

# 학습제어기를 이용한 직류전동기제어

홍기철, 남광희 (Kicheol Hong and Kwanghee Nam)

포항공과대학 전자전기공학과 (Dept. of Elec. Eng., POSTECH)

## ABSTRACT

Since the control parameters of classical PID controller are fixed for all control period, it is not easy to produce a desired transition phenomena. We incorporate an iterative learning scheme to the linear controller so that it has more flexibility and adaptation capability especially in the transition period. In this paper a hybrid type learning controller is proposed in which fixed linear controller guides learning at the beginning stage. Once a perfect learning is achieved, then the control action is performed by only the learning controller. A computer simulation result demonstrates better performance during transition time than that with only linear PD controller.

### 1. 서론

특히 요즈음 시스템제어기에 빠른 속도와 정확한 응답특성이 절실히 요구되는 플랜트가 많아졌다. 그러나 기존의 PID 제어기 - 본 논문에서는 고전제어기라고도 칭한다. - 로서는 과도 특성(transient characteristics)을 효과적으로 처리해 줄 수 없기 때문에 따르는 문제들이 있다. 시스템제어의 평가기준으로 흔히 사용되는 계단입력(step input)의 경우를 살펴 보면, 출력특성중 정상상태 응답특성(steady state response)은 고전제어기를 가지고도 충분히 해결이 되나, 과도 응답특성(transient response)은 고전제어기의 경우 제어성능(control performance)이 고정적(fixed)이기 때문에 일정한도이상의 성능의 향상은 불가능하다. 즉 오버슈트(overshoot), 오름시간(rising time), 안정화시간(settling time)등등의 과도현상의 개선이 빠른 속도와 정밀한 제어를 위해서 요구되는데 반하여 고전제어기로서는 과도현상의 개선에 한계가 있다.

최근에 "학습제어"(Learning control)라고 불리는 새로운 제어방법이 관심을 끌고 있다. 이 학습제어방법은 불확실한 다이내믹스를 갖는 시스템에 대해서도 간단한방법으로 제어가 가능하다는데 큰 장점이 있다. 또한 학습율(Learning Rule)이 간단하여서 많은 계산을 필요로 하지 않는다. 학습율은 Uchiyama[1]에 의해서 소개되었고 Arimoto[2]에 의해서 좀 더 정형화되었다. 학습제어율이란 원하는 궤적을 오차없이 따라가도록 보정제어입력(Compensating control input)을 오차를 이용하여 만들어내는 알고리즘이다. Arimoto가 제안한 방법은 다음과 같다.

$$U^{i+1}(k) = U^i(k) + \alpha \cdot e^i(k)$$

여기서  $e^i(k)$ 는 목표출력값(desired output)과 실제 출력값(plant output) 사이의 오차이다.  $i$ 는 반복횟수이며  $k$ 는 샘플링시간(sampling time)을 나타내는 스텝 수(step number)이고  $\alpha$ 는 상수이다.  $U^i(k)$ 는 플랜트에 가해질 제어입력이다. 위에서 제안한 학습율만을 이용한 제어방법은 매우 간단하기는 하지만 전체시스템의 안정도(Stability)를 보장하지는 못한다. 왜냐하면  $e^i(k)$ 가 매우 크거나  $\alpha$ 가 잘 설정되지 않으면 반복학습을 통해서 나온 제어입력이 전체시스템을 불안정영역으로 몰 수도 있기 때문이다. 따라서 제어성능을 높이고 안정도를 확실하게 보장하기위한 방법으로 고전제어기와 학습제어기를 혼용하는 혼성제어기의 형태가 바람직하다. 이렇게 하면 고전제어기는 전체시스템의 안정도문제와 기본적인 제어입력부분을 담당하고, 학습제어기는 고전제어기가 담당하지 못하는 부분을 담당하는 형태가 된다. 즉 대부분의 궤적은 고전제어기에서 나오는 플랜트입력으로 담당하고, 고전제어기가 해결하지 못하는 부분은 학습제어기를 이용하여 해결하는 형태가 된다. 결국 학습제어기와 고전제어기의 혼용을 통하여 임의의 원하는 궤적을 따라가게 할 수 있을 뿐만아니라 시스템의 비모델화된 오차(unmodeled error)도 해결할 수 있다.

본 논문에서는 고전제어기와 학습제어기를 혼용하여 직류전동기의 과도응답을 개선하는 학습율을 제안하였고 이를 모의실험(simulation)을 통하여 확인하였다.

## 2. 학습이론과 시스템구성

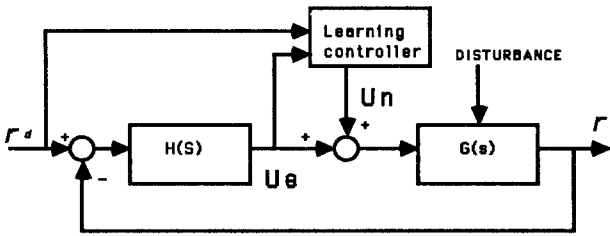


그림 1. 학습제어기를 부착시킨 시스템 구성도

$r_d(k)$  : 기준 입력  
 $r(k)$  : 출력  
 $U_0(k)$  : 고전제어기의 출력  
 $U_n(k)$  : 학습제어기의 출력  
 $U_p(k)$  : 플랜트에 가해질 입력, 즉  $U_n(k) + U_0(k)$   
 $U_d(k)$  : 이상 플랜트 입력 (ideal plant input)  
 $G(s)$  : 플랜트의 전달함수  
 $H(s)$  : 고전제어기의 전달함수

$i$  : iteration number  
 $k$  : step number  
 $\beta$  : learning rate

일반적으로 플랜트를 정확히 수학적으로 기술하기가 어렵기 때문에 어떤 임의의 궤적  $r_d$  를 따라가는 이상 플랜트 입력  $U_d(k)$  을 정확히 알 수는 없다. 실사 플랜트를 정확히 모델링(modeling)하였다 하더라도  $U_d(k)$  를 계산하기는 어렵다. 특히 시간이 충분히 흐르지 않은 상태인 과도시간(transient time) 동안에는 출력의 응답이 기준입력과 상당히 다르다. 따라서 이러한 과도응답을 개선하기 위해서 고전제어기에서 나오는 오차  $U_0(k)$  를 학습제어기의 입력으로 넣고 학습이론을 이용하여 플랜트의 이상입력(ideal input)  $U_d(k)$  를 만들어 내도록 학습을 시키는 것이다. 이를 위해 그림 1과 같이 학습제어기를 고전제어기와 병용하여 시스템을 구축할 수 있다. 여기서 학습율(learning rule)은 다음과 같은 방법으로 유도된다. 학습제어기의 출력  $U_n(k)$  를 만들어내는 지표함수(index function)를 매 샘플포인트  $k$  에서 식 (1)과 같이 정한다.

$$J(k) = \frac{1}{2} \| U_d(k) - U_n^i(k) \|^2 \quad (1)$$

그리고 경사율(gradient decent rule)을 식 (2)와 같이 적용한다.

$$U_n^{i+1}(k) = U_n^i(k) - \beta \cdot \left\{ J(k) / \partial U_n^i(k) \right\} \quad (2)$$

$$U_n^{i+1}(k) = U_n^i(k) + \beta \cdot [U_d(k) - U_n^i(k)] \quad (3)$$

식 (2)는 에너지함수(energy function)로서 학습제어기에서 나오는 플랜트입력이 이상 플랜트입력을 따라가게 하도록 학습율을 구성하기 위한 함수이다. 식 (2)에 식 (1)을 대입하여 식 (3)을 얻는다. 학습율 (3)을 이용해서 반복학습횟수를 증가시키면  $U_n(k)$  는  $U_d(k)$  를 따라가게 된다. 물론  $\beta$  값의 설정이 중요하다. 그러나 식 (1)의 지표함수가 수렴할 수 있도록 국부경계(local boundary) 안에서 작은  $\beta$  값을 설정하면 지표함수는 수렴할 수 있다. 그런데 문제는  $U_d(k)$  가 미지이기 때문에 학습율 (3)을 실질적으로 쓸 수가 없다.

## 3. 2차 시스템의 학습제어기 구성도

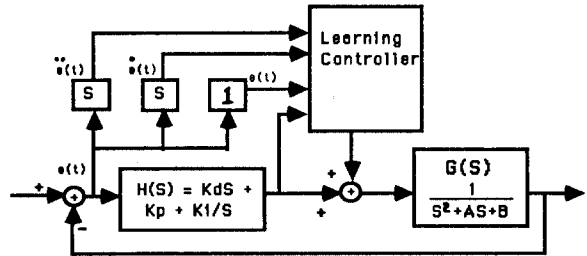


그림 2. 학습제어기가 있는 2차 시스템

그림 2의 시스템에서 플랜트의 전달함수는 식 (4), 제어기의 전달함수는 식 (5)와 같다.

$$G(s) = 1 / (s^2 + A s + B) \quad (4)$$

$$H(s) = (K_p + K_d s) U_0(s) \quad (5)$$

2차 시스템의 경우는  $[U_d(k) - U_n^i(k)]$  를 궤적오차의 값, 궤적오차의 1차 미분값, 궤적오차의 2차 미분값을 이용하여 근사식으로 구할 수 있고 이들 식 (4)와 식 (5)에 적용하면 식 (6)과 같다.

$$U_n^{i+1}(k) = U_n^i(k) + \beta \cdot [ \ddot{e}(k) + (A + K_d) \dot{e}(k) + (B + K_p) e(k) ] \quad (6)$$

## 4. 학습제어기의 직류전동기에의 응용

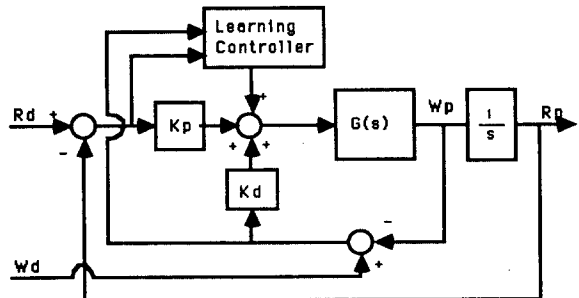


그림 3. 학습제어기가 부착된 전동기시스템 구성도

전동기의 전달함수와 PD제어기의 전달함수를 쓰면 식 (7)과 식 (8) 와 같다.

$$G(s) = \frac{K_t}{L_a J_m s^2 + (R_a J_m + B_m L_a) s + (K_b K_t + R_a B_m)} \quad (7)$$

$$H(s) = K_p + K_d s^{-1} \quad (8)$$

- $R_a$  : armature resistance
- $L_a$  : armature inductance
- $J_m$  : rotor inertia of motor
- $B_m$  : viscous friction coefficient
- $K_b$  : back emf constant
- $K_t$  : torque constant
- $K_p$  : proportional gain
- $K_d$  : derivative gain
- $W_d$  : desired angle velocity
- $W_p$  : plant angle velocity
- $\theta_d$  : desired position
- $\theta_p$  : plant position

전동기 모의실험(simulation)에서는 위치오차와 각속도오차를 이용하여 기준입력의 위치와 각속도를 따라가도록 하는 지표함수를 세웠다. 지표함수는 식 (9) 과 같다.

$$J(k) = \frac{1}{2} \| (W_d - W_p)^2 + (\theta_d - \theta_p)^2 \|^2 \quad (9)$$

식 (9)의 지표함수를 이용하여 학습율을 만들면 식 (10)과 같다.

$$U_n^{i+1}(k) = U_n^i(k) + [\partial W_p / \partial U_n \cdot (W_d - W_p) + \partial P_p / \partial U_n \cdot (\theta_d - \theta_p)] \quad (10)$$

## 5. 모의실험(Simulation)결과 및 토의

식 (10)를 이용하여 학습제어기의 출력을 만들어 이를 모의실험한 결과가 그림 4.1부터 그림 4.8에 나타나 있다. 이 모의실험에서 사용된 직류전동기의 전달함수식에 필요한 상수와 PD제어기의 이득값(gain value)은 다음과 같다.  $R_a$ 는 7.56  $\Omega$ ,  $L_a$ 는 0.055 H,  $K_t$ 는 4.23 V/rad·sec<sup>-1</sup>,  $K_b$ 는 4.23 Nm/rad,  $J_m$ 은 0.136 Nm·sec<sup>2</sup>,  $B_m$ 은 0.273 Nm/rad·sec<sup>-1</sup>이며, PD이득  $K_p$ 는 10,  $K_d$ 는 0.5 이다. 이 모의실험은 VAX-8800에서 FORTRAN을 사용하여 실행하였다.

과도특성의 개선은 실제 시스템성능에 크게 연관된다. 특히 빠른 속도를 가지고 짧은 시간안에 정밀한 반복작업을 하도록 요구되는 플랜트의 경우 - 예를 들면 NC Machine - 에는 과도특성의 문제로 인하여 작업

조건에 제한이 많이 따랐다. 시스템이 과소제동(under damping)으로 인한 오버슈트가 발생하면 안되는 작업의 경우, 대체로 시스템응답의 안정을 위하여 과제동(overdamping)을 주어서 시스템을 제어하게 된다. 그러나 과제동을 줄 경우 시스템응답이 늦어서 안정화시간(settling time)이 길어지게 되고 작업속도는 그만큼 떨어진다. 그림 4.1과 그림 4.2는 학습제어기가 없는 순수한 PD제어기만 가지고 목표위치(desired position)와 목표각속도(desired angle velocity)를 제어하였을 때의 결과이다.

그림 4.1에서 보듯이 시스템응답이 과제동(overdamping)되어 있는데 실제로 로봇(robot)나 여러 시스템에서 오버슈트의 영향때문에 시스템제어에 과제동을 주어서 쓰고 있다. 이런 경우 학습제어기를 통하여 과도특성을 개선한 제어는 학습된 궤적에 대해서는 과제동으로 인한 안정화시간의 지연을 막아서 시스템응답이 빨라지므로 작업속도는 대단히 증가된다. 그리고 오버슈트의 발생을 없앴으로써 작업범위(work range)도 훨씬 융통성을 갖게 된다. 또한 플랜트가 수학적으로 완벽히 기술되지 않아서 발생하는 모델링오차(modeling error)나 동적오차(dynamic error)를 학습을 통하여 반영시킴으로써 크게 줄일 수 있다.

그림 4.3부터 그림 4.6까지는 학습횟수에 따라 시스템응답이 개선되어서 목표궤적에 거의 일치함을 볼 수 있다.

그림 4.7은 반복학습의 횟수가 증가함에 따라 학습제어기에서 나오는 플랜트입력이 PD제어기에서 나오는 플랜트입력보다 커짐을 보여주고 있다. 그림 4.8은 PD제어기에서 나오는 플랜트입력이 그림 4.7과 반대로 학습횟수가 증가함에 따라 감소함을 보여준다. 이 과정은 즉 학습초기에는 PD제어기가 대부분의 제어입력을 담당하다가 학습을 함에 따라 필요한 제어입력을 학습제어기에게 이양(transfer)하는 과정으로 볼 수 있다.

## 6. 결론

모의실험의 결과로 식 (10)에서 제안된 학습율으로써 과도특성이 현저히 개선되었음을 볼 수 있다. 이는 곧 임의의 목표궤적(desired trajectory)에 대해서도 잘 따라가도록 하는 제어입력을 만들 수 있다는 것이 된다.

또한 본 논문에서 제안된 학습제어기와 고전제어기를 혼성한 형태는 학습제어기만으로 플랜트를 제어할 경우보다 전체시스템의 안정도(stability)에 기여한다. 왜냐하면 이러한 혼성제어기(Hybrid controller)의 형태는 PD제어기와 같은 고전제어기가 전체시스템의 안정도를 보장하고 있기 때문이다.

앞으로의 연구방향은 일반적인 전달함수를 갖는 플랜트에 대해서도 잘 적용이 되는 학습율을 고안하고, 또한 학습속도가 좀 더 빠르도록 제안되어야 하겠다. 이를 위한 한가지 방법으로 신경회로망(Neural Network)을 이용한 학습제어기가 연구되고 있다.

## 7. 참고문헌

- (1) UCHIYAMA, M. : "Formulation of High-Speed Motion Pattern of a Mechanical Arm by Trial" , Trans. Society of Instrument and Control Engineers of Japan, 1978, 14, (6)
- (2) ARIMOTO, S., KAWAMURA, S., and MIYAZAKI, F. : "Bettering Operation of Robots by Learning". Journal of Robotic Systems, 1984, 1, (2)
- (3) KAWAMURA, S., MIYAZAKI, F., and ARIMOTO, S. : "Realization of Robot Motion Based on a Learning method", IEEE Trans. on Systems, Man, and Cybernetics, 1988, 18, (1)
- (4) Kwanghee Nam and Tae-Young Kuc, "An Application of the CMAC to Robot Control", Proc. of the Korean Automatic Control Conference Vol.2, 1988.
- (5) 허경무 : "A Second-Order Iterative Learning Control Method" 대우기보 1989.4
- (6) B. C. Kuo, "Automatic Control Systems," 3rd ed., Prentice-Hall, Englewood Cliffs, N.J., 1975

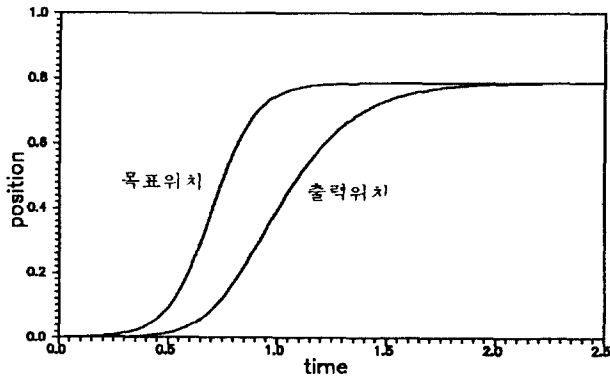


그림 [4.1] PD 제어기만 있을 경우의 출력위치

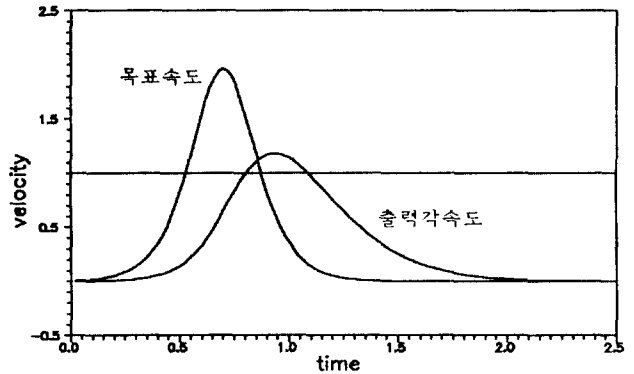


그림 [4.2] PD 제어기만 있을 경우의 출력각속도

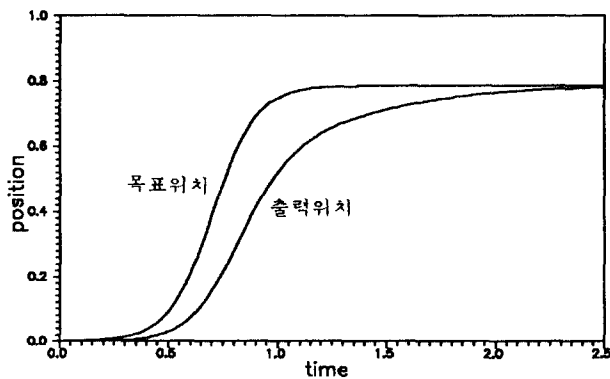


그림 [4.3] 반복학습횟수가 5번일 때의 출력위치

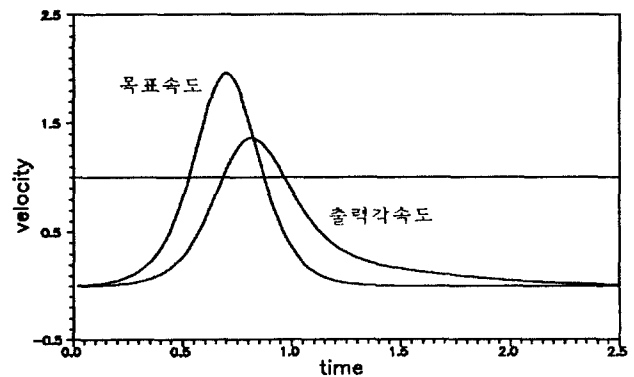


그림 [4.4] 반복학습횟수가 5번일 때의 출력각속도

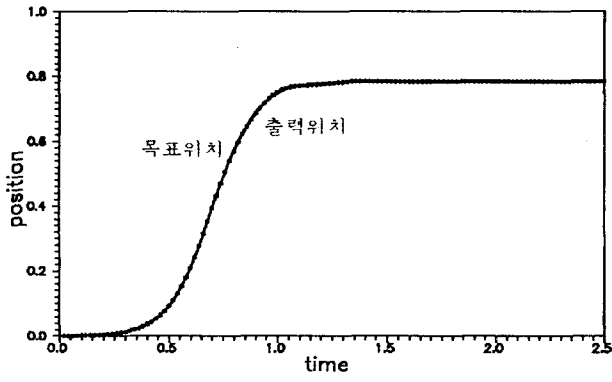


그림 [4.5] 반복학습횟수가 100번일 때의 출력위치

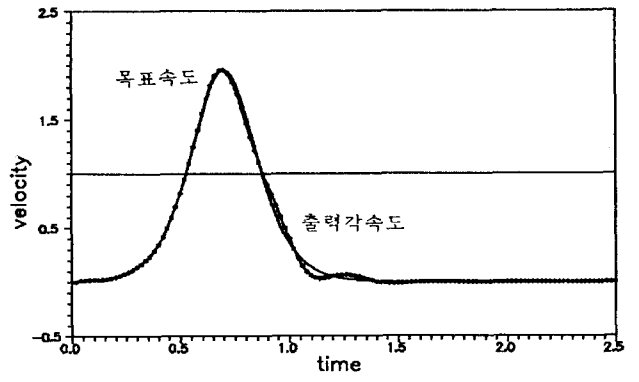


그림 [4.6] 반복학습횟수가 100번일 때의 출력각속도

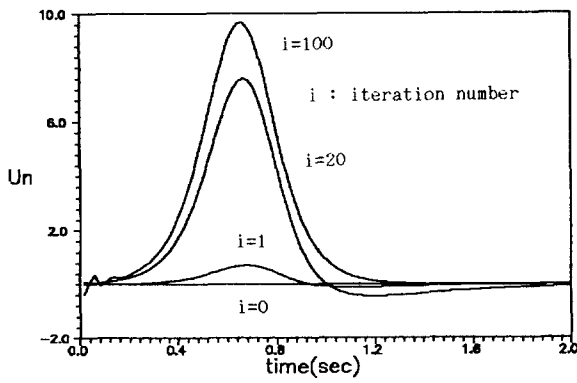


그림 [4.7] 반복학습횟수에 따른  $U_n(k)$

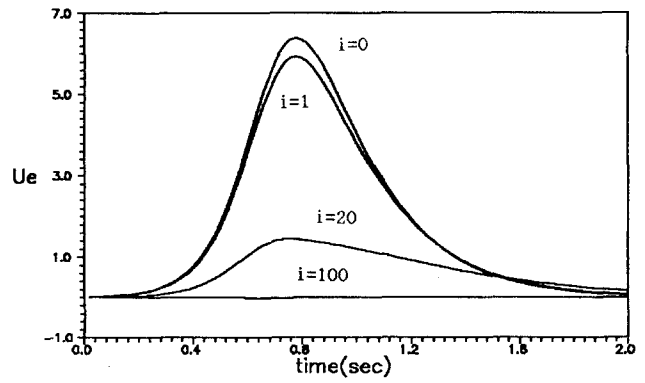


그림 [4.8] 반복학습횟수에 따른  $U_e(k)$