

동적 마찰이 있는 다변수 시스템에서의 PID 학습 제어

정병목[#]

PID Learning Controller for Multivariable System with Dynamic Friction

Byeong Mook Chung[#]

ABSTRACT

There have been many researches for optimal controllers in multivariable systems, and they generally use accurate linear models of the plant dynamics. Real systems, however, contain nonlinearities and high-order dynamics that may be difficult to model using conventional techniques. Therefore, it is necessary a PID gain tuning method without explicit modeling for the multivariable plant dynamics. The PID tuning method utilizes the sign of Jacobian and gradient descent techniques to iteratively reduce the error-related objective function. This paper, especially, focuses on the role of I-controller when there is a steady state error. However, it is not easy to tune I-gain unlike P- and D-gain because I-controller is mainly operated in the steady state. Simulations for an overhead crane system with dynamic friction show that the proposed PID-LC algorithm improves controller performance, even in the steady state error.

Key Words : PID (Proportional, Integral, and Derivative), LC (Learning Control: 학습제어), Multivariable System (다변수 시스템), Dynamic Friction (동적 마찰), Overhead Crane (크레인)

1. 서론

PID 제어기는 구조가 간단하고 사용이 편리하여 현재까지 산업 현장에서 가장 널리 사용되고 있는 제어기다. 그러나 PID 제어 게인을 최적으로 조정하는 문제는 아직까지도 많은 연구의 대상이 되고 있다. Ziegler 와 Nichols 는 단위 계단 입력에 대한 플랜트의 응답을 보면서 변수를 구하는 방법과 비례게인의 증가에 의해 처음으로 진동이 발생하였을 때의 제어게인과 주파수에 기초를 둔 방법을 제시했다.¹ 이것은 시스템의 모델식을 알 수 없을 때 편리하게 사용할 수 있지만 실험적으로 임계게인과 임계주파수를 찾아내야 하고 이를 이용

하여 일괄적으로 제어기를 설계하는 방식이기 때문에 좋은 제어기를 구하기가 쉽지 않은 단점이 있다. Astrom & Haggglund 는 Nyquist 선도에서 임계게인과 임계 주파수를 지연 피드백을 이용하는 방법을 제시했다.^{2,3} 이것은 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 이론, 등의 인공 지능 이론을 도입하여 최적의 PID 이득을 자동 튜닝(auto-tuning)하는 방법에 대한 연구가 활발하다.^{4,5} 그러나 이들 연구는 제어 대상 시스템이 단일 입출력(SISO: Single Input & Single Output) 시스템이므로 다중 입출력 (MIMO:

접수일: 2006 년 10 월 13 일; 게재승인일: 2007 년 10 월 4 일

교신저자: 영남대학교 공과대학 기계공학부

E-mail: bmchung@yu.ac.kr Tel. (053) 810-2569

Multi-Inputs & Multi-Outputs)을 갖는 다변수 시스템에 대해서는 적용하기 어렵다. 다변수 시스템의 제어에서는 LQR 과 같은 상태 피드백 제어가 많이 사용되지만 모델식을 알고 있는 선형 시스템에 대해서만 적용이 가능하다. 따라서 산업 현장과 같이 시스템의 모델식을 구할 수 없는 경우 뉴럴 네트워크를 이용하여 시스템을 묘사하는 모델을 구한 다음에 이 모델을 토대로 적응 제어를 설계하는 방법이 알려져 있지만 설계된 제어기의 성능이 기대치에 못 미치는 경우가 많다.^{6,9}

Lim & Chung 은 모델식을 모르는 다변수 시스템에서 학습에 의해 직접적으로 PD 제어를 구하는 방법을 제시한 바 있다.¹⁰ 대부분의 제어가 PD 제어만으로 충분하지만 기계시스템의 구동시에 발생하는 마찰 문제는 PD 제어 후에 정상상태 (steady state)에서 오차를 유발시키기 때문에 추가적으로 I(적분) 제어를 사용하지 않으면 곤란하다. 그러나 PID 학습 제어에서는 I 계인의 학습이 다른 PD 계인의 학습에 큰 영향을 주므로 학습을 동시에 수행할 수 없다. 따라서 본 논문에서는 다변수 시스템에서 정상상태의 오차를 줄이기 위한 I 제어의 학습방법을 제시하고 동적 마찰이 있는 크레인 시스템에 대해 정상상태의 오차가 개선되는 것을 시뮬레이션을 통해 보이고자 한다.

2. PID 제어기의 학습법

제어 시스템은 시스템의 출력이 목표치를 잘 추적하도록 하는 것이 목적이므로 목표치와 출력 사이의 오차를 없애는 것이 목적이다. 따라서 p 개의 입력과 q 개의 출력을 가진 시스템에 대한 다중 PID 제어기의 제어입력은 오차와 미분오차, 그리고 적분오차로 구할 수 있다.

$$\mathbf{u}(k) = \mathbf{K}_p \mathbf{e}_p(k) + \mathbf{K}_D \mathbf{e}_D(k) + \mathbf{K}_I \mathbf{e}_I(k) \quad (1)$$

여기서 $\mathbf{K}_p, \mathbf{K}_D, \mathbf{K}_I$ 는 각각 비례게인, 미분 게인, 적분게인 행렬이고 $\mathbf{e}_p(k), \mathbf{e}_D(k), \mathbf{e}_I(k)$ 는 오차, 미분오차, 적분오차 벡터로서 다음과 같이 정의된다.

$$\begin{aligned} \mathbf{e}_p(k) &= \mathbf{d}(k) - \mathbf{y}(k) \\ &= [d_1(k) - y_1(k), \dots, d_q(k) - y_q(k)]^T \end{aligned} \quad (2)$$

$$\begin{aligned} \mathbf{e}_D(k) &= \mathbf{e}_p(k) - \mathbf{e}_p(k-1)^T \\ &= [e_1(k) - e_1(k-1), \dots, e_q(k) - e_q(k-1)]^T \end{aligned} \quad (3)$$

$$\mathbf{e}_I(k) = \sum \mathbf{e}_p(k) = \left[\sum_{j=1}^k e_1(j) \quad \dots \quad \sum_{j=1}^k e_q(j) \right]^T \quad (4)$$

식 (1)은 다음과 같이 행렬을 이용해 간단히 나타낼 수 있다.

$$\mathbf{u}(k) = \mathbf{K} \mathbf{e}(k) \quad (5)$$

여기서

$$\mathbf{K} = [\mathbf{K}_p \quad \mathbf{K}_D \quad \mathbf{K}_I], \quad (6)$$

$$\mathbf{e}(k) = \begin{bmatrix} \mathbf{e}_p(k) \\ \mathbf{e}_D(k) \\ \mathbf{e}_I(k) \end{bmatrix} \quad (7)$$

이다. PID 계인을 학습하기 위해서 학습의 목표함수를 전체 비례오차, 미분오차 적분오차의 합으로 나타내면 다음과 같고

$$C = C_p + C_D + C_I \quad (8)$$

이때 C_p, C_D, C_I 는 다음과 같이 정의된다.

$$C_p = \frac{1}{2} \sum_{h=0}^{\infty} \mathbf{e}_p^T(k+h) \mathbf{W}_p \mathbf{e}_p(k+h) \quad (9)$$

$$C_D = \frac{1}{2} \sum_{h=0}^{\infty} \mathbf{e}_D^T(k+h) \mathbf{W}_D \mathbf{e}_D(k+h) \quad (10)$$

$$C_I = \frac{1}{2} \sum_{h=0}^{\infty} \mathbf{e}_I^T(k+h) \mathbf{W}_I \mathbf{e}_I(k+h) \quad (11)$$

여기서 $\mathbf{W}_p, \mathbf{W}_D, \mathbf{W}_I$ 는 양수로 이뤄진 대각선 가중치 행렬(diagonal positive weighting matrices)이다. 그러나 적분제어는 정상상태에서의 오차를 줄이기 위해 사용하므로 정상상태 이전의 오차는 고려하지 않는 것이 바람직하다. 따라서 s 번째 스텝부터 정상상태가 시작한다면 $k < s$ 에 대해서는 $\mathbf{W}_I = \mathbf{0}$ 이다. 그리고 행렬 \mathbf{W} 를 다음과 같이 정의하면

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_D & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W}_I \end{bmatrix} \quad (12)$$

학습의 목표함수는 다음과 같이 간단히 나타낼 수 있다.

$$C = \frac{1}{2} \sum_{h=0}^{\infty} \mathbf{e}^T(k+h) \mathbf{W} \mathbf{e}(k+h) \quad (13)$$

위의 목표함수를 이용하여 PID 제어 계인은 기울기법을 이용하여 다음과 같이 구할 수 있다.

$$\Delta \mathbf{K} \propto - \left[\frac{\partial C}{\partial \mathbf{K}} \right] = - \left[\frac{\partial C}{\partial \mathbf{u}(k)} \right]^T \left[\frac{\partial \mathbf{u}(k)}{\partial \mathbf{K}} \right] \quad (14)$$

동적 시스템에서 식 (13)과 같이 k 번째 입력이 k 번째 이후의 모든 출력에 영향을 주기는 하지만 학습을 위해서는 $k+n$ 번째 출력이 과거 $k+n$ 번째 이전에 작용한 입력들 가운데 어떤 입력으로부터 얼마나 영향을 받았는지를 알아야 한다. 만일 그것을 알 수 있으면 현재 출력 오차를 과거의 입력이 영향을 준 비율대로 나누고 그 비율의 오차만큼 제어게인 \mathbf{K} 를 수정할 수 있다. 하지만 비율을 알기는 거의 불가능하므로 학습에서는 k 번째 입력이 $k+n$ 번째 출력에 100% 영향을 줬다고 가정하고 학습한다.¹⁰⁻¹² 그러면 입력 $\mathbf{u}(k)$ 는 출력 오차 $\mathbf{e}(k+n)$ 에 만 영향을 줬으므로 $\Delta \mathbf{K}$ 는 다음과 같이 구할 수 있다.

$$\Delta \mathbf{K} \propto - \left[\frac{\partial C}{\partial \mathbf{e}(k+n)} \frac{\partial \mathbf{e}(k+n)}{\partial \mathbf{y}(k+n)} \frac{\partial \mathbf{y}(k+n)}{\partial \mathbf{u}(k)} \right]^T \left[\frac{\partial \mathbf{u}(k)}{\partial \mathbf{K}} \right] \quad (15)$$

여기서 시스템의 자코비안 \mathbf{J} 를 도입하면

$$\mathbf{J} \equiv \frac{\partial \mathbf{y}(k+n)}{\partial \mathbf{u}(k)} \quad (16)$$

이고, 또 식 (5)에 의해

$$\frac{\partial \mathbf{u}(k)}{\partial \mathbf{K}} = \mathbf{e}^T(k) \quad (17)$$

이다. 그리고 식 (2), (3), (4)에 의해

$$\begin{aligned} & \left[\frac{\partial \mathbf{e}(k+n)}{\partial \mathbf{y}(k+n)} \right]^T \\ &= \left[\frac{\partial \mathbf{e}_p(k+n)}{\partial \mathbf{y}(k+n)} \frac{\partial \mathbf{e}_D(k+n)}{\partial \mathbf{y}(k+n)} \frac{\partial \mathbf{e}_I(k+n)}{\partial \mathbf{y}(k+n)} \right] \end{aligned}$$

$$= [-\mathbf{I} \quad -\mathbf{I} \quad -\mathbf{I}] \quad (18)$$

이므로

$$\begin{aligned} & \left[\frac{\partial \mathbf{y}(k+n)}{\partial \mathbf{u}(k)} \right]^T \left[\frac{\partial \mathbf{e}(k+n)}{\partial \mathbf{y}(k+n)} \right]^T = \mathbf{J}^T [-\mathbf{I} \quad -\mathbf{I} \quad -\mathbf{I}] \\ &= -[\mathbf{J}^T \quad \mathbf{J}^T \quad \mathbf{J}^T] \quad (19) \end{aligned}$$

이다. 따라서

$$\mathbf{J}_a^T = [\mathbf{J}^T \quad \mathbf{J}^T \quad \mathbf{J}^T] \quad (20)$$

라고 두면 PID 제어 계인은 다음과 같이 구할 수 있다.

$$\begin{aligned} \Delta \mathbf{K} &= \eta [\mathbf{J}_a^T] [\mathbf{e}^T(k+n) \mathbf{W}]^T [\mathbf{e}^T(k)] \\ &= \eta [\mathbf{J}_a^T] [\mathbf{W} \mathbf{e}(k+n)] [\mathbf{e}^T(k)] \quad (21) \end{aligned}$$

따라서 각각의 제어 계인은 다음과 같이 구한다.

$$\Delta \mathbf{K}_P = \eta [\mathbf{J}^T \mathbf{W}_P \mathbf{e}_P(k+n) \mathbf{e}_P^T(k)] \quad (22)$$

$$\Delta \mathbf{K}_D = \eta [\mathbf{J}^T \mathbf{W}_D \mathbf{e}_D(k+n) \mathbf{e}_D^T(k)] \quad (23)$$

$$\begin{aligned} \Delta \mathbf{K}_I &= \eta [\mathbf{J}^T \mathbf{W}_I \mathbf{e}_I(k+n) \mathbf{e}_I^T(k)] \\ & \quad \text{단, } \mathbf{W}_I = \mathbf{0} \text{ for } k < s \quad (24) \end{aligned}$$

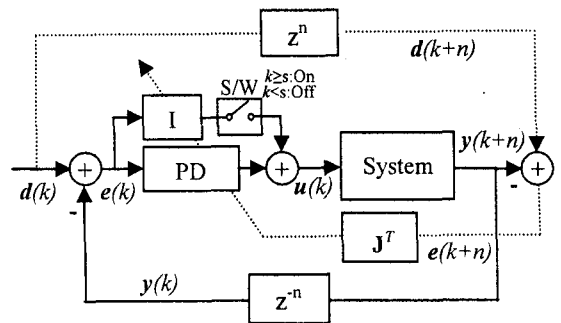


Fig. 1 Block Diagram of PID learning controller

Fig. 1 은 식 (22)-(24)의 PID 계인 학습을 블록선도로 나타낸 것이다. 시스템의 출력지연을 n 스텝이라고 둘 때, n 스텝 후의 오차와 자코비안을 이용하여 수정한다. 그리고 PD 계인의 경우는 조건없이 학습을 하지만 I 계인에 대해서는 정상상

태에서만 학습을 하도록 On/Off 스위치를 두었다.

3. 학습의 수렴

학습의 안정성을 해석하기 위해서는 수정된 제어기에 의한 출력오차가 학습이 진행됨에 따라 수렴하는지 알아볼 필요가 있다. 식 (5)와 (21)에 의해 학습 후의 입력은 학습전의 입력과 다음과 같은 차이가 난다.

$$\Delta u(k) = \Delta K e(k) = \eta \left[J_a^T \right] W e(k+n) e^T(k) e(k) \quad (25)$$

그리고 $e^T(k) e(k)$ 는 양의 스칼라 값이므로 η 에 포함시킬 수 있다. 따라서 학습이 i 번 진행되고 난 후에 $k+n$ 번째 스텝에서 오차의 변화량은 다음과 같이 나타낼 수가 있다.¹⁰

$$e^i(k+n) = \left[I - \eta J \left[J_a^T \right] W \right] e^0(k+n) \quad (26)$$

여기서 I 는 단위행렬(identity matrix)이고 $e^0(k+n)$ 는 학습전의 오차다. 식 (26)에서 만일 학습이 무한히 반복되어 $i \rightarrow \infty$ 이면 그 때에는 더 이상 수정이 이루어지지 않아야 학습이 수렴하게 된다. 따라서 $e^\infty(k+n)$ 이 영이 되기 위해서는 $\left[I - \eta J \left[J_a^T \right] W \right]^i$ 이 영이 되어야 하므로 다음의 조건을 만족하면 된다.

$$\left\| I - \eta J \left[J_a^T \right] W \right\| < 1 \quad (27)$$

그런데 행렬 $\left[I - \eta J \left[J_a^T \right] W \right]$ 는

$$\begin{aligned} & \left[I - \eta J \left[J_a^T \right] W \right] \\ &= \begin{bmatrix} J J^T W_P & 0 & 0 \\ I - \eta & J J^T W_D & 0 \\ 0 & 0 & J J^T W_I \end{bmatrix} \\ &= \begin{bmatrix} I - \eta J J^T W_P & 0 & 0 \\ 0 & I - \eta J J^T W_D & 0 \\ 0 & 0 & I - \eta J J^T W_I \end{bmatrix} \quad (28) \end{aligned}$$

이므로

$$\begin{aligned} & \left\| I - \eta J \left[J_a^T \right] W \right\| \\ &= \left\| \left[I - \eta J \left[J^T \right] W_P \right] \left[I - \eta J \left[J^T \right] W_D \right] \left[I - \eta J \left[J^T \right] W_I \right] \right\| \quad (29) \end{aligned}$$

이 된다. 따라서 식 (27)을 만족하기 위해서는 다음의 조건을 각각 만족하면 충분하다.

$$\left\| I - \eta J \left[J^T \right] W_P \right\| < 1 \quad (30)$$

$$\left\| I - \eta J \left[J^T \right] W_D \right\| < 1 \quad (31)$$

$$\left\| I - \eta J \left[J^T \right] W_I \right\| < 1 \quad (32)$$

여기서 $J \left[J^T \right]$ 는 양의 한정(positive definite) 대칭행렬이고 W_P, W_D, W_I 는 양수로 이뤄진 대각선가중치 행렬이므로 $J \left[J^T \right] W_P, J \left[J^T \right] W_D, J \left[J^T \right] W_I$ 역시 각각 양의 한정(positive definite) 대칭행렬이 된다.¹³ 그러므로 학습율(η)를 충분히 작은 값으로 선정하면 $\eta J \left[J^T \right] W_P$ 이 1 보다 작게 되어 식 (30)의 조건을 만족하게 된다. 마찬가지로 식 (31)과 (32)까지 만족하는 충분히 작은 η 를 선정하면 제안한 PID 학습의 수렴성은 보장된다.

4. 시뮬레이션

Fig. 2 에서 보는 바와 같이 크레인 시스템은 1 개의 입력과 2 개의 출력이 있는 시스템이다. 크레인 시스템은 운반물을 크레인으로 잡아서 트레일러를 일정한 거리로 이동시키는 장치로써 트레일러의 이동 위치를 제어함과 동시에 운반물을 잡은 크레인의 진동도 역시 제어해야 한다. 따라서 트레일러를 재빨리 1m 이동시키면서 이때 발생하는 크레인의 각도를 제어하는 문제에 대해 PID 학습 제어기를 적용해 보고자 한다.

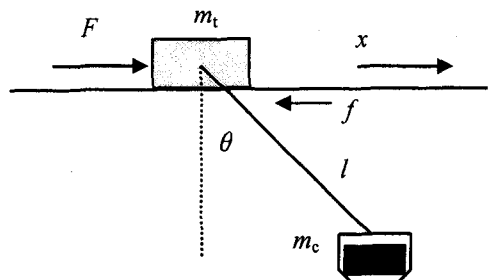


Fig. 2 Schematic Diagram of Overhead Crane

동적마찰(dynamic friction or Coulomb friction)을 고려한 크레인 시스템의 운동방정식은 다음과 같이 구할 수 있고 마찰력 f 는 Fig. 3 과 식 (35)와 같이 정의 된다.¹⁴

$$(m_t + m_c)\ddot{x} + f + m_c l(\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta) = F \quad (33)$$

$$m_c \ddot{x} \cos \theta + m_c l^2 \ddot{\theta} + m_c g l \sin \theta = 0 \quad (34)$$

$$f = F_c \text{sign}(\dot{x}) \quad (35)$$

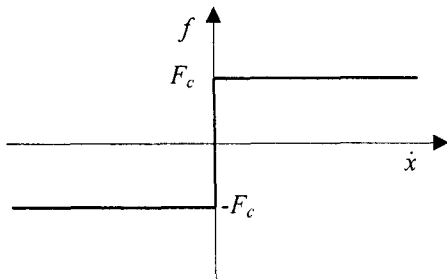


Fig. 3 Characteristic Diagram of Dynamic Friction

여기서 트레일러(m_t)와 크레인(m_c)의 질량은 각각 100kg 과 200kg 이고 줄의 길이(l)는 2m 이며 g 는 중력가속도 이다. 시뮬레이션에서는 마찰이 없는 경우에 대해 LQR 과 PD 학습제어를 먼저 비교하고 다음으로 마찰이 있는 경우에 대해 PD 학습제어와 PID 학습제어를 비교하고자 한다. 먼저 학습법을 적용하기 위해서는 시스템의 자코비안 값을 알아야 하는데 정확한 자코비안을 구하기가 쉽지 않다. 하지만 시스템 입력 증가에 따른 출력의 증감 부호만을 이용해 근사 자코비안을 구하고 이를 이용해 학습할 수 있는 방법이 제시된 바 있다.¹² 본 논문의 크레인 시스템에서는 입력 F 의 증가에 대해 거리 x 는 증가하지만 줄의 각도 θ 는 감소한다는 것을 직관적으로 간단히 알 수 있으므로 다음과 같은 근사 자코비안을 사용하기로 한다.

$$\mathbf{J} \approx \text{sign}(\mathbf{J}) = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad (36)$$

4.1 마찰이 없는 경우: PD 제어

크레인 시스템에서 마찰을 고려하지 않을 경우, 정상상태에서 오차가 발생하지 않으므로 PD 제어기만으로 제어가 가능하다. 먼저 마찰이 없는 경

우에 대한 PD 학습(PD-LC) 제어기와 LQR 제어기를 비교하고자 한다.

LQR 제어기를 구하기 위해서 식 (33)과 (34)를 선형의 상태방정식으로 나타내면 다음과 같다.

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{(m_c + m_t)g}{m_t l} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{m_c}{m_t} g & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{m_t l} \\ 0 \\ \frac{1}{m_t} \end{bmatrix} F \quad (37)$$

입력의 최대를 300N 으로 제한하고 MatLab 을 이용하여 식 (38)과 같은 가중치를 이용하여 LQR 제어기를 설계해 보면 식 (39)와 같은 제어기를 얻을 수 있다.

$$\mathbf{Q} = \text{diag}[10 \ 50 \ 10 \ 1], \mathbf{R} = 0.0001 \quad (38)$$

$$\mathbf{K} = [-566 \ -37 \ 316 \ 486] \quad (39)$$

PD 학습 제어기는 식 (22) 와 (23)을 이용하고 여기에 사용하는 자코비안은 식 (36)의 근사 자코비안을 사용하였다. 가중치 행렬은 모든 초기값을 10 으로 두고 과잉응답(overshoot)이 나타나면 미분 계인이 증가하도록 \mathbf{W}_D 의 가중치를 증가시킴으로써 다음과 같은 가중치 행렬과 PD 제어 계인을 최종적으로 구했다.

$$\mathbf{W}_P = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \mathbf{W}_D = \begin{bmatrix} 20 & 0 \\ 0 & 60 \end{bmatrix} \quad (40)$$

$$\mathbf{K}_P = [-406 \ 406], \mathbf{K}_D = [-219 \ 657] \quad (41)$$

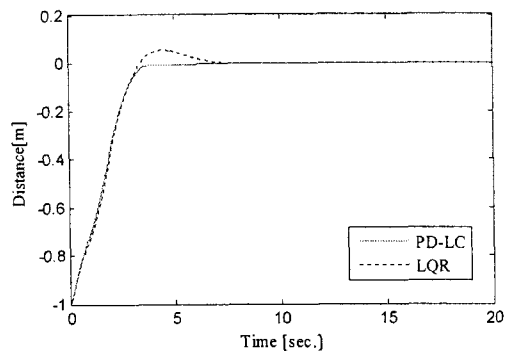


Fig. 4 Comparison of PD-LC and LQR in Distance

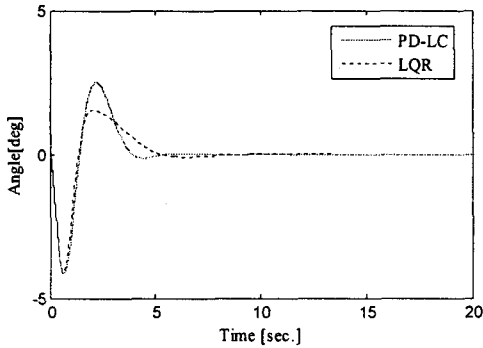


Fig. 5 Comparison of PD-LC and LQR in Angle

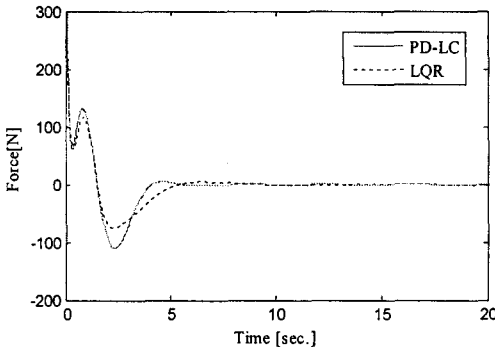


Fig. 6 Comparison of PD-LC and LQR in Input

이때, 학습율은 $\eta=0.01$ 이었고 25 회 동안 학습을 반복했다. Fig. 4 와 5 는 크레인의 위치와 각도에 대한 PD 제어와 LQR 제어 결과를 보여주고 Fig. 6 에서는 이 때의 제어 입력을 보여준다. 모든 그림에서 실선은 PD 제어의 결과를 점선은 LQR 제어 결과를 나타낸다.

마찰을 고려하지 않은 크레인의 제어에서 LQR 제어는 정착시간이 약 7 초였으나 PD-LC 는 크레인의 진동이 약간 더 심하긴 하지만 과잉응답이 없으면서 약 4 초로서 LQR 에 비해 더 좋은 결과를 얻었음을 알 수 있다.

4.2 마찰이 있는 경우 : PID 제어

크레인 시스템에서 마찰의 영향을 고려할 경우, 정상상태에서 오차가 발생하므로 적분(I)제어를 추가할 필요가 있다. 적분제어가 정상상태에서의 오차를 줄이기 위해 사용하는 것이기는 하지만 일반적인 PID 제어에서는 적분제어를 처음부터 사용해도 과잉응답이 조금 더 증가하는 정도 외에 더 이상의 큰 문제가 발생하지는 않는다. 그러나 PID

학습제어에서는 처음부터 적분제어를 포함해 학습할 경우에는 P, D 게인과 관련한 오차나 미분오차보다 I 게인과 관련한 적분오차가 상대적으로 크기 때문에 PD 게인이 제대로 학습되지 않는다. 즉, 식 (27)의 학습 수렴 조건에서 볼 때, 적분오차까지 포함된 오차의 제곱 항 $e^T(k)e(k)$ 을 고려하면 학습율(η)는 아주 작아야 만이 수렴조건을 만족할 수 있다. 그런데 학습율이 너무 작으면 학습이 진행되지 않고 조금이라도 크면 학습이 발산하게 된다. 따라서 I 게인을 학습할 경우, 그 학습과 적용을 정상상태만으로 제한하는 것이 필요하다.

본 논문에서는 정상상태에 대한 조건으로 오차가 10% 이내이면서 오차의 기울기 변화가 거의 없는(0.1% 이내) 상태를 계속 유지하는 경우로 제한하였다. 일반적인 제어에서 나타나는 진동현상에서 이러한 조건을 순간적으로 만족하는 경우도 있지만 정상상태의 범위를 벗어나면 그 동안 적분한 오차를 다시 초기화시킴으로써 정상상태에서 오차가 변하지 않을 때에만 I 제어 게인을 학습하고 적용하였다. 마찰이 있는 크레인 시스템에 대하여 LQR 제어기와 PD-LC 그리고 PID-LC, 이렇게 서로 다른 3 가지 제어방법을 비교 해 보았고 그 결과를 Fig. 7, 8, 9 에서 나타내었다. 마찰력은 Fig. 3 에서 보는 바와 같이 비선형 함수이다. 따라서 LQR 제어기에서는 이것을 고려할 수가 없기 때문에 마찰을 무시한 LQR 제어기를 사용했다. PD-LC 에서는 식 (40)의 가중치를 그대로 사용하여 학습한 결과 19 회의 학습만에 다음과 같은 제어게인을 얻었다.

$$K_P = [-333 \ 333], K_D = [-159 \ 479] \quad (42)$$

그림에서 점선은 PD-LC 의 결과를, 일점쇄선은 LQR 의 결과를 나타낸다. 트레일러의 운동에서 동적마찰이 있을 경우에 그림에서 보는 바와 같이 PD-LC 와 LQR 모두 약 2%의 정상상태 오차를 가지고 있었다. 또한 트레일러의 운동 마찰로 인해 크레인의 진동이 오랫동안 지속됨을 알 수 있다. PID-LC 에서는 정상상태 오차를 줄이기 위해 식 (43)과 같은 I 가중치 행렬을 이용해 I 게인을 학습하였고 23 회의 학습 후에 식 (44)와 같은 PID 제어 게인을 구할 수 있었다.

$$W_I = \begin{bmatrix} 100 & 0 \\ 0 & 200 \end{bmatrix} \quad (43)$$

$$\begin{aligned} \mathbf{K}_P &= [-367 \ 367], \mathbf{K}_D = [-183 \ 550], \\ \mathbf{K}_I &= [-96 \ 193] \end{aligned} \quad (44)$$

I 제어기의 추가로 인해 트레일러의 위치제어는 Fig. 7 에서 보는 바와 같이 정상상태에서의 오차가 크게 개선 되었다. 그러나 Fig. 8 의 각도제어의 결과는 마찰력이 없는 Fig. 5 와 비교해 볼 때, 트레일러의 마찰로 인한 운동 제약으로 제어방법에 상관없이 크레인의 진동이 상당히 길게 나타남을 알 수 있다. Fig. 9 의 제어입력 선도를 보면 PD-LC 나 LQR 에서는 10 초 이후에 더 이상의 제어 입력변화가 없다. 하지만 PID-LC 에서는 오차의 적분에 비례한 입력이 정상상태의 오차를 지속적으로 줄이고 있는 것을 알 수 있다.

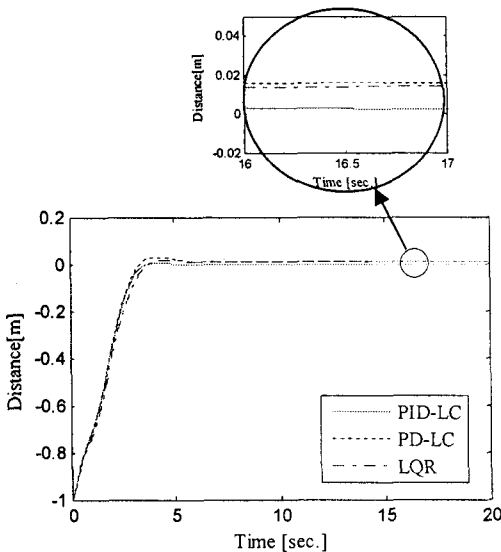


Fig. 7 Comparison of PID-LC and PD-LC in Distance

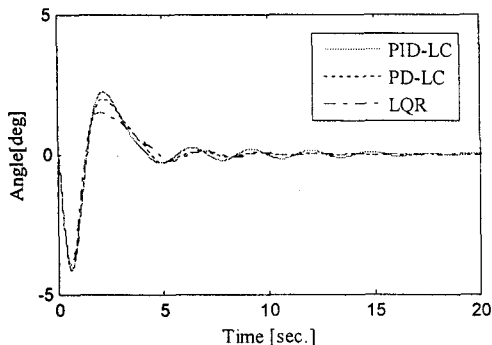


Fig. 8 Comparison of PID-LC and PD-LC in Angle

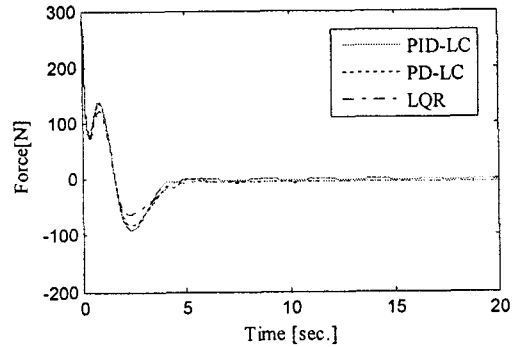


Fig. 9 Comparison of PID-LC and PD-LC in Input

식 (44)의 학습된 제어게인을 살펴보면 그 부호는 자코비안과 같고 크기는 가중치 행렬에 비례하는 것을 알 수 있다. 이것은 학습 알고리즘이 식 (22)-(24)에서 보는 바와 같이 비례, 미분, 적분 각각의 오차에 대해 가중치 행렬에 비례한 값으로 제어게인을 개선했기 때문이다. 따라서 PID 학습 제어의 핵심은 먼저 자코비안을 통해 학습에 필요한 부호를 얻는 것과 다음에 학습 결과를 토대로 LQR 제어처럼 가중치 행렬을 조정하면서 좋은 출력의 제어게인을 찾아내는 것이다.

5. 결론

본 논문에서는 다변수 시스템의 제어에서 정상상태의 오차가 발생할 때 이를 제거하기 위한 I 제어 게인의 학습법과 학습의 수렴 조건을 제안하였다. 동적 마찰을 갖는 크레인 시스템의 경우, LQR 이나 PD 학습 제어에서는 정상상태에서의 오차를 제대로 제거할 수 없었지만 본 논문에서 제안한 PID 학습법은 정상상태에서의 오차를 잘 제거할 수 있다는 것을 알 수 있었다. PID 학습에서 I 게인의 학습은 P, D 게인과는 달리 과도응답 (transient response)구간에서는 학습할 수 없어서 정상상태에서만 학습이 가능하도록 하였다. 지금까지 다변수 시스템의 제어에서 널리 사용되어 온 LQR 제어가 마찰과 같은 비선형 성분을 포함한 문제를 제대로 해결하지 못했다. 따라서 PID 제어가 가능한 다변수 비선형 시스템의 제어에 대해서는 본 논문이 제시하는 PID 학습 제어법을 사용하면 최적의 PID 게인을 찾아내는데 아주 유익하리라고 본다.

후 기

이 논문은 2005 학년도 영남대학교 학술연구 조성비에 의한 것임.

참고문헌

1. Ziegler, J. G. and Nichols, N. B., "Optimum setting for automatic controllers," *Trans. ASME*, Vol. 64, pp. 759-768, 1942.
2. Astrom, K. J. and Haggglund, T., "Automatic tuning of simple regulators with specifications on phase and amplitude margins," *Automatica*, Vol. 20, No. 5, pp. 645-651, 1984.
3. Astrom, K. J., Hang, C. C., Person, P. and Ho, W. K., "Towards Intelligent PID control," *Automatica*, Vol. 28, No. 1, pp. 1-9, 1992.
4. Zhao, Z. Y., Tomizuka, M. and Isaka, S., "Fuzzy gain scheduling of PID controllers," *IEE Trans. Systems, Man, & Cybernetics*, Vol. 23, No. 5, pp. 1392-1398, 1993.
5. Warwick, K. and Kang, Y. H., "Self-tuning proportional, integral and derivative controller based on genetic algorithm least squares," *Proc. of the Institute of Mechanical Engineers, part I: Journal of Systems and Control Engineering*, Vol. 212 No. 6, pp. 437-448, 1998.
6. Narendra, K. S. and Parthasarathy, K., "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, Vol. 1, No. 1, pp. 4-26, 1990.
7. Sanner, R. M. and Slotine, J. E., "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Networks*, Vol. 3, No. 6, pp. 837-863, 1992.
8. Chen, F. C., "Back-propagation neural networks for nonlinear self-tuning adaptive control," *IEEE Control System Magazine*, Vol. 10, No. 3, pp. 44-48, 1990.
9. Chen, F. C. and Khalil, H. K., "Adaptive control of nonlinear systems using neural networks," *Int. J. of Control*, Vol. 55, No. 6, pp. 1299-1317, 1992.
10. Lim, Y. K. and Chung, B. M., "A learning method of PID controller by Jacobian in multi variable system," *J. of KSPE*, Vol. 20, No. 2, pp. 112-119, 2003.
11. Lim, Y. K. and Chung, B. M., "A learning method of LQR controller using Jacobian," *J. of KSPE*, Vol. 22, No. 8, pp. 34-41, 2005.
12. Chung, B. M., "Control of nonlinear multivariable Systems using direct fuzzy learning method," *Int. J. of Intelligent & Fuzzy Systems*, Vol. 5, No. 3, pp. 297-310, 1997.
13. Zhang, F., "Matrix Theory: Basic Results and Techniques," Springer, pp. 159-207, 1999.
14. Ahn, H. S., Chen, Y. Q. and Dou, H., "State-Periodic Adaptive Compensation of Cogging and Coulomb Friction in Permanent-Magnet Linear Motors," *IEEE Trans. Magnetics*, Vol. 41, No. 1, pp. 90-98, 2005.