

RLS 알고리즘과 극점배치방법을 이용한 DC전동기의 자기동조 속도제어기의 구현

차응석, 지준근

순천향대학교 전기전자공학부

Implementation of Self-Tuning Speed Controller for DC Motor Drive System using RLS Algorithm and Pole-Placement Method

Eung-seok Cha, Jun-keun Ji

Department of Electrical & Electronics, Soonchunhyang University

Abstract - This paper describes the design of self-tuning speed controller for DC motor drive system using RLS(Recursive Least Squares) algorithm and Pole-Placement method. The model parameters, related to inertia and damping coefficient of motor, are estimated on-line by using RLS estimation algorithm. And a control signal is calculated by using pole placement method. Simulation and experimental results show that the proposed controller possesses excellent adaptation capability than a conventional PI/IP controller under parameter change.

1. 서 론

산업현장에서의 전동기 구동 시스템에서는 정교한 속도 제어 및 빈번한 속도의 반전이 요구된다. 고정밀 속도 제어에 사용되고 있는 종래의 고정된 이득(gain)의 PI/IP 제어기는 양호한 정상상태 특성을 가지고 있으나 급격한 시스템 파라미터(parameter)의 변동 시에는 만족스런 성능을 주지 못한다. 따라서 산업체에서 요구되는 사항을 만족시키기 위해서는 환경변화에 순응할 수 있는 적응제어이론의 적용이 요구된다.[1][4][5]

자기동조 제어기(Self-Tuning Regulator : STR)는 산업체에서의 이러한 요구 사항들을 만족시킬 수 있는 실현가능하고 적용이 가능한 이상적인 제어기이다. 자기동조 제어기에서는 다음의 두 가지 방법을 통하여 제어 신호를 결정한다. 첫째로, 제어 대상 시스템 모델(Model)의 파라미터들은 시스템의 입출력 데이터로부터 알고리즘을 이용하여 실시간(on-line)계산으로 추정된다. 둘째로, 추정된 파라미터를 가지고 미리 정해진 제어 방식을 사용하여 조건에 부합하는 최적의 제어가 행해진다. 본 논문에서는 RLS(Recursive Least Squares) 알고리즘을 이용하여 파라미터들을 추정하고 극점배치(Pole-Placement)방법을 사용하여 제어기의 이득을 결정한다.

본 논문에서는 PC 인터페이스 시스템과 4상한 PWM Chopper구동 시스템을 사용하여 기존의 PI/IP 속도 제어 알고리즘(Algorithm)을 구현하고, 환경 변화에 따른 기존의 PI/IP 속도 제어기의 이득을 사용자가 조정(Tuning)하는 부담을 제거하면서 조건변화에 따른 파라미터의 변화를 추정하여 제어기의 이득을 자동으로 조정하는 자기동조 제어기를 설계·구현하고자 한다.

2. 본 론

2.1 DC 전동기의 모델링

DC 전동기의 운동방정식은 식 (1)과 같이 전개할 수 있다.

$$K_a I_a(t) = J \frac{dw(t)}{dt} + Bw(t) + T_L(t) \quad (1)$$

J : 관성모멘트(kg · m²)
 B : 점성마찰상수(N · m/rad/s)
 K_a : 토크상수(N · m/A · Wb)
 T_L : 부하토크(N · m)

부하토크(T_L)가 없다고 가정하고 Laplace변환에 의해 다음의 전달함수를 얻을 수 있다.

$$W(s) = \frac{K_a I_a(s)}{Js + B} \quad (2)$$

식 (2)를 ZOH(Zero-Order Hold) 사용하여 Z변환을 하면 속도와 전류에 관한 식 (3)을 구할 수 있다.[4][6]

$$W(z) = \frac{K_a I_a(z)(1 - e^{-T/\tau_m})/B}{z - e^{-T/\tau_m}} = \frac{b_1 I_a(z) z^{-1}}{1 - a_1 z^{-1}} \quad (3)$$

$$a_1 = e^{-T/\tau_m} \quad b_1 = K_a(1 - e^{-T/\tau_m})/B$$

τ_m : 기계적시정수(Mechanical time Constant)
 = (J/B)

T : 샘플링시간(Sampling Time)

2.2 RLS(Recursive Least Squares) 알고리즘 파라미터 a_1 , b_1 을 recursive하게 추정하기 위하여 식(3)을 전개하여 식 (4)를 얻을 수 있다.[4][5]

$$w(k) = a_1 w(k-1) + b_1 I_a(k-1) = \Phi^T(k-1) \theta(k-1) \quad (4)$$

여기서 θ 는 입출력 $[w(k-1) I_a(k-1)]^T$ 를 나타내고 θ 는 추정하려는 $[a_1 b_1]^T$ 이 된다. $y(k) = w(k)$ 는 시스템의 출력이다. 시스템 파라미터의 recursive한 식별(Identification)을 위해서 RLS 알고리즘을 사용한다. 식 (5),(6),(7)은 RLS 알고리즘을 나타내고 있다.[1][2][4][5]

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k-1)[y(k) - \Phi^T(k) \hat{\theta}(k-1)] \quad (5)$$

$$K(k-1) = P(k-1) \Phi(k-1) \quad (6)$$

$$= P(k-1) \Phi(k) [\lambda + \Phi^T(k) P(k-1) \Phi(k)]^{-1}$$

$$P(k) = [I - K(k-1) \Phi^T(k)] P(k-1) / \lambda \quad (7)$$

Covariance 행렬 P의 초기치는 매우 큰 값을 갖는 α 에 의해서 αI 로서 정해진다. 본 논문에서는 $700 \cdot I$ 의 초기치를 사용하였다. λ 는 forgetting factor로서 전형적인 값의 범위는 $0.95 \leq \lambda \leq 1$ 이다.[1][4][5]

2.3 PI/IP 제어기와 극점배치방법

그림 2.1과 2.2는 전류제어기가 이상적이라고 가정했을 때의 PI/IP 속도제어기를 보여주고 있다.[3][4][5]

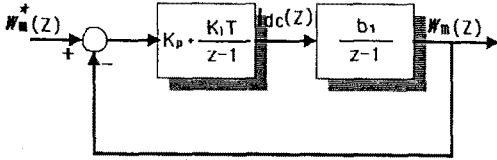


그림 2.1 PI 속도제어기

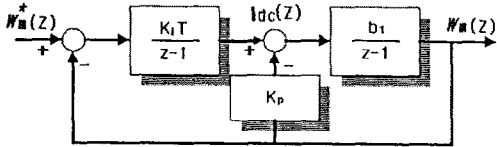


그림 2.1 IP 속도제어기

$$G_{PI}(z) = \frac{W_m(z)}{W_m^*(z)} \quad (8)$$

$$= \frac{b_1 K_p z + b_1 (K_i T - K_p)}{z^2 - (1 + a_1 - b_1 K_p)z + [a_1 + b_1 (K_i T - K_p)]}$$

$$G_{IP}(z) = \frac{W_m(z)}{W_m^*(z)} \quad (9)$$

$$= \frac{b_1 K_i T}{z^2 - (1 + a_1 - b_1 K_p)z + [a_1 + b_1 (K_i T - K_p)]}$$

식 (8), (9)는 PI/IP 속도제어기의 전달함수를 나타내고 있다. 두 식의 특성방정식은 동일하므로 극점배치 방법을 위해 식 (8), (9)의 특성방정식과 원하는 특성방정식 $D(z)$ 를 비교한다. $D(z)$ 의 특성방정식을 쓰면 다음과 같다.[1][4][5]

$$D(z) = (z - p_1)(z - p_2) = z^2 - (p_1 + p_2)z + p_1 p_2 \quad (10)$$

식 (8), (9)와 식 (10)의 계수를 비교하면 각 항에 대한 식 (11), (12)를 얻을 수 있다.

$$p_1 + p_2 = 2e^{-\zeta \omega_n T} \cos(\omega_n T \sqrt{1 - \zeta^2}) = 1 + a_1 - b_1 K_p \quad (11)$$

$$p_1 p_2 = e^{-2\zeta \omega_n T} = a_1 + b_1 (K_i T - K_p) \quad (12)$$

ζ : 감쇠비(damping ratio)

ω_n : 비감쇠 고유주파수

(undamped natural frequency)

위의 식 (11), (12)을 정리하여 다음의 제어신호 이득을 구할 수 있다.

$$K_p = \frac{1 + a_1 - 2e^{-\zeta \omega_n T} \cos \omega_n T \sqrt{1 - \zeta^2}}{b_1} \quad (13)$$

$$K_i = \frac{(e^{-2\zeta \omega_n T} + b_1 K_p - a_1)}{b_1 T} \quad (14)$$

3. 시뮬레이션

본 논문에서 제안한 알고리즘의 성능을 확인하기 위하여 전동기의 모델을 이용하여 시뮬레이션을 하였다.

표 1은 본 논문의 시뮬레이션과 실험에 사용된 전동기의 사양을 나타내고 있다.

망각지수(Forgetting factor)는 1을 사용하였고, 초기의 \hat{a}_1, \hat{b}_1 은 0과 1을 사용하였다. 전동기의 모델을 이용하여 시뮬레이션을 통한 무부하상태의 전동기 파라미터 추정성능을 그림 3.1에서, 그림 3.2는 $J=0.044$ ($\text{kg} \cdot \text{m}^2$) 증가시 파라미터 추정을 보여주고 있다.

표 1 전동기 사양

정격	120V, 175W, 1750rpm, 2.8A
전기자저항(R)	8Ω
전기자인덕턴스(L)	80mH
관성모멘트(J)	0.0025 $\text{Kg} \cdot \text{m}^2$

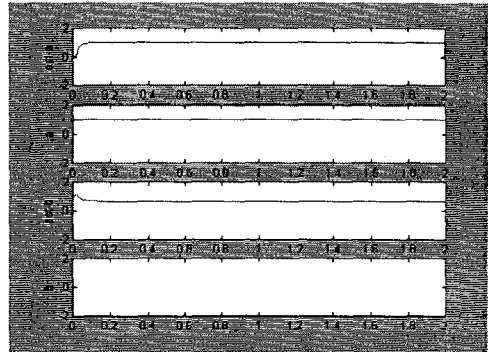


그림 3.1 무부하시 파라미터 추정

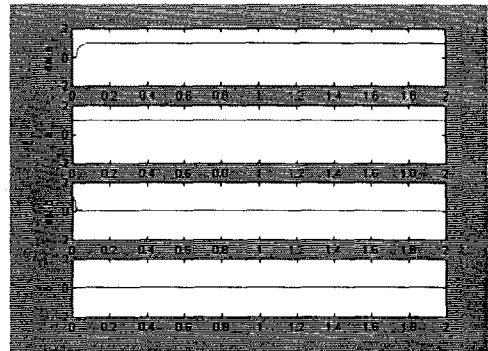


그림 3.2 $J=0.044$ ($\text{kg} \cdot \text{m}^2$) 증가시 파라미터 추정

전동기 사양에 의한 무부하시 a_1, b_1 은 0.9947과 0.6209이고, 추정된 \hat{a}_1, \hat{b}_1 은 각각 0.9947과 0.6197이다. 또, 파라미터 변화시의 a_1, b_1 은 0.9997과 0.0335이고, 추정된 \hat{a}_1, \hat{b}_1 은 각각 0.9997과 0.0335이다.

4. 실험 및 결과

그림 4.1은 실험에 사용한 시스템을 보여주고 있다. PC인터페이스 시스템과 PWM을 이용한 4상한 Chopper를 이용하여 실험을 하였다. 초기 구동시 시스

템의 안정성을 보장하기 위하여 30ms를 기존의 계산에 의한 이득을 사용하고 있다. 전동기의 사양에 의하여 계하여 계산된 a_1 과 b_1 을 이용한 이득은 $K_p = 0.5020$, $K_i = 7.3226$ 이다. 그림 4.2는 전동기 무부하시 500(rpm) \leftrightarrow 500(rpm)으로 구동시의 응답특성을 보여주고 있다.

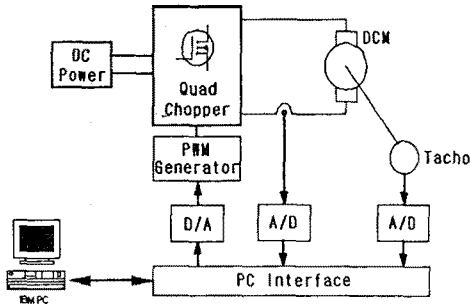


그림 4.1 실험시스템 구성도

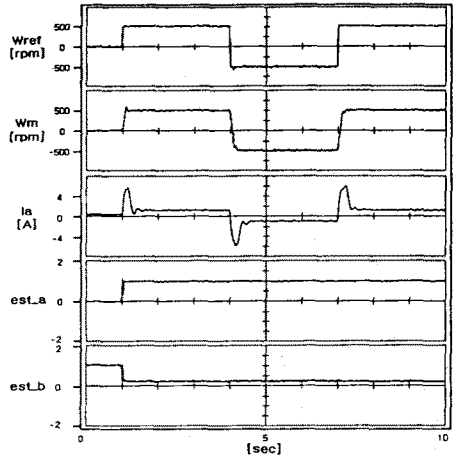


그림 4.4 동기기 부하시

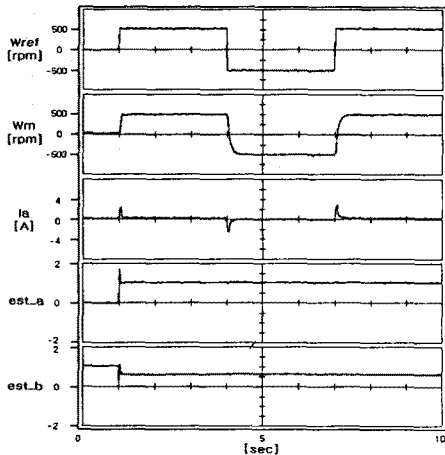


그림 4.2 무부하시 응답특성

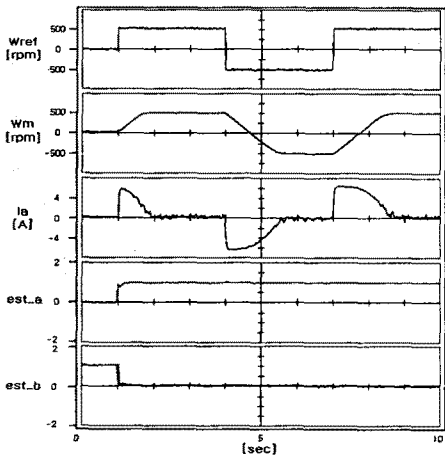


그림 4.3 $J=0.044(\text{kg} \cdot \text{m}^2)$ 증가시

파라미터 추정시간은 약 125ms 정도이며, 망각지수는 1을 사용하였고, $P(0)$ 는 700을 사용하였다. 그림 4.3에서 보여지듯 파라미터를 변화시($J=0.0044(\text{kg} \cdot \text{m}^2)$ 증가)에도 응답추정특성이 양호함을 볼 수 있다. 그림 4.4는 동기기를 키퍼링 하여 나타난 응답특성이다.

3. 결 론

본 논문에서는 RLS 알고리즘과 극점배치방법을 이용한 DC 전동기의 속도제어기의 설계를 제시하였다. 시뮬레이션과 실험을 통하여 얻어진 결과는 전동기 파라미터의 변화($J=0.0025(\text{kg} \cdot \text{m}^2) + J=0.044(\text{kg} \cdot \text{m}^2)$)에 대하여 속도응답의 결과가 만족하게 나타났으며, 외부에 동기기를 결합하여도 안정된 응답과 기존의 PI/IP 속도제어기에 비하여 오버슈트가 감소하는 특성을 보여 주었다.

(참 고 문 헌)

- [1] Karl J. Astrom, Bjorn Wittenmark, "Computer Controlled Systems", Prentice Hall, 416-455, 1990
- [2] Lennart Ljung, "System Identification", Prentice Hall, 303-311, 1987
- [3] K. Nandam, Pradeep, "Analog and Digital Speed Control of DC drive Using Proportional-Integral and Integral-Proportional Control Techniques", IEEE Trans. on, Industrial Electronics, vol. IE-34, No.2, pp 227-233, 1987
- [4] "알연용 대용량 전동기 구동 시스템의 연구개발(II)", 서울대학교 기초전력 연구소(재) 전력전자연구실, 1993
- [5] "전동기 구동시스템 설계 및 실험", 기초전력공학공동연구소 서울대학교 전력전자연구실, 1995
- [6] Jun-Keun Ji, Seung-Ki Sul, "DSP-Based Self-Tuning IP Speed Controller with Load Torque Compensation for Rolling Mill DC Drive", IEEE Trans. on, Industrial Electronics, vol. 42, No.4, pp 382-386, 1995