

Explicit M.R.A.C. 알고리즘을 이용한 직류 전동기 속도 제어 (D.C. Motor Speed Control Using Explicit M.R.A.C. Algorithms)

金 鍾 煥*, 朴 俊 烈**, 崔 桂 根***

(Chong Hwan Kim, Joon Yeal Park and Kye Keun Choi)

要 約

본 연구는 explicit M.R.A.C. 알고리즘으로 마이크로프로세서를 사용하여 직류전동기 속도 제어를 하였다.

실함에 사용된 적응 제어 알고리즘으로는 먼저 지수함수적 가중 최소자승법(exponentially weighted least square method: E.W.L.S) 알고리즘의 계산상의 불안정과 수행시간을 최소화 하기 위하여 시간 지연이 있는 E.W.L.S 알고리즘에 UDU^T -인수화법을 도입한 UDU^T -인수화법 알고리즘을 사용하였고, 또한 gradient-type의 알고리즘으로도 초기에 dead-zone을 갖는 직류전동기를 효과적으로 제어하기 위하여 gradient-type의 알고리즘에 smoothing polynomial과 상수 θ 을 사용한 SM gradient-type의 알고리즘을 제안하였다.

UDU^T 인수화법 알고리즘을 적용한 결과 수행시간이 단축되었으며, SM gradient-type의 알고리즘의 경우는 dead-zone을 위한 기동전압을 사용하지 않고도 효과적인 속도제어를 할 수 있음을 확인하였다.

Abstract

In this paper, the application of the explicit M.R.A.C. algorithms to the D.C. motor speed control using the microprocessor is studied.

The adaptation algorithms are derived from the gradient method and the exponentially weighted least square [E.W.L.S.] method. In order to minimize the computational instability of the E.W.L.S. method, the adaptation algorithm of UDU^T factorization method is developed, and because of the characteristics of the D.C. motor [dead-zone phenomenon], the SM. gradient type algorithm is also improved from the gradient type algorithm.

Computer simulations and experiments show that these algorithms adapt well to the rapid change of the reference input and the load.

I. 序 論

*準會員, ***正會員, 서울대학교 工科大学 電子工學科
(Dept. of Electronic Eng., Seoul National Univ.)

**正會員, 弘益대학교 工科大学 電子工學科
(Dept. of Electronic Eng., Hong Ick Univ.)

接受日字: 1983年 3月 28日

적응 제어방식은 제어하고자 하는 공정의 특성을 확실히 모르는 경우나 부하 변동등으로 공정의 파라미터가 변하는 경우에도 원하는 목적대로 제어할 수 있다. 이러한 적응제어 알고리즘으로는 hyperstability 알고리즘(1)과 과거 정보에도 비중을 두어 파라미터를 찾

는 E. W. L. S¹² 13 알고리즘과 전단계의 정보를 이용하여 파라미터를 찾는 gradient-type의 알고리즘¹⁴ 등이 연구되어 있다.

그러나 hyperstability 알고리즘이나 E. W. L. S. 알고리즘은 마이크로 컴퓨터를 사용하여 수행시키면 계산상의 불안정과 계산시간이 오래 걸리는 단점을 갖고 있다. 이러한 점을 보완하기 위하여 許¹⁵는 E. W. L. S. 알고리즘으로 부터 제공된 알고리즘을 유도하였다.

본 연구는 許의 제공된 알고리즘에 time delay도 함께 고려하여 E. W. L. S. 알고리즘에 인수화법¹⁶을 적용한 UDC^T 인수화법 알고리즘으로 time delay가 있는 직류전동기의 속도 제어를 하였다.

또한 김⁷은 직류전동기의 dead-zone의 특성을 고려하여 gradient-type의 알고리즘을 적용한 직류전동기 속도제어에서 기동 전압(1.2V)을 사용하였으나, 본 연구에서는 gradient-type의 알고리즘에 smoothing polynomial과 상수 θ 를 사용하여 유도된 SM.gradient-type의 알고리즘을 적용함으로써 기동전압없이 dead-zone을 벗어날 수 있게 하였으며, 그 결과 수렴속도도 개선되었다.

이상의 제안된 두 방법의 알고리즘을 적용하는 적응 제어 방식으로는 예측기를 사용하지 않고 각 순간의 출력을 측정하여 바로 공정과 model reference와의 전달함수를 같게 하는 explicit M. R. A. C¹⁸ (model reference adaptive control) 방식을 사용하였고, 이를 마이크로프로세서 z-80을 이용하여 직류전동기 속도 제어를 하여 좋은 결과를 얻었다.

II. 線型 制御器 設計¹⁸

制御하려는 공정은 다음과 같이 단일 入力, 出力의 時不變 差分方程式으로 표시된다고 假定한다.

$$A(q^{-1}) Y(k) = q^{-d} B(q^{-1}) U(k) + W(k) \quad (1)$$

$$A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_n q^{-n}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \dots + b_m q^{-m}$$

여기서 $y(k)$ 는 공정의 出力이고, $U(k)$ 는 공정의 入力이다.

q^{-1} 은 backward shift operator이고, d 는 陽의 常數로 공정의 時間 遲延을 나타낸다. $W(k)$ 는 測定 不可能한 disturbance이다.

이 공정에 대하여 다음과 같은 假定을 한다.

- 1) 차수 n, m 과 시간지연 d 는 알고 있다.
- 2) 공정은 minimum phase이다.
- 3) b_0 는 零이 아니다.

Reference model은 다음과 같은 差分 方程式으로 표시된다고 假定한다.

$$A_M(q^{-1}) Y_M(k) = q^{-d} B_M r(k) \quad (2)$$

$$A_M(q^{-1}) = 1 + a_{1M} q^{-1} + \dots + a_{nM} q^{-n}$$

$$B_M(q^{-1}) = b_{0M} + b_{1M} q^{-1} + \dots + b_{mM} q^{-m}$$

여기서 $Y_M(k)$ 와 $r(k)$ 는 model의 出力과 入力を 나타내며 bound 되어 있다. $A_M(q^{-1}), B_M(q^{-1})$ 의 계수는 0이 될 수도 있고 $A_M(q^{-1})$ 은 asymptotically stable polynomial이다.

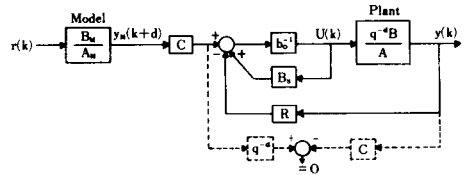


그림 1. 선형 제어기
Fig. 1. Linear controller.

여기서는 disturbance를 고려하지 않은 deterministic한 경우만 다루기로 한다.

$W(K) = 0$ 에서 制御 目的은 다음과 같다.

$$A_M(q^{-1}) Y(k) = q^{-d} B_M(q^{-1}) r(k) \quad (3)$$

식(3)과 그림 1에서 $C(q^{-1})$ 은 다음 조건을 만족해야 한다.

$$C(q^{-1}) = A(q^{-1}) S(q^{-1}) + q^{-d} R(q^{-1}) \quad (4)$$

$$C(q^{-1}) = 1 + C_1 q^{-1} + \dots + C_{nc} q^{-nc}$$

$$S(q^{-1}) = 1 + S_1 q^{-1} + \dots + S_{ns} q^{-ns}$$

$$R(q^{-1}) = r_0 + r_1 q^{-1} + \dots + r_{nr} q^{-nr}$$

$$B_s(q^{-1}) = b_0 - B(q^{-1}) S(q^{-1})$$

식(4)는 $n_s = d-1, n_r = \max(n-1, n_c-d)$ 이면 唯 한 解 $S(q^{-1}), R(q^{-1})$ 을 갖는다.

Plant-model error는 다음과 같다.

$$e(k) = y(k) - y_M(k) \quad (5)$$

제어 목적은 다음식이 만족되면 이루어진다.

$$C(q^{-1}) e(k+d) = 0 \quad (6)$$

식(6)은 다음과 같이 쓸 수 있다.

$$\begin{aligned} C(q^{-1}) e(k+d) &= B(q^{-1}) S(q^{-1}) U(k) + R(q^{-1}) \cdot \\ & \quad Y(k) - C(q^{-1}) Y_M(k+d) \\ &= b_0 U(k) + P_0^T \Phi_0(k) - C(q^{-1}) \cdot \\ & \quad Y_M(k+d) \\ &= P^T \phi(k) - C(q^{-1}) Y_M(k+d) \quad (7) \end{aligned}$$

$$\Phi_0^T(k) = [U(k-1), \dots, U(k-m-d+1), Y(k), \dots, Y(k-nR_s)]$$

$$P_0^T(k) = [b_0 S_1 + b_{11}, b_0 S_2 + b_{12} S_1 + b_2, \dots, b_m S_{d-1} + r_0, \dots, r_{nr}]$$

$$\phi(k) = [U(k); \Phi_0^T(k)], P^T = [b_0; P_0^T]$$

식 (7)을 0으로 놓음으로써 다음과 같은 制御 入力로 제어 목적이 이루어진다.

$$U(k) = \frac{1}{b_0} [C(q^{-1}) Y_M(k+d) - P_0^T \phi_0(k)] \quad (8)$$

III. 直流電動機 모델링 [9] [10]

直流電動機의 角速度에 대한 傳達函數는 다음과 같이 모델링할 수 있다.

$$\frac{Y(s)}{U(s)} = \frac{G}{s + \alpha} \quad (9)$$

여기서 Y(s)는 角速度이고, U(s)는 直流電動機에 가해 주는 入力이다.

식 (9)를 微分方程式으로 표시하면 다음과 같다.

$$Y(t+1) = -\alpha Y(t) + GU(t) \quad (10)$$

入力は 샘플링 週期 T동안 일정하게 가해 수므로 U(t) = U(KT), KT ≤ t < (K+1)T라 할 수 있다.

따라서 식(10)는 다음과 같이 쓸 수 있다.

$$Y(KT+T) = e^{-\alpha T} Y(KT) + \int_{KT}^{(K+1)T} e^{-\alpha(KT+r)} G dr \cdot U(KT) \\ = e^{-\alpha T} Y(KT) + \int_0^T e^{-\alpha r} G dr \cdot U(KT) \quad (11)$$

식(11)에서 e^{-αT}와 ∫₀^T e^{-αr} G dr는 K에 관계없는 常數이므로 각각 a, b라 놓고 T = 1 秒로 하면 다음과 같다.

$$Y(K+1) = aY(K) + bU(K) \quad (12)$$

IV. Explicit M. R. A. C. 알고리즘

실험에 사용할 알고리즘에 대하여 알아본다.

1. UDU^T 인수화법 알고리즘

Explicit M. R. A. C 알고리즘은 공정의 파라미터를 모르므로 벡터 P₀, 계수 b₀ 대신 adaptation mechanism에 의하여 추정된 $\hat{p}_0(k)$, $\hat{b}_0(k)$ 를 사용하여 유도한다. 그러면 制御 入力 U(k)는 식(8)에서 다음과 같이 된다.

$$U(k) = \hat{b}_0^{-1}(k) [C(q^{-1}) Y_M(k+d) - \hat{p}_0^T(k) \phi_0(k)] \quad (13)$$

식(13)에서 다음의 관계를 얻는다.

$$C(q^{-1}) Y_M(k+d) = \hat{p}^T(k) \phi(k) \quad (14)$$

단, $\hat{p}^T(k) = [\hat{b}_0(k); \hat{p}_0^T(k)]$

식(14)을 식(7)에 대입하면 다음과 같다.

$$c(q^{-1}) e(k+d) = [\hat{p} - p(k)]^T \phi(k) \quad (15)$$

Filtered plant-model error e_r(k)를 다음과 같이

정의한다.

$$e_r(y) = c(q^{-1}) e(k) = [p - \hat{p}(k-d)]^T \phi(k-d) \quad (16)$$

제어 목적은 다음 條件을 만족시키는 adaptation mechanism을 찾는 것이다.

$$\lim_{k \rightarrow \infty} e_r(k) = 0 \quad \forall e_r(0) \neq 0, \quad \hat{p}(0) \in R \quad (17)$$

$$\| \phi(k) \| \leq M < \infty \quad \forall k$$

식(17)은 plant-model error가 0으로 收斂하고, 공정의 入力와 出力은 bound되어 있음을 의미한다.

그러나 이러한 문제는 時間 遲延때문에 직접 해결할 수 없으므로 auxiliary error를 다음과 같이 정의한다.

$$\bar{e}(k) = [\hat{p}(k-d) - \hat{p}(k)]^T \phi(k-d) \quad (18)$$

그리고 e_r(k)와 $\bar{e}(k)$ 를 합한 augmented error e*(k)를 정의한다.

$$e^*(k) = e_r(k) + \bar{e}(k) = [p - \hat{p}(k)]^T \phi(k-d) \quad (19)$$

식(19)의 誤差는 실제 공정의 係數p와 추정치 $\hat{p}(k)$ 의 差에 기인한 것이다. 이 추정오차를 最小化시키는 알고리즘을 구하기 위하여 다음과 같은 평가함수를 둔다.

$$J(k) = \sum_{j=d}^k \lambda_2 \lambda_1^{k-j} [c(q^{-1}) y(j) - \hat{p}^T(k) \phi(j-d)]^2 \quad (20)$$

단, 0 < λ₁ < 1

식(20)을 $\hat{p}(k)$ 로 微分하여 이를 0으로 두면 다음과 같은 $\hat{p}(k)$ 에 대한 관계식을 얻는다.

$$\hat{p}(k) = [\Phi(k) \lambda(k) \Phi(k)^T]^{-1} \Phi(k) \lambda(k) Y(k) \quad (21)$$

단, $\Phi(k) = [\phi(0); \phi(1); \dots; \phi(k-d)]^T$

$Y(k) = [\bar{y}(d); \bar{y}(d+1); \dots; \bar{y}(k)]^T$

$\bar{y}(j) = c(q^{-1}) y(j)$

$$\lambda(k) = \begin{bmatrix} \lambda_2 \lambda_1^{k-d} & \lambda_2 \lambda_1^{k-d-1} & & 0 \\ & \ddots & \ddots & \\ 0 & & \lambda_2 \lambda_1 & \lambda_2 \end{bmatrix}$$

또한 $\Gamma(k) = [\Phi(k) \lambda(k) \Phi(k)^T]^{-1}$ 라 놓으면 (22)

$$\hat{P}(k) = \Gamma(k) \Phi(k) \lambda(k) Y(k) \quad (23)$$

(22)식에 matrix inversion lemma를 이용하여 반복식을 구하면 다음과 같은 E.W.L.S 알고리즘을 얻는다.

$$\hat{p}(k+1) = \hat{p}(k) + L(k+1) [\bar{y}(k+1) - \phi^T(k-d+1) \cdot \hat{p}(k)] \quad (24)$$

$$\Gamma(k+1) = \frac{1}{\lambda_1} [1 - L(k+1) \phi(k-d+1)^T] \Gamma(k)$$

$$L(k+1) = \frac{\frac{\Gamma(k) \phi(k-d+1)^T}{\lambda_1}}{\frac{1}{\lambda_2} + \phi(k-d+1)^T \frac{\Gamma(k)}{\lambda_1} \phi(k-d+1)}$$

이러한 E. W. L. S 알고리즘에 UDUT^T 인수화법을 도입한다. 이때 시간 지연 d를 고려하여 다음과 같이 관계식을 정의할 수 있다.

$$f = U^T(k-1) \phi(k-d) \tag{25}$$

$$V = D(k-1) f / \lambda_1 \tag{26}$$

$$\text{단 } f^T = [f_1 \ f_2 \ \dots \ f_n]$$

$$V^T = [V_1 \ V_2 \ \dots \ V_n]$$

$$V_i = d_i(k-1) f_i / \lambda_1 \quad i = 1, \dots, n.$$

$$r_i = \frac{1}{\lambda_2} + V_i f_i \tag{27}$$

$$d_i(k) = d_i(k-1) / \lambda_1 \lambda_2 r_i \tag{28}$$

$$K_2^T = [V_1 \ 0 \ \dots \ 0] \tag{29}$$

다음의 식들을 두번째 단계부터 n번째까지 반복적으로 구하여 식(24)의 L(k)를 구할 수 있다.

$$r_j = r_{j-1} + V_j f_j \tag{30}$$

$$d_j(k) = \frac{d_j(k-1)}{\lambda_1} \cdot \frac{r_{j-1}}{r_j} \tag{31}$$

$$u_j(k) = u_j(k-1) + w_j k, \quad \text{단, } w_j = -\frac{f_j}{r_{j-1}} \tag{32}$$

$$k_{j+1} = k_j + v_j u_j(k-1) \tag{33}$$

단, $u_j(k-1)$; j-th column of matrix U(k-1)

$$L(k) = \frac{k_{n-1}}{r_n} \tag{34}$$

$$\hat{P}(k+1) = \hat{P}(k) + \frac{\phi(k-d+1) [c(q^{-1})y(k+1) - \hat{p}^T(k) \phi(k-d+1)]}{1 + \phi(k-d+1)^T \phi(k-d+1)} \tag{35}$$

그러나 위의 알고리즘으로는 초기에 deade-zone을 갖는 공정을 효과적으로 제어할 수 없으므로, 직류 전동기 속도제어를 위하여 식(35)에 smoothing polynomial $S_M(q^{-1})$ 과 상수 ℓ 을 사용한 S_M gradient type 알고리즘을 유도한다.

$$\hat{p}(k+1) = \hat{p}(k) + \frac{\phi(k-d+1) S_M(q^{-1}) E(k)}{\ell + \phi(k-d+1)^T \phi(k-d+1)} \tag{36}$$

$$\text{단, } S_M(q^{-1}) = 1 + S_{M1}q^{-1} + \dots + S_{Mn_{SM}}q^{-n_{SM}}$$

$$E(k) = c(q^{-1}) \hat{y}(k+1) - \hat{p}^T(k) \phi(k-d+1)$$

$$0 < \ell \leq \epsilon \Phi_s^T \Phi_s$$

$S_M(q^{-1})$ 은 asymptotically stable polynomial 이고, Φ_s 는 기준 모델로 공정이 수렴하였을 때의 입출력으로 구성된 벡터이며, 이때 ϵ 는 $\Phi_s^T \Phi_s$ 에 비하여 매우 작은 양의 값이다.

$S_M(q^{-1})$ 은 초기에 기준 모델로 수렴해 갈 때나, 공정의 파라미터가 변하는 경우나, 기준 모델이 바뀌는 경우에 제어를 smooth하게 하며, 상수 ℓ 은 초기에 추정된 파라미터가 급변하는 경우 이를 서서히 참값으로 수렴하게 한다. 이 때 ℓ 의 값은 기준 모델의 크기에 따라서 설정하여 무시할 수 있는 수렴오차내에 있게 한다.

식(36)은 $S_M(q^{-1})$ 이 asymptotically stable polynomial 이고, ℓ 이 $0 < \ell \leq \epsilon \Phi_s^T \Phi_s$ 이므로 식(35)와 마찬가지로 전제어계의 stability[4]를 보장한다.

V. 시뮬레이션 結果

시뮬레이션은 FACOM minicomputer로 하였다.

Model reference는 $Y_M(k+1) = A_M Y_M(k) + B_M R(k)$ 로 하여 $A_M=0.5, B_M=0.5$ 의 계수를 사용하였다.

그리고 제어 대상인 직류전동기는 $Y(k+1) = AY(k) + BU(k)$ 인 일차로 모델링하여 $A=0.3, B=0.8$ 의 계수를 사용하였다.

그림 3은 처음엔 $R(k) = 1000$ 으로 하여 충분히 收斂한 후 $R(k)$ 를 1500으로 바꾼 경우 각 알고리즘에 대한 出力을 나타냈다.

시뮬레이션 결과 E. W. L. S 방식과 UDUT^T 인수화법은 거의 같은 결과로 3 step만에 $R(k) = 1000$ 에 수렴하였고, $R(k) = 1500$ 으로 바뀐 경우에는 바로 수렴하였다.

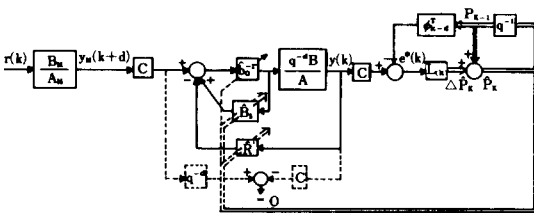


그림 2. 제어 시스템 구조
Fig. 2. Control system structure.

2. SM. Gradient Type 알고리즘

식(19)에서 다음과 같은 gradient type 알고리즘을 유도한다.[4]

그림 4는 model reference에 충분히 수렴한 후 파라미터를 약간 변화시킨 경우의 출력으로 공정의 파라미터가 변하면 수렴 속도가 매우 늦음을 알 수 있다.

이 때 $Y_M(1) = 0, P(0, 1) = P(0, 2) = 1, \lambda_1 = 0.91, \lambda_2 = 0.09, \Gamma(0) = 10$ 으로 하였다.

그림5은 Sm. gradient type의 알고리즘을 이용한 경우로 $k \geq 240$ 에서 파라미터가 약간 변하였고, $k \geq 240$ 에선 $R(k)$ 가 1500으로 바뀐 경우의 시뮬레이션 결과이다.

$Y_M(1) = 0, P(0, 1) = P(0, 2) = 1, S_M(q^{-1}) = 1 - 0.85q^{-1}, \ell = 20000$ 이다.

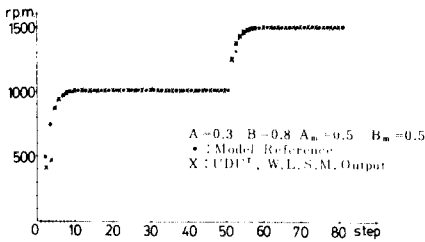


그림 3. Model reference가 변한 경우의 출력
Fig. 3. Output when model reference is varied.

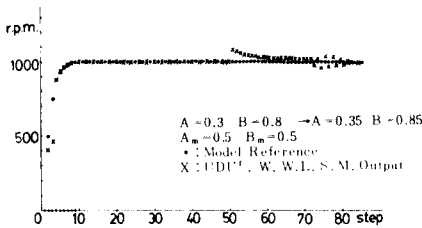


그림 4. 공정의 파라미터가 변한 경우의 출력
Fig. 4. Output when the plant parameter is varied.

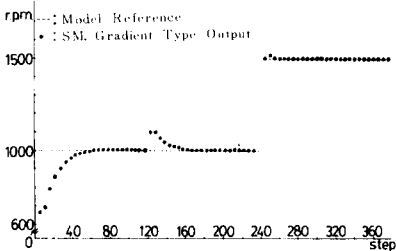


그림 5. Model reference와 공정의 파라미터가 변한 경우의 출력
Fig. 5. Output when model reference parameter and the plant parameter are varied.

시뮬레이션 결과 이 알고리즘은 다른 알고리즘에 비하여 收斂 速度가 낮으나 演算 速度가 빠른 마이크로 컴퓨터를 사용하면 실제 실험에서는 문제가 되지 않는다.

VI. 實驗 및 測定

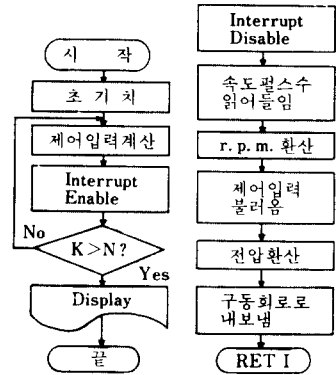


그림 6. 순서도
Fig. 6. Flow chart.

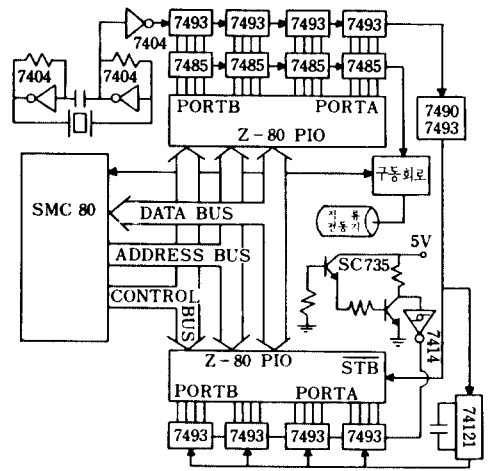


그림 7. 제어시스템 블럭 다이어그램
Fig. 7. Block diagram of control system.

實驗은 직류전동기를 다음과 같은 差分方程式으로 modelling하였고,

$$Y(k) = a_1 Y(k-1) + b_1 U(k-1) \quad (37)$$

마이크로 프로세서 z-80A (clock 주파수 2.5MHz)을 사용하여 그림7의 制御시스템을 구성하였다.

제어 시스템은 전체 제어를 맡는 z-80 C. P. U와

120개의 슬리로 된 직경16.2cm의 원판으로부터 회전 속도를 측정하는 회전 속도 측정부와, 측정된 회전 속도로부터 제어입력을 계산한 후 직류전동기에 인가할 펄스를 만드는 펄스폭 변조기와, 驅動回路로 구성되어 있다.

직류전동기의 회전 속도 측정은 원판에 120개의 슬리를 만들어 光센서와 TTL계수기를 사용하였으며, 회전 속도 제어는 카운터와 디지털 비교기로 구성된 펄스폭 변조기와 다알링턴 트랜지스터로 브릿지형으로 구성되어 단일전원으로 가역운전 할 수 있는 驅動回路로써 직류 초퍼를 만들어 사용하였다. 그리고 모든 연산은 3바이트 floating point 연산을 하였다.

그림 6의 순서도에서 r.p.m 환산계수 C는 원판의 슬리 수를 N, 샘플링 주파수를 f라 하면 C는 다음의 식으로 결정된다.

$$C = \frac{f \times 60}{N} \quad (38)$$

이로부터 순간회전속도를 r.p.m으로 환산할 수 있다. 그리고 알고리즘에 의해 계산된 제어입력은 전동기 구동전압을 65535등분하여 적당한 환산계수를 찾아 전압으로 환산하였고 계산된 제어입력이 음수인 경우는 그 크기의 역 전압을 가하였고, 계산된 제어입력이 65535등분치보다 더 크게 환산되면 등분치의 최대값으로 하였다.

실험은 IV절에 제시한 UDU^T 인수화법 알고리즘과 Sm gradient type의 알고리즘을 적용하여 기준 모델이 바뀌는 경우, 그리고 똑같은 마찰관 부하를 사용

한 경우에 대하여 각각 행하였으며, sampling은 10Hz로 했고, 제어기 계수의 초기치는 임의로 설정하였다.

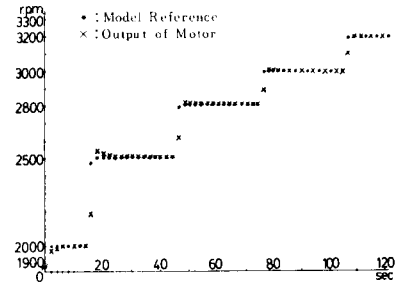


그림 10. UDU^T인수화법에 의한 출력
Fig. 10. Output using UDU^T factorization method.

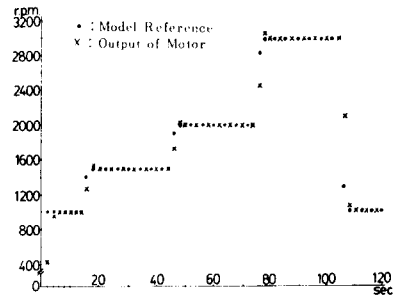


그림 11. SM gradient type의 알고리즘에 의한 출력
Fig. 11. Output using Sm gradient type algorithm.

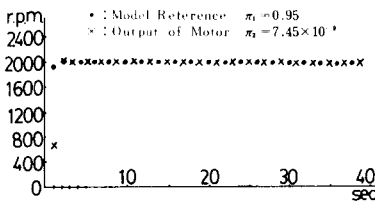


그림 8. UDU^T인수화법에 의한 출력
Fig. 8. Output using UDU^T factorization method.

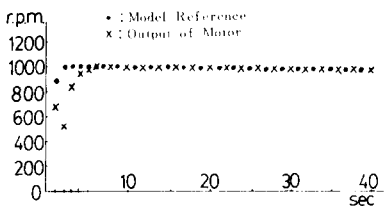


그림 9. SM gradient type의 알고리즘에 의한 출력
Fig. 9. Output using Sm gradient type algorithm.

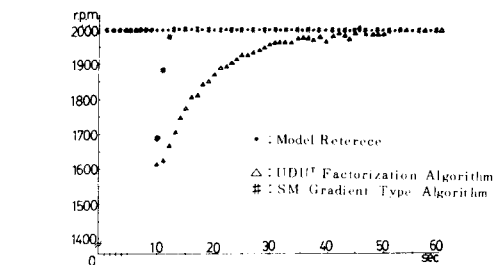


그림 12. 직류 전동기에 부하가 달린 경우의 출력
Fig. 12. Outputs (when load is connected to the D.C. motor).

일반적으로 직류 전동기는 dead zone과 선형부분의 경계에서 파라미터가 급변하므로, 전단계의 정보만으로 새로운 파라미터를 추정하는 gradient type의 알고리즘으로는 dead zone을 벗어 나기 위한 기동전압을 사용하지 않고는 원하는 목적대로 제어할 수 없다. 그

러나 Sm. gradient type의 알고리즘을 사용하므로써 기동전압을 고려하지 않고도 무부하에서 기준모델로 smooth하게 수렴(5~6초)시킬 수 있으며, 직류 전동기에 부하가 걸려 파라미터가 변하는 경우 전단계의 정보로 새로운 파라미터를 추정하므로 기준 모델로의 수렴속도가 (약3초) 빠르다.

UDU^T 인수화법의 알고리즘을 이용한 속도 제어에서는 forgetting factor λ_1 과 λ_2 를 사용하므로써 직류 전동기의 dead zone 특성을 고려하지 않고도 무부하에서 기준 모델로 빨리 수렴(3~4초)시킬 수 있으나, 부하가 걸리는 경우는 과거 정보에 지수 함수적으로 비중을 두어 새로운 파라미터를 추정하므로 기준 모델로의 수렴(약40초)이 늦다.

이상은 V질의 시뮬레이션 결과와 일치한다.

실험에서 허용 수렴 오차는 0.5%로 하였으며, 이는 회전 원판의 slit 수를 더 늘려 속도 측정의 정밀도를 높이면 줄일 수 있다.

Ⅶ. 結 論

본 연구에서는 제시된 UDU^T 인수화법 알고리즘과 Sm. gradient type의 알고리즘을 이용하여 직류 전동기 속도 제어를 하였다.

UDU^T 인수화법 알고리즘을 이용하여 계산 시간과 반올림 오차를 최소로 하여 전체 제어계를 안정화 시켰고, Sm. gradient type의 알고리즘을 이용하여 dead-zone을 갖는 직류 전동기를 기동 전압을 사용하지 않고도 속도 제어를 하였다.

실험 결과 무부하에서 UDU^T 인수화법 알고리즘을 이용했을 경우 3~4초, Sm. gradient type의 알고리즘을 이용했을 때는 5~6초만에 기준모델로 각각 수렴 오차 0.5%이내로 수렴하였고, 부하를 알았을 때는 UDU^T인 수화법의 알고리즘은 약40초, SM gradient type의 알고리즘은 약3초만에 기준모델로 각각 수렴오차 0.5%이내로 수렴하였다.

이상으로 무부하에서는 UDU^T 인수화법의 알고리즘을 사용하면 빠른 시간내에 기준모델로 수렴시킬 수 있으며, 공정에 부하가 걸리는 경우는 Sm gradient type의 알고리즘을 이용하는 것이 바람직하다.

參 考 文 獻

- [1] Y.D. Landau, *Adaptive Control*. Dekker, New York, 1979.
- [2] T. Suzuki, T. Nakamura, M. Koga, "Discrete adaptive observer with fast convergence," *Int. J. Contr.*, vol. 31, no. 6, pp. 1107-1119, 1980.
- [3] 김도현, 김성국 "Direct M. R. A. C에 대한 Weighted Jeast Square Method의 적용에 관한 연구", 대한전자공학회 추계종합학술대회논문집, vol. 4, no. 3, 1981.
- [4] G.C. Goodwin, P.J. Ramadge, P.E. Caines, "Discrete-time multivariable adaptive control," *IEEE Trans. on Automat. Contr.*, vol. AC-25, no. 3, pp. 449-456, June 1980.
- [5] 許旭烈, "내재성 기준모델을 사용한 적응 시스템에 관한 연구", 서울대학교 전기공학과 공학박사 학위 논문, 1982.
- [6] G.J. Bierman, *Factorization Method for Discrete Segquential Estimation*. Academic Press, New York, 1977.
- [7] 金成國, 金道鎰, 崔桂根, "직류전동기 속도제어에 대한 적응제어 이론의 적용에 관한 연구" 대한 전자공학회 추계종합학술대회논문집, vol. 4, no. 3, 1981.
- [8] R. Lozano, I.D. Landau, "Redesign of explicit and implicit discrete time model reference adaptive control schemes," *Int. J. Contr.*, vol. 33, no. 2, pp. 247-268, 1981.
- [9] S.M. Shinnars, *Modern Control System Theory and Application*. Addison-Wesley Publishing Company, 1972.
- [10] B.C. Kuo, *Digital Control Systems*. Holt, Rinehart and Winston, Inc., 1980.
- [11] Bo Egardt, "Unification of some discrete-time adaptive control schemes," *IEEE Trans. on Automat. Contr.*, vol. AC-25, no. 4, pp. 693-697, Aug. 1980.