳達函數와 같음을 알 수 있다.

$$\begin{aligned}
 B(q^{-1})S(q^{-1})U(k) &= T(q^{-1})B^M(q^{-1})r(k) \\
 -R(q^{-1})y(k) &\dots \dots \dots (2.4)
 \end{aligned}$$

$k \rightarrow \infty$ 에 따라 output error  $e(k) = y(k) - y^M(k)$ 가 罫이 되도록 polynomial  $R(q^{-1})$ 과  $S(q^{-1})$ (또는  $B(q^{-1})S(q^{-1})$ )의 係數를 계속하여 수정하는 것이 MRAC의 근본 원리이다.

Ref. input의 周波數 成分이 충분히 많으면 係數들이 식(2.3)을 만족하는 본래의 값(real value)에 수렴하지만 係數들이 어떤 條件을 만족하는 다른 값에 수렴해도 output error는 罫에 수렴한다.<sup>[5][11]</sup>

III. Motor Model의 Difference Eq.<sup>[12]</sup>

이 장에서는 直流電動機의 傳達函數로부터 difference eq.을 유도한다 유도된 difference eq.을 利用하여 다음 장에서 適應制御 algorithm을 얻는다. 直流電動機의 角速度에 대한 傳達函數는 다음과 같이

modelling을 할 수 있다.

$$\frac{Y(s)}{U(s)} = \frac{G}{S + \alpha} \dots\dots\dots (3.1)$$

여기서  $Y(s)$ 는 角速度이고  $U(s)$ 는 motor에 가해 주는 入力이다. 식(3.1)을 微分方程式으로 表示하면

$$\dot{y}(t) = -\alpha y(t) + Gu(t) \dots\dots\dots (3.2)$$

본 논문에서는 入力を 한 sampling interval  $[KT, KT + T]$  동안 D/A converter를 통하여 일정하게 가해 주므로  $KT \leq t < KT + T$ 에서 入力은  $u(t) = u(KT)$ 이다. 따라서 식(3.2)로부터 다음 식을 얻을 수 있다.

$$y(KT + T) = e^{-\alpha T} y(KT) + \int_{KT}^{KT+T} e^{-\alpha(KT+T-r)} G d\tau u(KT) \\ G d\tau u(KT) = e^{-\alpha T} y(KT) + \int_0^T e^{-\alpha r} G d\tau u(KT) \dots\dots\dots (3.3)$$

그런데  $e^{-\alpha T}$ 와  $\int_0^T e^{-\alpha r} G d\tau$ 는  $K$ 에 관계없는 常數이므로 각각  $a, b$ 라 놓고 sampling interval  $T$ 를 생략하여 표시하면(본 논문에서는  $T=1$  [sec]로 하였다.)

$$y(K + 1) = a y(K) + b u(K) \dots\dots\dots (3.4)$$

식(3.4)와 같은 difference eq.을 얻는다.

**IV. Adaptive Control Algorithm**

Ref. model을 다음과 같이 취한다.

$$y^M(k) = \frac{1}{2} y^M(k-1) + \frac{1}{2} r(k-1) \dots\dots (4.1)$$

output error  $e(k)$ 는 다음과 같다.

$$e(k) = y(k) - y^M(k) \dots\dots\dots (4.2)$$

filtered error  $e_f(k)$ 를 다음과 같이 정의한다.

$$e_f(k) = A^M(q^{-1}) e(k) \\ A^M(q^{-1}) = 1 - \frac{1}{2} q^{-1} \dots\dots\dots (4.3)$$

식(3.4)에서  $A(q^{-1}) = 1 - aq^{-1}$ 이므로  $e_f(k)$ 를 다음과 같이 표시할 수 있다.

$$e_f(k) = A^M(q^{-1}) y(k) - A^M(q^{-1}) y^M(k) \\ = [A(q^{-1}) + (a - \frac{1}{2}) q^{-1}] y(k) - \frac{1}{2} r(k-1) \\ = bu(k-1) + (a - \frac{1}{2}) y(k-1) - \frac{1}{2} r(k-1) \dots\dots\dots (4.4)$$

그런데 identification (parameter estimation)과 制御入力  $u(k)$ 를 계산하는데 걸리는 時間이 sampling interval에 비해 무시하지 못할 정도이므로  $u(k)$ 를 이전의 data만을 사용하여 구할 수 있도록 하기 위해 식(3.4)을 이용하여 식(4.4)를 약간 수정한다.

$$e_f(k) = bu(k-1) + (a - \frac{1}{2}) (ay(k-2) + bu(k-2)) - \frac{1}{2} r(k-1) \dots\dots\dots (4.5)$$

따라서 적당한 係數  $b_1, b_2, a_2$ 를 사용하여  $e_f(k)$ 를 다음과 같은 형태로 표시할 수 있다.

$$e_f(k) = b_1 u(k-1) + b_2 u(k-2) + a_2 y(k-2) - \frac{1}{2} r(k-1) \dots\dots\dots (4.6)$$

여기서  $b_1, b_2, a_2$ 는 미지의 상수이다.  $e_f(k)$ 를 간단히 표시하기 위해 vector  $\theta$ 와  $\varphi(k)$ 를 정의 한다.

$$\theta^T = [b_1 \quad b_2 \quad a_2] \\ \varphi^T(k) = [u(k) \quad u(k-1) \quad y(k-1)]$$

그러면  $e_f(k)$ 는 다음과 같다.

$$e_f(k) = \theta^T \varphi(k-1) - \frac{1}{2} r(k-1) \dots\dots\dots (4.7)$$

$\theta$ 의 estimate를  $\hat{\theta}$ 로 표시하고  $e_f(k-1)$ 의 estimate를 다음과 같이 정의한다.

$$\hat{e}_f(k-1 | k-1) = \hat{\theta}^T(k-1) \varphi(k-2) - \frac{1}{2} r(k-2) \dots\dots\dots (4.8)$$

Estimation error를 다음과 같이 정의한다.

$$\epsilon(k-1) = e_f(k-1) - \hat{e}_f(k-1 | k-1) \dots\dots\dots (4.9)$$

식(4.7)과 (4.8)을 이용하여  $\epsilon(k-1)$ 을 나타낸다.

$$\epsilon(k-1) = (\theta - \hat{\theta}(k-1))^T \varphi(k-2) \dots\dots\dots (4.10)$$

식(4.10)으로부터  $\epsilon^2(k-1)$ 을 최소가 되게 하는 gradient type algorithm을 이용하여 다음의 parameter estimation algorithm을 얻는다.<sup>[11][13]</sup>

$$\hat{\theta}(k) = \hat{\theta}(k-1) + a(k) \frac{\varphi(k-2)}{1 + \varphi^T(k-2)\varphi(k-2)} \epsilon(k-1) \dots\dots\dots (4.11)$$

$a(k)$ 는 보통 1이고  $a(k) = 1$ 로 해서  $\hat{b}_1(k)$ 가 0이 되는 경우에만  $a(k) = \frac{1}{2}$ 로 한다. Parameter estimate  $\hat{\theta}(k)$ 를 사용한  $e_f(k+1)$ 의 estimate  $\hat{e}_f(k+1 | k)$ 가 0이 되도록 control law를 식(4.12)와 같이 정하고 入力  $u(k)$ 를 구한다.

$$\hat{\theta}^T(k) \varphi(k) - \frac{1}{2} r(k) = 0 \dots\dots\dots (4.12)$$

$$u(k) = \frac{1}{\hat{b}_1(k)} \left( \frac{1}{2} r(k) - \hat{b}_2(k) u(k-1) - \hat{a}_2(k) y(k-1) \right) \dots\dots(4.13)$$

Reference input  $r(k)$ 가 bound 되면  $u(k)$ ,  $y(k)$ 도 bound 되고 이때  $\lim_{k \rightarrow \infty} e_f(k) = 0$  됨이 알려졌다.<sup>[7]</sup>  
 $\lim_{k \rightarrow \infty} e_f(k) = 0$  이면  $A^M(q^{-1})$ 이 stable polynomial 이므로  $\lim_{k \rightarrow \infty} e(k) = 0$  이 된다는 사실을 보여준다.

V. Simulation 結果

Simulation 은 FACOM minicomputer 로 행하였다. 먼저 motor 의 difference eq. (3.4)에서 係數가  $a=0.3$ ,  $b=0.8$  이고 ref. input 이  $r(k)=1000$  으로 일정한 경우에 parameter estimate 의 초기치를  $b_1(0)=1$ ,  $b_2(0)=0$ ,  $a_2(0)=0$  으로 놓고 simulation 을 하여 결과를 그림 2에 나타냈다.

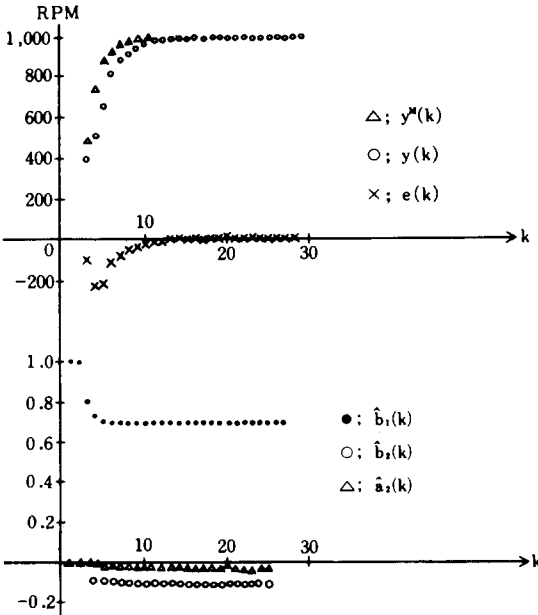


Fig. 2. Motor :  $y(k) = 0.3 y(k-1) + 0.8 u(k-1)$   
 ref. input :  $r(k) = 1000$ .

다음에 위의 simulation 과정에서 충분히 수렴한 뒤에 difference eq. 의 계수만  $a=0.8$ ,  $b=0.1$ 로 변화시켰을 때의 결과를 그림 3에 나타냈다.

마지막으로 두번째 과정에서 ref. input 만을 500으로 변화시킨 결과를 그림 4에 나타냈다.

制御機 係數의 본래의 값(real value)은 첫째 경우  $b_1=0.8$ ,  $b_2=-0.16$ ,  $a_2=-0.06$  이고 두번째 경우  $b_1=0.1$ ,  $b_2=0.03$ ,  $a_2=0.24$  인데 모두 다른 값에

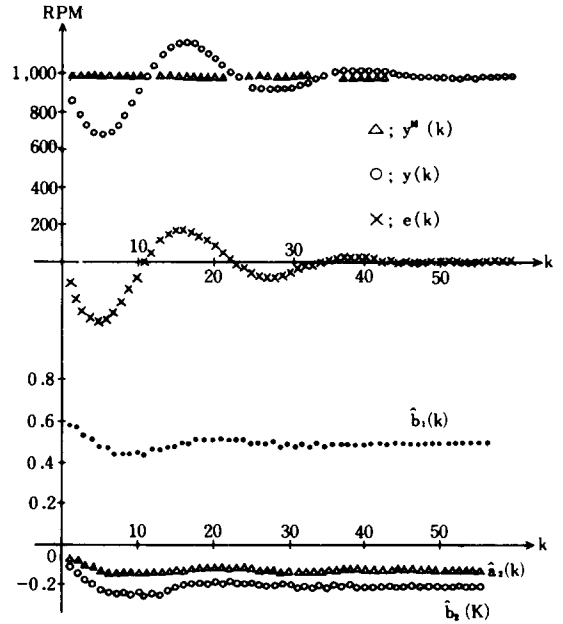


Fig. 3. Motor :  $y(k) = 0.8 y(k-1) + 0.1 u(k-1)$   
 ref. input :  $r(k) = 1000$ .

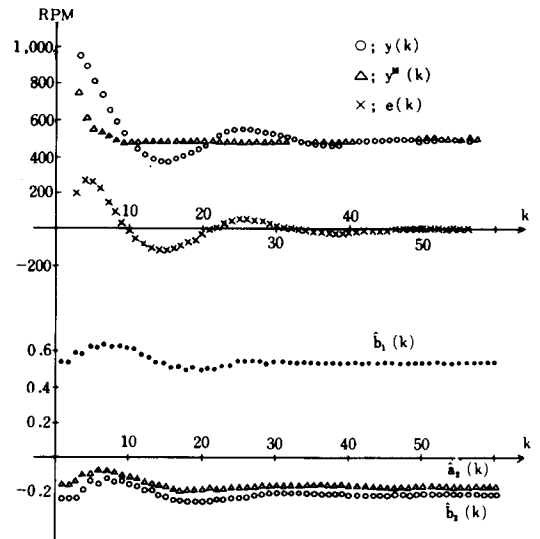


Fig. 4. Motor :  $y(k) = 0.8 y(k-1) + 0.1 u(k-1)$   
 ref. input :  $r(k) = 500$   
 $y(0) = y^M(0) = 1000$ .

수렴하였다. 본래의 값에 수렴했을 때 ref. input 만을 바꾸면 出力 오차(output error)가 0이 되지만 그림 4에서 보는 바와 같이 係數가 다른 값에 수렴했으

므로 ref. input을 변화시킨 경우에 出力오차가 零으로 수렴하는데 비교적 많은 시간이 걸렸다.

Ref. input의 周波數 性분이 충분히 많은 경우에는 制御機의 係數가 본래의 값에 수렴한다. 수렴속도 (rate of convergence)는 推定 (estimate)하는 係數의 數가 적을수록 빠르다.

본 논문에서 電動機의 傳達函數를 1次로 생각할 수 있는 것을 굳이 2次로 한다고 하면 계산하는 양이 많아져서 microprocessor M 6800을 사용하지 못할 수 있을 뿐만 아니라 수렴속도가 늦어져서 制御機의 性能이 더 나빠질 수 있기 때문이다.

VI. 實驗 및 測定

본 논문에서는 microprocessor M 6800을 사용하여 그림 5와 같은 실험회로를 구성하였다. 이때 program은 M 6800 assembly language로 짰으며 length는 2 k. byte 정도 되었다.

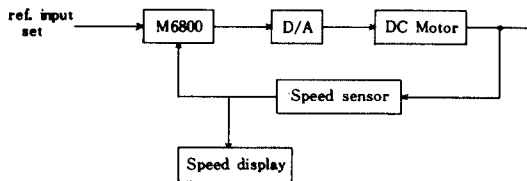


그림 5. 실험 계통도

Ref. input은 BCD 入力 switch에 의해 set 해 주며 모든 연산은 floating point로 했다. Motor의 회전속도는 일정한 간격으로 구멍이 뚫린 원판 lamp와 photo TR에 의해 측정되며 이를 BCD로 바꾸어 display 시킨다.

Microprocessor에 의한 制御의 概要를 설명하면 원하는 속도를 set 해 준 다음 reset 단자를 누르면 reset routine에 들어 가는데 여기서는 연산용의 RAM과 display용의 RAM을 clear 해 주고 set 된 ref. input을 받아 들어 RAM의 일정한 장소에 저장하고 interrupt를 기다린다.

Interrupt는 1초에 한번씩 걸려 制御하게 되는데 interrupt가 걸리면 motor의 入力  $u(k)$ 를 내보내고 motor의 속도  $y(k)$ 를 읽어 들인 다음 係數를 estimate하고 制御入力  $u(k)$ 를 계산한다. 다음에 interrupt가 걸리면 PIA와 D/A converter를 통하여 制御入력을 motor에 가해 주어 制御하게 된다. 1초마다 측정된 회전수는 256 byte RAM에 저장되며 256 byte가 모두 차면 정지하여 RAM의 내용을 차례로

display 하도록 되어 있다.

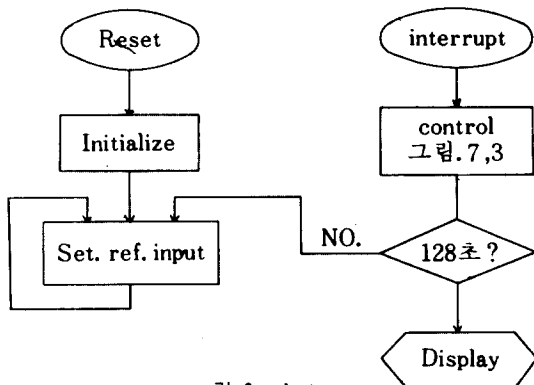


그림 6. 순서도

실험에서는 motor의 선형부분을 고려하여 motor에 가해 주는 入力이 최저 1.2 [V]에서 최고 10 [V]가 되도록 하였다. 즉 식 (4.13)에서 계산한 入力이 1.2 [V] 이하이면 1.2 [V]로 하고 10 [V] 이상이면 10 [V]로 하도록 하였다.

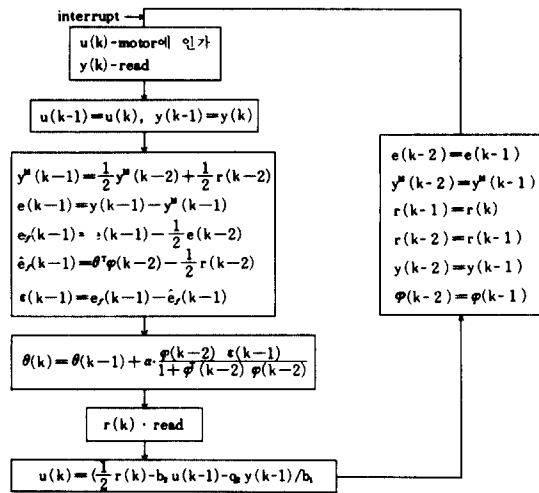


그림 7. 제어 순서도

制御機 係數의 초기치는 reset 단자를 누르면  $b_1(0) = 1, b_2(0) = 0, a_2(0) = 0$ 이 되도록 하였다. 無負荷인 경우  $r(k)$ 를 200과 300으로 했을 때의 出力을 그림 8에 나타냈다. 또 無負荷인 경우 처음에  $r(k)$ 를 200으로 하고 도중에 400으로 변화시켰을 때의 出力이 그림 9와 같다. Load에 발전기를 사용하고  $r(k)$ 는 300으로 한 경우를 그림 10에 나타냈다. 그리고 電動機가 無負荷로 돌아가는 중에 reset 단자를 누르고  $r(k) = 200$ 으로 하여 어느 정도 수렴한 뒤에

갑자기 발전기를 負荷로 연결했을 때의 결과를 그림 11에 나타냈다. 여기서 사용한 숫자의 25/6 배가 1분당 회전수이다.

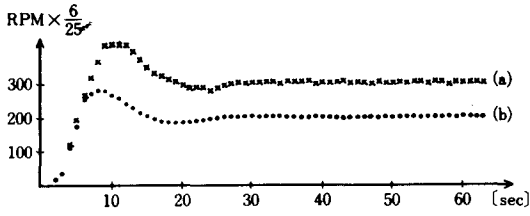


그림 8. 無負荷 (a)  $r(k) = 300$ , (b)  $r(k) = 200$

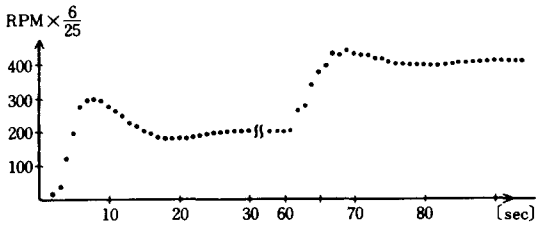


그림 9. 無負荷  $r(k) : 200 \rightarrow 400$

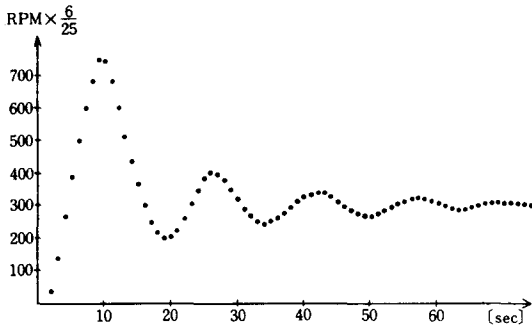


그림 10. 負荷 : 發電機,  $r(k) = 300$

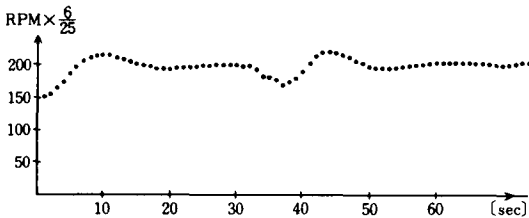


그림 11.  $r(k) = 200$   $k \leq 30$  : 無負荷  
 $k > 30$  : 負荷 - 發電機

실험 하기 전에 microprocessor M 6800을 사용하여 simulation을 하여 보았다. 여기서 나타난 유효숫자

에 의한 오차가  $\pm 0.5\%$ 였다. 또 motor에 일정한 電壓을 가지고 定常狀態에서 測定한 速度가 대략  $\pm 0.7\%$  정도로 오차가 있었다. 실험 결과로 나타난 오차는 이 두 가지 오차를 합한 정도였다. 유효숫자를 늘이고 속도 측정을 정확히 하면 더 좋은 결과를 얻을 것으로 기대된다.

VII. 結 論

본 논문에서 適應制御 理論을 直流電動機의 速度制御에 適用하였다.

Microprocessor를 適應制御의 비교적 간단한 algorithm을 수행하는데 사용할 수 있음을 보이고 만족스런 結果를 얻었다.

Ref. input을 일정하게 주었을 때 無負荷인 경우에 定常狀態의 오차가 1% 이내이고 약 30초 후에 定常狀態에 도달하였다. 負荷에 따라 定常狀態에 도달하는 시간이 변하였고 負荷의 變動에 바로 適應하는 것을 알 수 있었다.

負荷의 變動에 따라 制御機의 係數를 바꾸어 주므로 model과 같은 특성을 갖게 한다. 이러한 制御方式은 regulator 뿐만 아니라 원하는 出力에 따라 ref. input을 계산해 주면 tracking 문제에도 適用할 수 있다. 수렴속도를 개선하면 훌륭한 制御方式이 될 수 있을 것으로 생각되며 곱셈 나눗셈의 연산이 있고 성능이 좋은 microprocessor를 이용하면 좀 더 복잡한 制御 문제를 다룰 수 있을 것이다.

參 考 文 獻

1. K.J. Astrom, U. Borisson, L.I. Jung, E. Wittenmark. "Theory and application of selftuning regulators" Automatica, Vol. 13, pp. 457-478, 1977.
2. G. C. Goodwin, P. J. Ramadge. "Design of restricted complexity adaptive regulators" IEEE Trans on Automat Contr Vol. AC-24 No. 4, Aug. 1979, pp. 584-588.
3. T. Ionescu, R. V. Monopoli. "Discrete Model reference adaptive control with an augmented error signal" Automatica, Vol. 13, pp. 507-517. 1977.
4. K. J. Astrom, "Introduction to stochastic control theory," New York: Academic 1970.
5. Bo, Egardt "Stability analysis of discrete-time adaptive control schemes" IEEE trans on Automat Contr Vol. AC-25 No. 4 Aug. 1980.

6. Bo. Egardt, "Unification of some discrete-time adaptive control schemes" IEEE Trans on Automat Contr. Vol. AC-25 No. 4 Aug. 1980.
7. G. C. Goodwin, P. J. Ramadge, P. E. Caines, "Discrete-time multivariable adaptive control" IEEE Trans. on Automat, Contr. Vol. AC-25 No. 3, Jun. 1980.
8. K. S. Narendra, Y. H. Lin "Stable discrete adaptive control" IEEE Trans on Automat Contr. Vol. AC-25 No. 3, Jun. 1980.
9. Bo. Egardt, "Unification of some continuous-time adaptive control schemes" IEEE Trans. on Automat. Contr Vol. AC-23 No. 4 Aug. 1979.
10. K. B. Narendra, L. S. Valavani, "Stable adaptive controller design-direct control" IEEE Trans. on Automat. Contr Vol. AC-23 No. 4 Aug. 1978.
11. Y. D. Landau "Adaptive control, the model reference approach" New York and Basel, Marcel Dekker, Inc.
12. J. A. Cadzow, H. R. Martens. "Discrete time and computer control systems" Prentice-Hall, 1970.
13. JEAN-JACQUES J. FUCHS, Discrete Adaptive Control : A sufficient Condition for stability and amplifications." IEEE Trans. on Automat Contr. Vol. AC-25, No. 5, October 1980.

