

기울기법을 이용한 최적의 PID 제어 학습법

임윤규*, 정병목**

PID Learning Method using Gradient Approach for Optimal Control

Yoon Kyu Lim*, and Byeong Mook Chung**

ABSTRACT

PID control is widely used in industrial areas, but it is not easy to tune PID gains for an optimal control. The proposed learning method is to tune PID gains using the gradient approach. We use two estimation functions in this method : one is an error function for tuning of PID gains, and the other is a performance measuring function for a completion of learning. This paper shows that optimal PID controllers can be acquired when this learning method is applied to 10 systems with different natural frequencies and damping ratios.

Key Words : PID learning(PID 학습), PID control(PID 제어), Gradient approach(기울기법)

1. 서론

자동제어 이론이 나온 이후 오랜 기간 사용되어 온 PID 제어기는 현재에도 산업 현장에서 가장 널리 사용되고 있는 제어기이다. PID 제어기는 구조가 간단하여 작업자와 제어기 설계자에게 익숙하고 신뢰성도 있다. 그러나 PID 제어 게인을 최적적으로 조정하는 문제는 아직 까지도 많은 분야에서 연구되고 있다. Ziegler와 Nichols는 단위 계단 입력에 대한 플랜트의 응답을 보면서 변수를 구하는 방법과 비례게인의 증가에 의해 처음으로 진동이 발생하였을 때의 제어 게인과 주파수에 기초를 둔 방법을 제시하였다^[1]. 이 방법은 플랜트의 수학적 모델을 알 수 없을 때 편리하게 사용할 수 있지만 제어 출력이 진동을 유지하기 어려울 때는 사용할 수 없는 단점이 있다. Astrom과 Hagglund는 Nyquist선도에서 임계 게인과 임계 주파수를

지연 되먹임(delay feedback)을 이용하는 방법을 제시하였다^[2,3]. 이 방법은 잡음에 대해 신뢰성이 높지만 함수를 근사화 하였기 때문에 불감 시간(dead-time)이 긴 경우에는 임계점을 정확히 추정할 수 없다.

근래에는 신경망 이론, 퍼지 이론, 유전자 이론과 같은 인공 지능 이론을 이용하여 최적의 PID 게인을 자동 조정(Auto-tuning)하는 방법에 대한 연구가 활발하다^[4-6]. 이 방법들은 많은 메모리를 차지할 뿐 아니라 학습에 소요되는 시간이 많이 걸리기 때문에 On-Line 학습이 어렵다. 본 논문에서는 시스템의 수학적 모델링을 구하기 힘든 현장에 쉽게 적용할 수 있으며, 학습 소요 시간이 적게 걸리기 때문에 실시간 학습이 가능한 PID 학습법을 제안하고자 한다. 제안한 학습법은

* 영남대학교 대학원

** 영남대학교 기계공학부

PID 제어기의 P 게인, I 게인 그리고 D 게인을 자동으로 조정하기 위하여 기율기법을 이용하였다. 각 게인을 조정하는 방법은 플랜트의 원하는 출력과 실제 출력의 차인 오차를 비용함수로 정의하고 이를 최소화하는 방향으로 반복적으로 조정한다. 학습 평가 함수를 제어기의 직접적인 성능, 즉 상승 시간, 정착 시간, 피크 시간과 오버슈트의 합으로 정의하고 학습 성능을 학습 전의 평가 함수와 학습 후의 평가 함수로 정의하여 이 값이 최소가 되는 시점을 학습의 종료로 하였다. 그리고 최적의 학습을 수행하기 위해 학습 응답, 학습 주기, 최대 입력을 학습 조건으로 제시하였다. 이 학습 조건들은 학습에 소요되는 시간을 줄이고 구동기의 사양을 최소화하면서 최적의 PID 제어를 설계할 수 있도록 한다. 제안한 PID 학습법과 학습 조건을 이용하여 10개의 2차 진동 시스템의 레귤레이션(regulation) 문제에 적용하였다. 2차 진동 시스템의 고유진동수에 따라 고주파와 저주파로 나누고 각 주파수에서 감쇠비에 따라 5단계로 나누어 10개의 시스템을 선정하였다.

2. 시스템 선정

본 논문에서 제안한 PID 학습법을 다음과 같은 2차 진동 시스템에 적용하였다.

$$m\ddot{x}(t) + b\dot{x}(t) + kx(t) = u(k) \quad (1)$$

여기서 m 은 시스템의 질량, b 는 시스템의 댐핑 계수, k 는 스프링 상수이다. 위 식은 식(2)와 같이 나타낼 수 있다.

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = u(k)/m \quad (2)$$

여기서 ζ 는 감쇠비, ω_n 은 고유진동수이고, 그 값은

$$\omega_n = \sqrt{\frac{k}{m}}, \quad \zeta = \frac{b}{2m\omega_n} \text{이다.}$$

시스템을 제어하기 위한 샘플링 주파수는 시스템의 고유진동수 보다 커야 하고 샘플링 주기는 시스템의 주기 보다 작아야 한다. 이를 식(3)과 같이 나타낼 수 있다.

$$T_s = \frac{2\pi}{\omega_s} \leq T_n = \frac{2\pi}{\omega_n} \quad (3)$$

여기서 T_s 와 ω_s 는 샘플링 시간과 샘플링 주파수이고, T_n 와 ω_n 은 시스템 주기와 시스템 고유진동수이다. 샘플링 시간과 고유 진동수와의 관계를 식(4)와 같이 나타낼 수 있다.

$$\omega_n \leq \frac{2\pi}{T_s} \quad (4)$$

일반적인 구동기의 최소 구동 시간은 0.01[sec]이고, 샘플링 시간은 시스템의 고유진동수의 20배 이상을 선정한다. 시스템의 고유 진동수를 구하는 식은 식(5)와 같이 나타낼 수 있다.

$$\omega_n \leq \frac{2\pi}{N \times 0.01} \text{ [rad/sec]} \quad (5)$$

여기서 N 은 고유 진동수와 곱해지는 값으로 상수이다. 식(5)를 이용하여 N 이 20일 때 고유 진동수의 최대 값(고주파)은 31.416[rad/sec]이고, 저주파는 N 이 200일 때의 고유진동수 3.142[rad/sec]를 선정하였다. 그리고 식(2)의 감쇠비는 각 주파수마다 $\zeta=[0.00, 0.25, 0.50, 0.75, 1.00]$ 과 같이 5단계로 나누어 10개의 시스템을 선정하였다. 이를 Table 1에 나타내었다.

Table 1 Values of system parameter.

system	A	B	C	D	E	F	G	H	I	J
m [kg]	1.0					1.0				
ζ	0.00	0.25	0.50	0.75	1.00	0.00	0.25	0.50	0.75	1.00
N	20					200				
ω_n [rad/s]	31.4					3.14				

3. PID 학습법 및 학습조건

본 논문에서 제안한 PID 학습법은 목적함수를 정의하고 이를 최소화하는 방향으로 각각의 PID 계인을 학습한다. 학습의 종료시점은 이장에서 정의하는 학습 성능이 최고점에서 학습을 종료한다. 그리고 최적의 학습을 하기 위해서 최대 입력, 학습 주기, 학습 응답을 조건으로 하여 최소의 시간과 입력을 사용하도록 학습하였다.

시스템의 출력은 식(6)과 같은 비선형 차분 방정식으로 표현할 수 있다.

$$y(k) = g[y(k-1), \dots, y(k-p), u(k-n), \dots, u(k-m)] \quad (6)$$

여기서, n 은 시스템의 시간지연, m 은 시스템 입력 차수, p 는 시스템 차수, $y(k-p)$ 는 p 스텝 전의 시스템 출력이다. 식(6)을 제어 입력 $u(k)$ 에 대한 식으로 나타내면 식(7)과 같이 표현할 수 있다.

$$u(k) = f[y_d(k+n), \dots, y_d(k+1), y(k), \dots, y(k-p+n), u(k-1), \dots, u(k-m-n)] \quad (7)$$

여기서 $y_d(k+n)$ 은 n 스텝 후의 원하는 시스템 출력이다. 식(7)의 제어 입력을 변수의 오차, 오차 변화율과 오차의 합에 대한 함수로 나타낼 수 있다.

$$u(k) = F[e(k), \dot{e}(k), \int_0^k e(t) dt] \quad (8)$$

여기서, 오차는 $e(k) = y_d(k) - y(k)$, 오차 변화율은 $\dot{e}(k) = \frac{e(k) - e(k-1)}{T_s}$, 오차의 합은 $\int_0^k e(t) dt = \sum_{t=0}^k e(t) T_s$ 이다. 식(8)을 PID 제어를 이용하여 제어 입력을 표현하면 식(9)와 같이 표현할 수 있다.

$$u(k) = K_p e(k) + K_d \dot{e}(k) + K_i \int_0^k e(t) dt \quad (9)$$

여기서 K_p, K_i, K_d 는 PID 제어 상수이다. Fig.1은 PID 제어에 대한 블록 선도를 나타낸 것이다.

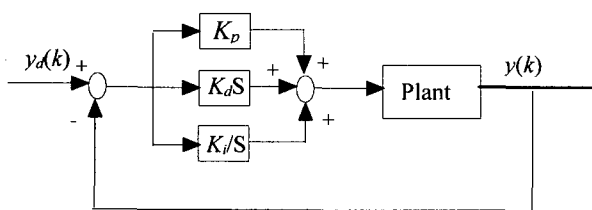


Fig. 1 Block diagram of PID control.

식(9)에서 보듯이 제어 입력은 제어 상수인 PID 계인 값에 의해 좌우된다. 이 제어기를 자동적으로 조정하기 위해서는 식(10)과 같이 목적함수를 정의하였다.

$$J(k) = \frac{1}{2} e(k)^2 \quad (10)$$

최적의 PID 제어기를 설계하기 위해서 식(10)의 목적함수를 최소화해야 한다. PID 각각의 계인에 대해 목적함수를 최소화하는 기물기법을 사용하여 식(11)-(13)에 나타내었다.

$$\Delta K_p \propto - \frac{\partial J(k)}{\partial K_p} \quad (11)$$

$$\Delta K_d \propto - \frac{\partial J(k)}{\partial K_d} \quad (12)$$

$$\Delta K_i \propto - \frac{\partial J(k)}{\partial K_i} \quad (13)$$

이식은 목적함수를 최소화하기 위하여 각각의 PID 계인을 학습하는 방향을 의미한다. 이것을 정리하면 식(14)-(16)과 같이 나타낼 수 있다.

$$\Delta K_p \propto e^2(k) \frac{\partial y(k)}{\partial u(k)} \quad (14)$$

$$\Delta K_d \propto e(k) \dot{e}(k) \frac{\partial y(k)}{\partial u(k)} \quad (15)$$

$$\Delta K_i \propto e(k) \int_0^k e(t) dt \frac{\partial y(k)}{\partial u(k)} \quad (16)$$

식(14)-(16)에 포함된 시스템 출력에 대한 입력의 편미분 $\frac{\partial y(k)}{\partial u(k)}$ 값은 시스템이 단조 증가 또는 단조 감소 함수라면 부호가 변하지 않으므로 같은 부호의 상수로 두고 학습하여도 수렴한다^[6-10]. 이 시스템은 단조 증가 시스템이므로 다음과 같이 표현할 수 있다.

$$\Delta K_p = \eta e^2(k) \quad (17)$$

$$\Delta K_d = \eta e(k) \dot{e}(k) \quad (18)$$

$$\Delta K_i = \eta \int_0^k e(t) dt \quad (19)$$

여기서 η 는 PID 계인에 대한 학습율이다. 학습 탐색 간격을 크게 하면 최적의 PID 계인을 찾기 위한 탐색 간격이 커져서 국소점(local minimum)에 빠지기 쉬우나 학습 시간이 짧은 장점이 있다. 반대로 학습 탐색 간격을 작게 하면 탐색 간격이 작아져 최적의 PID 계인을 찾을 수 있으나 학습 시간이 많이 걸린다는 단점이 있다. 그러므로 적당한 값을 선정하는 것이 중요하다. 식(17)-(19)에

의하여 PID 제어 계인을 수정하면 새로운 PID 제어 계인을 식(20)-(22)와 같이 구할 수 있다.

$$K_p(k+1) = K_p(k) + \Delta K_p \quad (20)$$

$$K_d(k+1) = K_d(k) + \Delta K_d \quad (21)$$

$$K_i(k+1) = K_i(k) + \Delta K_i \quad (22)$$

일반적으로 목적 함수를 정의하여 이 값의 최소 값을 찾는 것으로 종료된다. 그러나 제안한 PID 학습법에서는 목적 함수를 이용하여 학습 방향을 결정하고 매주기마다 학습이 진행되는 정도를 검사하여 학습 진행 정도가 최대일 때 학습을 종료한다. 학습 진행 정도를 판단하기 위해서는 제어기의 성능을 평가하는 평가 기준이 있어야 한다. 따라서 본 논문에서는 시스템의 응답성(상승 시간, 정착 시간, 피크 시간, 오버슈트)의 합을 평가 함수 값으로 정의하고 이 값이 작을 수록 제어기의 성능 좋음을 의미한다. 이를 식(23)에 나타내었다.

$$Est = w_r t_r + w_s t_s + w_p t_p + w_o os \quad (23)$$

여기서 Est 는 평가 함수 값, t_r 은 상승 시간, t_s 는 정착 시간, t_p 는 피크 시간, os 는 오버슈트이다. 그리고 w_r, w_s, w_p, w_o 는 가중치이다. 학습 전 시스템의 초기 응답 ($[x, \dot{x}] = [0.5m, 0m]$)을 Fig.2와 Fig.3에 나타내었다. 시스템의 응답을 관찰하는 시간을 6초로 하였다. 따라서 오버슈트의 최대 값은 1이고 나머지의 최대 값이 6이다. 평가함수의 가중치를 w_r, w_s, w_p 는 1, w_o 은 6으로 하였다. 이때의 평가 함수 값을 Table 2에 나타내었다. Table 2의 평가 함수 값에 따라 시스템의 성능을 비교하면 "D-E-C-B-I-H-J-G-A-F" 순서로 성능이 좋음을 알 수 있다.

Table 2 Initial estimation value.

System	A	B	C	D	E	F	G	H	I	J
Est	7.12	1.13	0.61	0.46	0.54	8.45	7.51	4.78	4.52	5.45

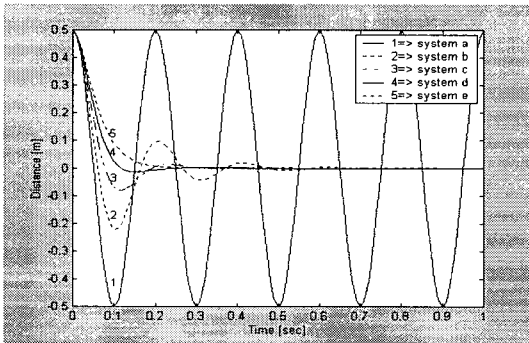


Fig. 2 Response of system A~E before learning.

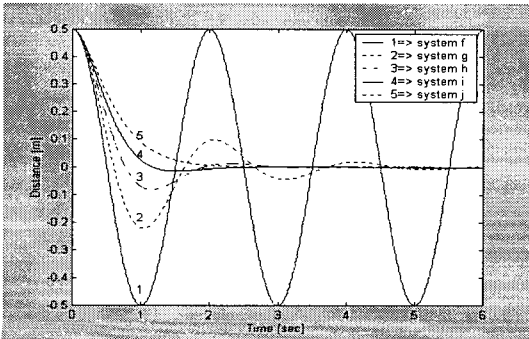


Fig. 3 Response of system F~J before learning.

식(23)을 이용하여 학습이 진행되는 정도를 나타내기 위해 식(24)와 같이 학습 성능을 정의하였다.

$$Lp(h) = \frac{Est(0) - Est(h)}{Est(0)} \times 100 \text{ [%]} \quad (24)$$

여기서 $Lp(h)$ 는 h 번 학습했을 때 학습 성능, $Est(0)$ 는 학습전의 평가함수 값, $Est(h)$ 는 h 번 학습 후 평가함수 값이다. 이 학습 성능의 값이 크면 클수록 학습이 잘 이루어졌음을 의미한다. 매 학습 주기마다 학습 성능을 검사하여 이 값이 최대가 될 때 학습을 종료한다.

본 논문에서는 최적의 제어기를 학습하기 위해서 다음과 같이 3가지 학습조건을 제시한다. 첫째는 최대 입력, 둘째는 학습 주기, 셋째는 학습 응답이다. 첫째 조건인 최대 입력은 입력의 최대 값을 의미한다. 이 조건은 같은 성능의 제어기를 구현하는데 사용되는 입력과 구동기의 용량을 최소화하여 경제성을 높일 수 있다. 이 조건을 식(25)에 나타내었다.

$$\begin{aligned} \text{if } u(k) \geq U_{max}, \text{ then } u(k) &= U_{max}, \\ \text{if } u(k) \leq -U_{max}, \text{ then } u(k) &= -U_{max}, \end{aligned} \quad (25)$$

둘째 학습 주기는 1회 학습하데 걸리는 시간을 의미하는 것으로 이를 짧게 할 수록 전체 학습에 소요되는 시간을 줄일 수 있다. 이를 식(26)에 나타내었다.

$$\int_0^M u(k)dt = \int_0^M [K_p e(k) + K_d e'(k) + K_i \int_0^k e(t)dt] dt$$

$$T_i = \int_0^M dt \quad (26)$$

여기서 M은 스텝 수이고 학습 주기 T_i 은 M스텝과 샘플링 시간의 곱이다. 셋째 조건인 학습 응답은 학습을 위해 시스템에 외란을 가하는 방법을 의미한다. 이 방법에는 임펄스 응답과 스텝응답이 있다.

4. 시뮬레이션

본 논문에서 제안한 PID 학습법을 시스템 'A'~'F'에 적용하였다. 이때 시뮬레이션 조건은 샘플링 시간 0.01초, PID 각 계인에 대한 학습율 η 은 스텝 응답에서 0.02, 임펄스 응답에서 1.0으로 하였다. 스텝응답은 초기 상태 $[x, \dot{x}] = [0.5m, 0m]$, 임펄스 응답은 500[N]을 1스텝(0.01초) 동안 가하여 학습하였다. 그리고 레귤레이션 문제에서 목표값 y_d 는 0(zero)이다. 최적의 학습을 하기 위해서

학습 성능이 최대가 되는 학습 조건을 찾아야 한다. 이를 위해 스텝 응답과 임펄스 응답을 이용하여 최대 입력을 구한 다음, 이 최대 입력을 이용하여 최소 학습 주기를 구한다. 그 다음 이 학습 조건에 대한 학습 성능을 비교하여 학습 응답을 정하였다. 학습 주기를 600스텝으로 정하고 최대 입력은 0N에서 5,000N까지 100N 간격으로 증가시켜서 학습 성능이 최대가 되는 최대 입력을 찾았다. 이 최대 입력을 이용하여 최적의 학습 주기를 찾기 위해 학습 주기를 0스텝에서 1000스텝까지 10스텝(0.1초) 간격으로 증가시켜 학습하였다. 위에서 구한 최대 입력과 학습 주기를 학습 조건으로 하여 스텝응답과 임펄스 응답을 주어 각 시스템에 대한 학습 응답을 구하였다. 이를 Table 3에 나타내었다. Table 4는 학습된 PID 계인과 학습 반복 횟수를 나타내었다. Fig.4~Fig.7은 시뮬레이션한 10개의 시스템 중에서 2개의 시스템('A', 'G')에 대한 결과를 나타내었다. Fig.4는 시스템 'A'에 대한 최대 입력의 변화에 따른 학습 성능을 나타내었다. 최대 입력이 스텝 응답과 임펄스 응답 등

Table 3 Learning conditions and learning performance.

System		A	B	C	D	E	F	G	H	I	J
Step response	Max. Input [N]	1200	1900	900	300	N	2100	400	200	N	N
	Learning period [sec]	0.1	0.1	0.1	0.2	N	0.1	0.1	0.3	N	N
	L_p [%]	99	96	78	30	N	99	80	42	N	N
Impulse response	Max. Input [N]	1200	1900	1000	300	N	2000	2100	2100	800	500
	Learning period [sec]	0.1	0.1	0.1	0.1	N	0.4	0.6	3.5	4.5	7.8
	L_p [%]	99	96	78	30	N	99	98	97	84	77

Table 4 PID gains and iteration number for maximum learning performance.

System		A	B	C	D	E	F	G	H	I	J
Step response	K_p	1.7	1.5	1.0	0.4	N	2.48	0.56	0.78	N	N
	K_i	0.01	0.03	0.03	0.02	N	0.11	0.03	0.12	N	N
	K_d	45.8	40.9	18.9	5.50	N	42.5	5.65	2.68	N	N
	iteration No.	120	128	66	20	N	139	14	8	N	N
Impulse response	K_p	8.2	7.2	3.5	1.0	N	29.7	77.9	112.1	67.17	49.5
	K_i	0.35	0.32	0.16	0.04	N	5.60	20.9	15.3	7.98	4.65
	K_d	46.1	40.9	19.4	5.44	N	40.7	41.8	40.9	14.8	8.40
	iteration No.	154	204	115	39	N	26	88	135	29	15

일하게 1200[N]에서 학습 성능이 99%이다. 최대 입력이 4000[N] 이상에서는 학습 성능이 낮아짐을 알 수 있다. Fig.5는 시스템 'A'에 대한 학습 주기의 변화에 따른 학습 성능을 나타내었다. 학습 주기 0.1[sec]에서 학습 성능이 최고치가 된다. 따라서 최대 입력 1200[N]과 학습 주기의 최소 값 0.1[sec]로 하여 학습한 결과를 Fig.6에 나타내었다. Fig.1의 '1'에 나타낸 학습 전의 응답과 비교해보면 최적의 제어기를 설계하였음을 알 수 있다. Fig.7 시스템 'G'에 대해 위와 동일한 방법으로 하여 학습된 응답을 나타내었다. 이 결과에서는 스텝 응답을 이용하여 학습한 결과와 임펄스 응답을 이용하여 학습한 결과가 많은 차이가 있음을 알 수 있다. Fig.2의 '1'에 나타낸 학습 전의 응답과 비교해보면 최적의 제어기를 설계하였음을 알 수 있다.

5. 결론

본 논문에서 제안한 PID 학습법은 PID제어기의 각 계인을 학습하기 위해 기율기법을 이용하여 학습의 방향을 정하고 학습 성능을 정의하여 이 값이 최대일 때 학습을 종료하였다. 이 학습에서 최대의 학습 효과를 주기 위하여 학습 조건 3가지를 제시하였다. 이 제안한 학습법과 학습 조건을 시스템의 고유진동수와 감쇠비에 따라 선정한 10개의 시스템에 적용하였다. 그 결과 각 시스템에 대한 학습 조건을 구하였으며 이 구한 학습 조건을 이용하여 PID 학습을 함으로써 좋은 성능의 PID 제어기를 설계할 수 있었다. 임펄스 응답을 이용하여 학습한 결과는 대부분 70% 이상의 높은 학습 효과를 얻었으나 고감쇠에 대한 학습 효과가 저 감쇠에 비해 상대적으로 작았다. 따라서 향후 저감쇠 시스템에 대해 학습 효과를 더 높일 수 있는 방법에 대한 연구가 필요하다.

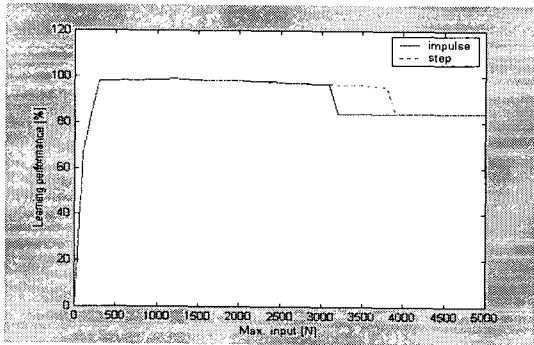


Fig. 4 Learning performance for a Max. input in system 'A'.

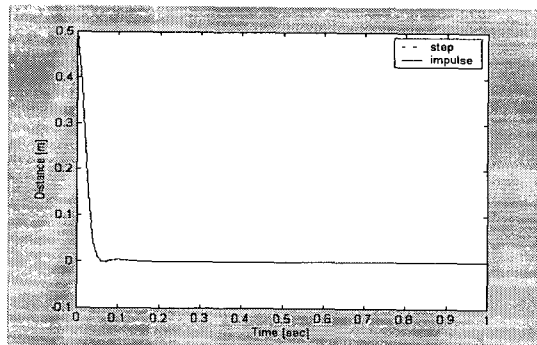


Fig. 6 Response of system 'A' after learning.

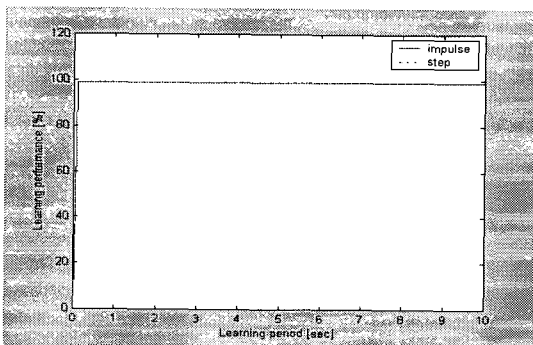


Fig. 5 Learning performance for a learning period in system 'A'.

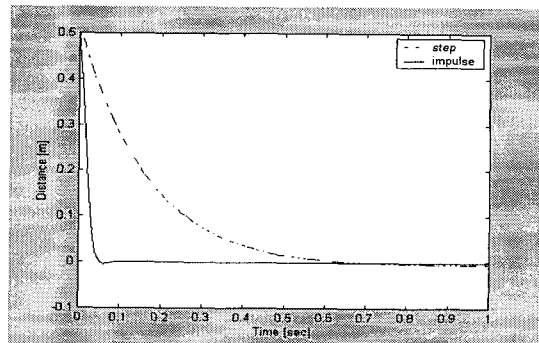


Fig. 7 Response of system 'G' after learning.

참고문헌

1. J. G. Ziegler and N. B. Nichols, "Optimum setting for automatic controllers," Trans. ASME, Vol. 64, pp. 759-768, 1942.
2. K. J. Astrom and T. Hagglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins," Automatica, Vol. 20, No. 5, pp. 645-651, 1984.
3. K. J. Astrom, C. C. Hang, P. persson and W. K. Ho, "Towards Intelligent PID control," Automatica, Vol. 28, No. 1, pp. 1-9, 1992.
4. Z. Y. Zhao, M. Tomizuka and S. Isaka, "Fuzzy gain scheduling of PID controllers," IEEE Trans. Systems Man & Cybernetics, Vol. 23, No. 5, pp. 1392-1398, 1993.
5. K. Warwick and Y-H Kang, "Self-tuning Proportional, integral and derivative controller based on genetic algorithm least squares," Journal of Systems and Control Engineer, Vol. 212 No. 16, pp. 437-448, 1998.
6. B. Chung and J. Oh, "Control of dynamic systems using fuzzy learning algorithm," Fuzzy sets and Systems, Vol. 59, No. 1, pp. 1-14, 1993.
7. B. Chung and J. Oh, "Autotuning method of membership function in a fuzzy learning control," Journal of Intelligent & Fuzzy Systems, Vol. 1, No. 4, pp. 335-349, 1993.
8. 임윤규, 정병목, 소범식, "불안정한 다변수 시스템에 대한 퍼지 학습 제어," 제어·자동화·시스템공학 논문지, 제5권, 7호, pp. 808-813, 1999.
9. 임윤규, 정병목, "퍼지학습법을 이용한 크레인 시스템의 다변수 제어," 한국정밀공학회지, 제16권, 7호, pp. 144-150, 1999.
10. B. Chung, J. Lee, H. Joo and Y. Lim, "Hybrid Fuzzy Learning Controller for an Unstable Nonlinear System," J. of the KSPE, Vol. 1, No. 1, pp. 79-84, 2000.