

다변수 시스템에서 자코비안을 이용한 PID 제어기 학습법

임윤규*, 정병목**

A Learning Method of PID Controller by Jacobian in Multi Variable System

Yoon Kyu Lim*, Byeong Mook Chung*

ABSTRACT

Generally, PID controller is not suitable to control multi variable system because it is very difficult to tune the PID gains. However, this paper shows that it is not hard to tune the PID gains if we can find a Jacobian matrix of the system. The Jacobian matrix expresses the ratio of output variations according to input variations. It is possible to adjust the input values in order to reduce the output error using the Jacobian. When the cost function is composed of error related terms, the gradient approach can tune the PID gains to minimize the function. In simulation, a hydrofoil catamaran with two inputs and two outputs is applied as a multi variable system. We can easily get the multi variable PID controller by the proposed learning method. When the controller is compared with LQR controller, the performance is as good as that of LQR controller with a modeling equation.

Key Words : PID control(PID 제어), Multi variable system(다변수 시스템), Hydrofoil catamaran(수중익 쌍동선), Gradient approach(기울기법), Jacobian(자코비안)

1. 서론

자동제어 이론이 나온 이후 오랜 기간 사용되어온 PID 제어기는 현재에도 산업 현장에서 가장 널리 사용되고 있는 제어기이다. PID 제어기는 구조가 간단하여 작업자와 제어기 설계자에게 익숙하고 신뢰성도 있다. 그러나 PID 제어 이득을 최적으로 조정하는 문제는 아직까지도 많은 분야에서 연구되고 있다. Ziegler와 Nichols는 단위 계단 입력에 대한 플랜트의 응답을 보면서 변수를 구하는 방법과 비레이드의 증가에 의해 처음으로 진동이 발생하였을 때의 제어이득과 주파수에 기초를 둔 방법을 제시하였다^[1]. 이 방

법은 시스템의 모델식을 알 수 없을 때 편리하게 사용할 수 있지만 제어 출력이 진동을 유지하기 어려울 때는 사용할 수 없는 단점이 있다. Astrom과 Hagglund는 Nyquist선도에서 임계이득과 임계 주파수를 지연 피드백(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 이론, 등의 인공지능 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 :

2002년 8월 23일 접수

* 영남대학교 기계공학부

Single Input Single Output) 시스템이므로 다중 입출력 (MIMO : Multi-Input Multi-Output) 시스템 제어에 적용하기 어렵다. 다중 입출력 시스템의 제어에 가장 많이 사용하는 기법은 LQR과 같은 상태 피드백 제어이다^[6-8]. 그러나 산업 현장과 같이 시스템의 모델식을 구할 수 없는 경우에는 적용할 수 없다^[9-11]. 따라서 본 논문에서는 다중 입출력 시스템 제어에서 시스템의 모델식을 구하지 않고 학습에 의해 자동으로 PID 게인 매트릭스로 구성된 다중 PID 제어기(Multi-PID)를 설계하는 학습법을 제안하며 학습의 수렴성을 증명한다. 이것은 시스템의 목표치와 현재 출력의 차인 에러를 목적함수로 정의하고, 이 목적함수를 최소화하도록 PID 제어기를 수정한다. 이 과정에서 시스템의 입력과 출력 관계를 나타내는 자코비안을 포함하고 있기 때문에 실험적인 방법으로 추정 자코비안을 구하는 방법을 제시한다. 다중 입출력 시스템으로 수종의 쌍동선 제어 문제에 대하여 PID 학습 제어기(PID-LC : PID Learning Controller)를 구하고, 이 PID 제어기와 모델식으로부터 구한 LQR 제어기와의 성능을 비교해 봄으로써 제안한 학습법의 우수성을 확인한다.

2. Multi-PID 제어기 학습법

2.1 학습 알고리즘

제어 시스템은 시스템의 출력이 목표치를 잘 추적하도록 하는 것이 목적이므로 목표치와 출력 사이의 에러를 없애는 것이 목적이다. 따라서 t_{n+1} 시간의 에러 $e(t_{n+1})$ 을 줄이기 위해서는 바로 전에 가해진 입력을 조정해야한다. p 개의 입력과 q 개의 출력이 있는 시스템에 대한 목적함수를 다음과 같이 정의하였다.

$$C = \frac{1}{2} e^T(t_{n+1})e(t_{n+1}) \\ = \frac{1}{2} [d(t_{n+1}) - y(t_{n+1})]^T [d(t_{n+1}) - y(t_{n+1})] \quad (1)$$

여기서 $d(t_n) = \begin{bmatrix} d_1(t_n) \\ \vdots \\ d_q(t_n) \end{bmatrix}$, $y(t_n) = \begin{bmatrix} y_1(t_n) \\ \vdots \\ y_q(t_n) \end{bmatrix}$ 이다.

목적함수를 줄이기 위한 입력의 조정량은 기울기법을 이용하여 수정하면

$$\Delta u(t_n) \propto - \frac{\partial C}{\partial u(t_n)} \quad (2)$$

이고 $u(t_n) = [u_1(t_n), \dots, u_p(t_n)]^T$ 이다. 목적 함수는 출력의 함수이므로 위 식은 다음과 같이 나타낼 수 있다.

$$\Delta u(t_n) \propto - \frac{\partial C}{\partial y(t_{n+1})} \frac{\partial y(t_{n+1})}{\partial u(t_n)} \\ = e^T(t_{n+1}) \frac{\partial y(t_{n+1})}{\partial u(t_n)} \quad (3)$$

이 식에서 시스템의 입력에 대한 출력의 편미분은 자코비안으로 나타낼 수 있으므로

$$J \equiv \frac{\partial y(t_{n+1})}{\partial u(t_n)} \\ = \begin{bmatrix} \frac{\partial y_1(t_{n+1})}{\partial u_1(t_n)} & \dots & \frac{\partial y_1(t_{n+1})}{\partial u_p(t_n)} \\ \vdots \\ \frac{\partial y_q(t_{n+1})}{\partial u_1(t_n)} & \dots & \frac{\partial y_q(t_{n+1})}{\partial u_p(t_n)} \end{bmatrix} \quad (4)$$

이다. 따라서 자코비안을 이용해 식(3)을 다시 표현하면

$$\Delta u(t_n) = \eta e^T(t_{n+1}) J \\ = \eta J^T e(t_{n+1}) \quad (5)$$

이고, 여기서 η 는 학습율이다. 위 식에 의해 에러를 줄이기 위한 입력의 변화량 $\Delta u(t_n)$ 를 구하기 위해서는 시스템의 자코비안이 필요하다는 것을 알 수 있다. 입력 $u(t_n)$ 를 시스템에 주었을 때 시스템의 출력은 $y(t_{n+1})$ 이 되고, 따라서 에러는 $e(t_{n+1})$ 이 된다. 이 에러를 줄이기 위해서 현재 출력에 입력이 영향을 미친 만큼 입력을 수정하여야 하고, 이를 위해서는 시스템의 모델을 알아야 한다. 그러나 시스템의 모델을 모르기 때문에 Fig. 1에서 처럼 자코비안을 이용하여 제어량을 수정한다.

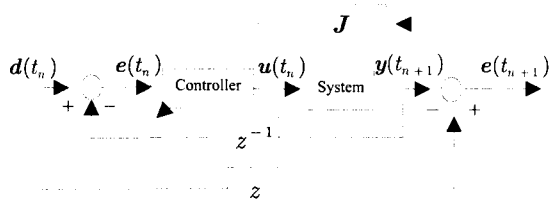


Fig. 1 Block diagram of learning the input of system using Jacobian

2.2 학습의 수렴

이장에서 식(5)의 학습식을 사용하여 학습하였을 때 에러가 영으로 수렴하는 것을 증명하려고 한다.

이산 시간에서 시스템의 상태 방정식은 다음과 같다.

$$\mathbf{x}(t_{n+1}) = \Phi \mathbf{x}(t_n) + \Gamma \mathbf{u}(t_n) \quad (6)$$

$$\mathbf{y}(t_{n+1}) = \mathbf{H} \mathbf{x}(t_{n+1}) \quad (7)$$

여기서 \mathbf{x} 는 상태 변수이고, Φ , Γ 와 \mathbf{H} 는 시스템 매트릭스이다. 식(4)를 이용하여 이산 시간에서 시스템의 자코비안을 구하면 다음과 같다.

$$\begin{aligned} \mathbf{J} &= \frac{\partial \mathbf{y}(t_{n+1})}{\partial \mathbf{x}(t_{n+1})} \frac{\partial \mathbf{x}(t_{n+1})}{\partial \mathbf{u}(t_n)} \\ &= \mathbf{H} \Gamma \end{aligned} \quad (8)$$

초기 입력 \mathbf{u}^0 에 의해 출력 \mathbf{y}^0 는 다음과 같이 구할 수 있다.

$$\begin{aligned} \mathbf{x}^0(t_{n+1}) &= \Phi \mathbf{x}^0(t_n) + \Gamma \mathbf{u}^0(t_n) \\ \mathbf{y}^0(t_{n+1}) &= \mathbf{H} \mathbf{x}^0(t_{n+1}) \end{aligned} \quad (9)$$

출력 $\mathbf{y}^0(t_{n+1})$ 가 목표치 $\mathbf{d}(t_{n+1})$ 에 도달하도록 입력을 수정하여야 한다. 학습식 (5)에 의해 입력은 다음과 같이 수정된다.

$$\Delta \mathbf{u}^0(t_n) = \eta \hat{\mathbf{J}}^T \mathbf{e}^0(t_{n+1}) \quad (10)$$

여기서 $\hat{\mathbf{J}}$ 은 실험적으로 추정된 시스템의 자코비

안이고, $\mathbf{e}^0(t_{n+1}) = \mathbf{d}(t_{n+1}) - \mathbf{y}^0(t_{n+1})$ 이다. 위 식에서 정확한 자코비안 대신에 추정 자코비안을 사용한 것은 시스템의 모델식을 모르는 경우이므로 자코비안을 실험적인 방법으로 구해야 하기 때문이다. 첫 번째 학습을 한 후의 입력은 다음과 같이 수정된다.

$$\mathbf{u}^1(t_n) = \mathbf{u}^0(t_n) + \Delta \mathbf{u}^0(t_n) \quad (11)$$

식(11)에 의해 상태 변수는

$$\begin{aligned} \mathbf{x}^1(t_{n+1}) &= \Phi \mathbf{x}^0(t_n) + \Gamma \mathbf{u}^1(t_n) \\ &= \mathbf{x}^0(t_{n+1}) + \eta \Gamma \hat{\mathbf{J}}^T \mathbf{e}^0(t_{n+1}) \end{aligned} \quad (12)$$

이 되므로 출력은

$$\begin{aligned} \mathbf{y}^1(t_{n+1}) &= \mathbf{H} \mathbf{x}^1(t_{n+1}) \\ &= \mathbf{H} \mathbf{x}^0(t_{n+1}) + \eta \mathbf{H} \Gamma \hat{\mathbf{J}}^T \mathbf{e}^0(t_{n+1}) \end{aligned} \quad (13)$$

이고, 식(13)의 $(\mathbf{H} \Gamma)$ 는 식(8)에 의해 자코비안 값이므로

$$\mathbf{y}^1(t_{n+1}) = \mathbf{y}^0(t_{n+1}) + \eta \mathbf{J} \hat{\mathbf{J}}^T \mathbf{e}^0(t_{n+1}) \quad (14)$$

이 된다. 식(14)를 이용하여 에러를 구하면 다음과 같이 나타낼 수 있다.

$$\mathbf{e}^1(t_{n+1}) = (\mathbf{I} - \eta \mathbf{J} \hat{\mathbf{J}}^T) \mathbf{e}^0(t_{n+1}) \quad (15)$$

두 번째 학습에서 에러 $\mathbf{e}^1(t_{n+1})$ 를 줄이기 위한 입력의 변화량은 식(10)에서와 같은 방법으로 구하면

$$\Delta \mathbf{u}^2(t_n) = \eta \hat{\mathbf{J}}^T \mathbf{e}^1(t_{n+1}) \quad (16)$$

이다. 식(16)의 입력이 변하였을 때 출력은 식(17)에서와 같이 구할 수 있다.

$$\mathbf{y}^2(t_{n+1}) = \mathbf{y}^1(t_{n+1}) + \eta \mathbf{J} \hat{\mathbf{J}}^T \mathbf{e}^1(t_{n+1}) \quad (17)$$

위 식을 식(15)에서와 같이 에러로 표현하면 다음 식과 같다.

$$\begin{aligned} e^2(t_{n+1}) &= (I - \eta J \hat{J}^T) e^1(t_{n+1}) \\ &= (I - \eta J \hat{J}^T)^2 e^0(t_{n+1}) \end{aligned} \quad (18)$$

앞과 같은 방법으로 i 번째 학습을 하였을 때 에러는

$$e^i(t_{n+1}) = (I - \eta J \hat{J}^T)^i e^0(t_{n+1}) \quad (19)$$

이 되므로, 학습이 ∞ 번 반복되었을 때 에러는 다음과 같이 표현할 수 있다.

$$e^\infty(t_{n+1}) = (I - \eta J \hat{J}^T)^\infty e^0(t_{n+1}) \quad (20)$$

학습이 수렴하기 위해서는 ∞ 번 반복했을 때 에러 $e^\infty(t_{n+1})$ 가 영으로 수렴해야 한다. 따라서 $e^\infty(t_{n+1})$ 이 되기 위해서는 다음 조건을 만족해야 한다.

$$|(I - \eta J \hat{J}^T)| < 1 \quad (21)$$

실험에 의해 구한 추정 자코비안이 실제 자코비안과 같다면, 식(21)의 $(J \hat{J}^T)$ 는 양의 한정(positive definite) 대칭행렬이 되므로 고유치(eigen value)는 항상 0보다 큰 실수 고유치를 갖는다. 따라서 충분히 작은 학습율(η)를 취하면 식(21)의 조건을 만족하고 $e^\infty(t_{n+1})$ 가 영이 되므로 제안한 학습이 결국에는 수렴함을 알 수 있다. 그러나 실질적으로 추정 자코비안과 실제 자코비안이 정확히 같을 수는 없으며, 실험조건에 따라 오차가 발생할 수 있지만 자코비안의 부호(방향성)가 바뀌지 않는 범위 내에서는 학습이 수렴한다.

2.3 PID 제어기 학습

p 개의 입력과 q 개의 출력을 가진 시스템에 대한 다중 PID 제어기의 제어 입력은 에러, 에러 변화율과 에러의 합으로 구할 수 있다.

$$u(t_n) = K_P e(t_n) + K_D \dot{e}(t_n) + K_I \int e(t_n) \quad (22)$$

여기서 에러 변화율은

$$\dot{e}_d(t_n) = \left[\frac{e_1(t_n) - e_1(t_{n-1})}{\Delta t}, \dots, \frac{e_q(t_n) - e_q(t_{n-1})}{\Delta t} \right]^T$$

이고, 에러의 합은

$$e_I(t_n) = \left[\sum_{k=0}^n [e_1(t_k) \Delta t], \dots, \sum_{k=0}^n [e_q(t_k) \Delta t] \right]^T \text{이며,}$$

PID 제어 계인은

$$K_P = \begin{bmatrix} k_{p_1} & \dots & k_{p_p} \\ \vdots & & \vdots \\ k_{p_1} & \dots & k_{p_p} \end{bmatrix}, K_I = \begin{bmatrix} k_{i_1} & \dots & k_{i_q} \\ \vdots & & \vdots \\ k_{i_1} & \dots & k_{i_q} \end{bmatrix}, K_D = \begin{bmatrix} k_{d_1} & \dots & k_{d_q} \\ \vdots & & \vdots \\ k_{d_1} & \dots & k_{d_q} \end{bmatrix} \text{이다.}$$

다중 PID 제어기 학습의 목적함수는 식(23)-(25)와 같이 P 제어의 목적함수, I 제어의 목적함수와 D 제어의 목적함수로 정의하였다.

$$C_p = \frac{1}{2} e^T(t_{n+1}) W_p e(t_{n+1}) \quad (23)$$

$$C_I = \frac{1}{2} e_I^T(t_{n+1}) W_I e_I(t_{n+1}) \quad (24)$$

$$C_d = \frac{1}{2} \dot{e}_d^T(t_{n+1}) W_d \dot{e}_d(t_{n+1}) \quad (25)$$

$$\text{여기서 } W_p = \begin{bmatrix} w_{p_1} & 0 & \dots & 0 \\ 0 & w_{p_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_{p_p} \end{bmatrix}, W_I = \begin{bmatrix} w_{i_1} & 0 & \dots & 0 \\ 0 & w_{i_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_{i_q} \end{bmatrix}$$

$$\text{와 } W_d = \begin{bmatrix} w_{d_1} & 0 & \dots & 0 \\ 0 & w_{d_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_{d_q} \end{bmatrix} \text{는 각 제어의 가중치 매트릭스이다.}$$

2.3.1 P 제어 계인의 튜닝

P 제어 계인의 학습은 식(23)의 P 제어의 목적함수를 최소화하도록 한다.

$$\begin{aligned} \Delta K_P(t_n) &\propto - \frac{\partial C_p}{\partial K_P(t_n)} \\ &= \frac{\partial C_p}{\partial u(t_n)} \frac{\partial u(t_n)}{\partial K_P(t_n)} \end{aligned} \quad (26)$$

식(26)의 $\frac{\partial C_p}{\partial u(t_n)}$ 와 $\frac{\partial u(t_n)}{\partial K_P(t_n)}$ 는 식(5)와 식(22)에서 구할 수 있다. 따라서 식(26)을 다시 정리

하면 P제어 계인의 학습 식을 구할 수 있다.

$$\Delta K_P(t_n) \approx \eta J^T W_p e(t_{n+1}) e^T(t_n) \quad (27)$$

2.3.2 I 제어 계인의 튜닝

I 제어 계인 수정도 P제어 계인과 같은 방법으로 I 제어의 목적함수를 최소화시키도록 한다.

$$\begin{aligned} \Delta K_I(t_n) &\propto - \frac{\partial C_I}{\partial u(t_n)} \frac{\partial u(t_n)}{\partial K_I(t_n)} \\ &= - \frac{\partial C_I}{\partial e_I(t_{n+1})} \frac{\partial e_I(t_{n+1})}{\partial y(t_{n+1})} \frac{\partial y(t_{n+1})}{\partial u(t_n)} \frac{\partial u(t_n)}{\partial K_I(t_n)} \end{aligned} \quad (28)$$

식(28)을 정리하면 I 제어 계인의 학습 식을 구할 수 있다.

$$\Delta K_I(t_n) = \eta J^T W_I e_I(t_{n+1}) e_I^T(t_n) \quad (29)$$

2.3.3 D 제어 계인의 튜닝

D 제어 계인 수정도 앞과 같은 방법으로 D 제어의 목적함수를 최소화시키도록 한다.

$$\begin{aligned} \Delta K_D(t_n) &\propto - \frac{\partial C_d}{\partial u(t_n)} \frac{\partial u(t_n)}{\partial K_D(t_n)} \\ &= - \frac{\partial C_d}{\partial e_d(t_{n+1})} \frac{\partial e_d(t_{n+1})}{\partial y(t_{n+1})} \frac{\partial y(t_{n+1})}{\partial u(t_n)} \frac{\partial u(t_n)}{\partial K_D(t_n)} \end{aligned} \quad (30)$$

식(30)을 정리하면 D 제어 계인의 학습 식을 구할 수 있다.

$$\Delta K_D(t_n) = \eta J^T W_d e_d(t_{n+1}) e_d^T(t_n) \quad (31)$$

3. 자코비안의 추정

수중의 쌍동선은 공기 부양선에 비해 거친 해상 상태에서도 운동 성능이 우수하기 때문에 차세대 고속선으로써 지속적인 연구 개발이 수행되고 있다^[12]. 수중의 쌍동선은 Fig. 2에서 보는 것처럼 선두부의 수중의 입사각 u_1 과 선미부의 수중의 입사각 u_2 를 조정하여 쌍동선의 상하 진동 변위 y_1 와 피치 각도

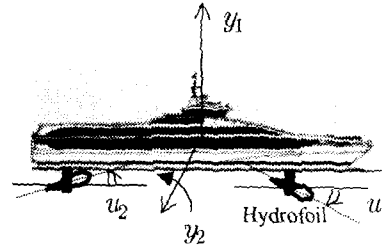


Fig. 2 Schematic diagram of Hydrofoil catamaran

y_2 를 제어한다. 시스템의 모델식을 알면 모델식을 이용하여 자코비안을 구할 수 있지만, 모델식을 모르는 경우에는 실험을 통하여 자코비안을 구할 수 밖에 없다. 따라서 식(4)의 입력에 대한 출력의 자코비안을 식(32)와 같이 실험적 방법에 의해 근사적으로 자코비안을 구한다.

$$\hat{J} = \begin{bmatrix} \frac{y_1^1(t_{n+1}) - y_1(t_n)}{\Delta u_1(t_n)} & \frac{y_1^2(t_{n+1}) - y_1(t_n)}{\Delta u_2(t_n)} \\ \frac{y_2^1(t_{n+1}) - y_2(t_n)}{\Delta u_1(t_n)} & \frac{y_2^2(t_{n+1}) - y_2(t_n)}{\Delta u_2(t_n)} \end{bmatrix} \quad (32)$$

여기서 \hat{J} 는 추정 자코비안이고, $y_1^1(t_n)$ 는 t_n 시간에 입력 변화량 $\Delta u_1(t_n)$ 을 가하였을 때 다음 스텝의 출력이다. 초기 입력을 $u_1(t) = 10^\circ$ 와 $u_2(t) = 10^\circ$ 을 주었으며, Fig. 3과 4에서처럼 4.00초에 $\Delta u_1 = 20^\circ$ 을 주고, 7.00초에 $\Delta u_2 = 30^\circ$ 을 주었을 때의 y_1 과 y_2 의 응답을 Fig. 3과 Fig. 4에 나타내었다. Fig. 3에서 정상 상태 $y_1 = 0.2675[m]$ 이고, Δu_1 의 영향으로 변한

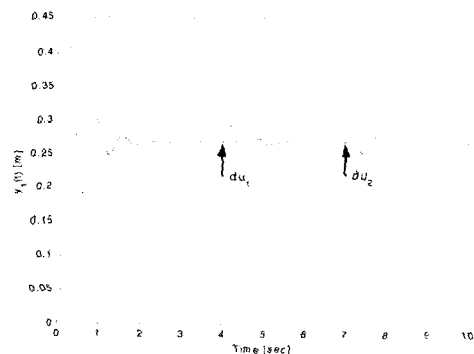


Fig. 3 Vertical distance for step inputs of Hydrofoil catamaran

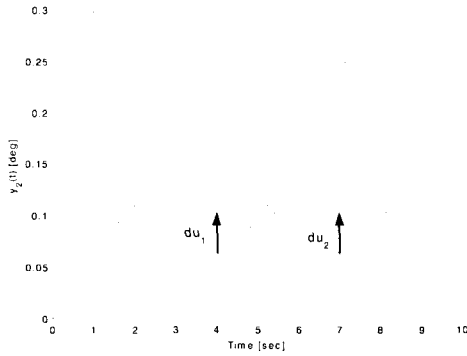


Fig. 4 Pitch angle for step inputs of Hydrofoil catamaran

출력 $y_1^1=0.2692[m]$ 을 구하였으며, Δu_2 의 영향으로 변한 출력 $y_1^2=0.2694[m]$ 을 구하였다. Fig. 4에서 정상 상태 $y_2=0.1055^\circ$ 이고, Δu_1 의 영향으로 변한 출력 $y_2^1=0.0997^\circ$ 을 구하였다. Δu_2 의 영향으로 변한 출력 $y_2^2=0.1165^\circ$ 을 구하였다. 위에서 구한 값을 식 (32)에 대입하여 자코비안을 구하였다.

$$\hat{J} = 10^{-2} \begin{bmatrix} 0.0083 & 0.0063 \\ -0.0290 & 0.0369 \end{bmatrix} \quad (33)$$

4. 시뮬레이션

수중익 쌍동선은 2개의 입력, 2개의 출력과 각 출력의 속도를 가진 4개의 상태 변수를 가진 시스템이다. 식(34)는 수중익 쌍동선의 상태방정식을 나타내었다^[12].

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -109.21 & -4.19 & -0.24 & -5.69 \\ 0 & 0 & 1 & 0 \\ -26.45 & -3.54 & -6.43 & -94.10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1.68 & 1.30 \\ 0 & 0 \\ -5.91 & 7.61 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (34)$$

수중익 쌍동선은 파도, 바람, 등의 해상 환경에 따라 많은 외란(disturbance)을 받게 되므로, 외란을 받고 있을 때 이를 안정화하는 문제에 제안한 제어기 학습을 적용하였다. 학습에 의해 설계된 제어기와 LQR 제어기의 성능을 비교하려고 한다. 비교의 객관성을 위해 LQR 제어기는 다른 논문[12]에서 이미 설계된 바 있는 LQR 제어기 값을 사용하였다. 여기서 제시한 LQR 제어의 상태 가중행렬과 피드백 행렬에 의해 구한 피드백 게인을 식(35)와 (36)에 나타내었다.

$$Q = \begin{bmatrix} 500 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad R = \begin{bmatrix} 1.0 & 0 \\ 0 & 0.8 \end{bmatrix} \quad (35)$$

위 상태 가중치를 이용하여 피드백 게인을 구하면 다음과 같다.

$$K = \begin{bmatrix} 3.06 & 4.76 & -3.16 & -1.68 \\ 4.39 & 3.59 & 3.66 & 2.42 \end{bmatrix} \quad (36)$$

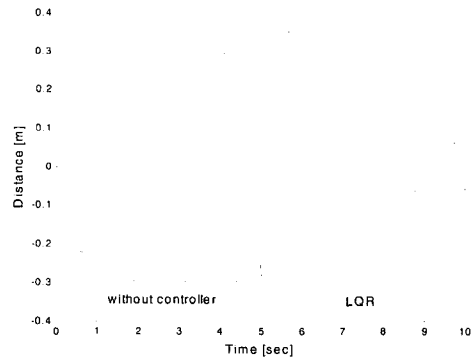


Fig. 5 Vertical distance by LQR

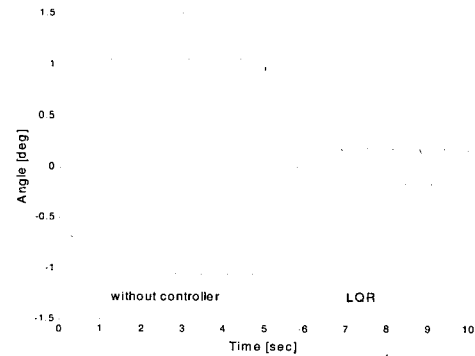


Fig. 6 Pitch angle by LQR

Fig. 5와 6은 외란에 의해 수중익 쌍동선의 흔들림을 나타내었으며 5초 이후에 LQR 제어를 하였을 때의 수중익 쌍동선의 상하 진동 변위와 피치 각도를 나타내었다. 수중익 쌍동선이 2개의 입력과 2개의 출력 시스템이므로 PID 제어 입력은 다음과 같다.

$$\begin{aligned} \begin{bmatrix} u_1(t_n) \\ u_2(t_n) \end{bmatrix} &= \begin{bmatrix} k_{p_{11}} & k_{p_{12}} \\ k_{p_{21}} & k_{p_{22}} \end{bmatrix} \begin{bmatrix} e_1(t_n) \\ e_2(t_n) \end{bmatrix} + \begin{bmatrix} k_{d_{11}} & k_{d_{12}} \\ k_{d_{21}} & k_{d_{22}} \end{bmatrix} \begin{bmatrix} \dot{e}_1(t_n) \\ \dot{e}_2(t_n) \end{bmatrix} \\ &+ \begin{bmatrix} k_{I_{11}} & k_{I_{12}} \\ k_{I_{21}} & k_{I_{22}} \end{bmatrix} \begin{bmatrix} e_{I_1}(t_n) \\ e_{I_2}(t_n) \end{bmatrix} \end{aligned} \quad (37)$$

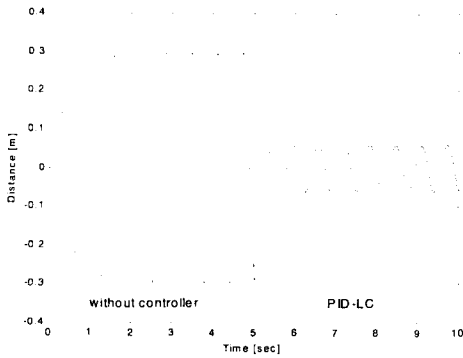


Fig. 7 Vertical distance by PID-LC

$$(W_p = \begin{bmatrix} 10^2 & 0 \\ 0 & 10^2 \end{bmatrix}, W_d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix})$$

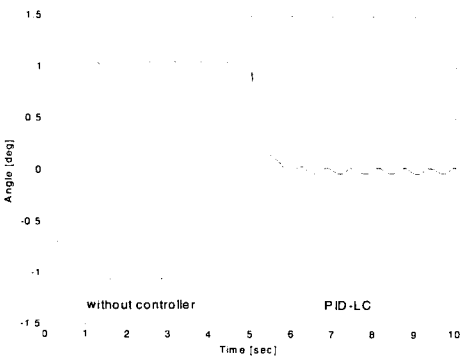


Fig. 8 Pitch angle by PID-LC

$$(W_p = \begin{bmatrix} 10^2 & 0 \\ 0 & 10^2 \end{bmatrix}, W_d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix})$$

자코비안은 시뮬레이션에서 구한 식(33)의 추정 자코비안 \hat{J} 를 사용하였으며, 다중 PID 제어 중에서 PD 제어만 사용하였으며, 가중치는 다음과 같이

하여 학습하였다.

$$\begin{aligned} W_p &= \begin{bmatrix} 10^2 & 0 \\ 0 & 10^2 \end{bmatrix}, W_I = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \\ W_d &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \eta = 1 \end{aligned} \quad (38)$$

학습한 PID 제어기(PID-LC)를 이용한 수중익 쌍동선의 응답을 Fig. 7과 8에 나타내었다. 이 응답과 Fig. 5와 6의 LQR 제어기의 응답과 비교하였을 때 LQR 제어기 보다 더 우수함을 알 수 있었다. 그러나 더 좋은 성능의 제어기를 구하기 위하여 가중치를 다음과 같이 다시 조정하여 학습하였다.

$$W_p = \begin{bmatrix} 10^2 & 0 \\ 0 & 5 \times 10^3 \end{bmatrix}, W_d = \begin{bmatrix} 10^2 & 0 \\ 0 & 1 \end{bmatrix} \quad (39)$$

학습이 52회 이상에서 목적함수가 더 이상 줄어 들지 않으므로 학습은 종료되었으며, 학습한 다중 PID 제어기를 식(40)에 나타내었다.

$$K_p = \begin{bmatrix} 94.47 & -128.88 \\ 73.48 & 164.43 \end{bmatrix}, K_d = \begin{bmatrix} 9.01 & -11.94 \\ 7.01 & 15.23 \end{bmatrix} \quad (40)$$

이 학습한 다중 PID 제어기를 이용한 수중익 쌍동선의 응답을 Fig. 9와 10에 나타내었다. 그림에서 보듯이 수중익 쌍동선의 상하 진동 변위와 피치 각도의 진동이 현저히 줄었으며 매우 빠르게 안정화되는 것을 볼 수 있었다. LQR 제어기의 응답과 비교하였을 때 LQR 제어기 보다 다중 PID 학습이 더 우수함을 알 수 있었다.

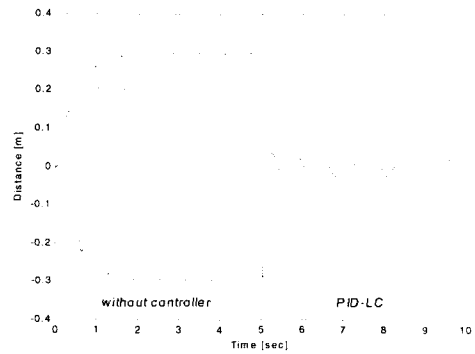


Fig. 9 Vertical distance by PID-LC

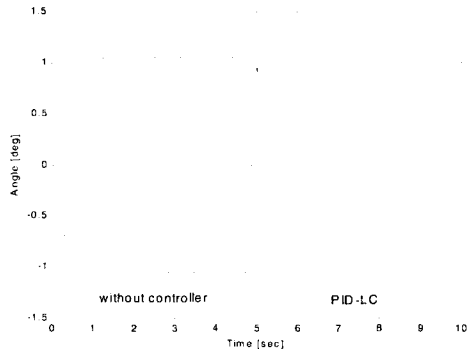


Fig. 10 Pitch angle by PID-LC

5. 결론

본 논문에서는 다중 입출력 시스템 제어에서 운동 방정식을 구하기 어려운 경우에 시스템의 모델을 사용하지 않고 학습에 의해 PID 제어기를 구하는 방법을 제안하였다. 다변수 시스템의 학습을 위해서는 시스템의 자코비안 값이 필요하다는 것을 알았고 자코비안을 이용했을 때, 학습이 수렴한다는 것을 증명하였다. 따라서 모델식을 모르는 시스템에서 실험적으로 자코비안을 구하는 방법도 제안하였다. 제안한 학습법을 일반적인 PID제어기로는 설계하기 어려운 다중 입출력 시스템인 수중익 쌍동선의 외란을 안정화하는 문제에 적용하였다. 먼저 시뮬레이션을 통해 실험적으로 자코비안을 추정하였고, 이를 이용하여 수중익 쌍동선의 다중 PID 제어기를 학습하였다. 학습을 통해 구한 PID 제어기(PID-LC)의 성능을 평가하기 위하여 타 논문^[12]에서 제시한 LQR 제어기와 비교하였다. 그 결과 제안한 PID 학습법에 의해 설계된 PID 제어기의 성능이 모델식을 알고 구한 LQR 제어기보다 우수하다는 것을 확인하였다.

후 기

이 논문은 2002년도 두뇌한국 21 사업에 의하여 지원되었음.

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. Haggglund, "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 gainscheduling of PID controllers," IEE 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," Proc. Instn Mech Engrs, part I, Journal of Systems and Control Engineer, Vol. 212, No. 16, pp. 437-448, 1998.
 6. Maciejowski J. M., "Multivariable feedback design," Addison-Wesley Inc., 1989.
 7. G. F. Franklin, J. D. Powell, and A. Emami-Naeini, "Feedback Control of Dynamic Systems," Addison-Wesley Inc., 1994.
 8. R. C. Dorf and R. H. Bishop, "Modern Control Systems," Addison-Wesley Inc., 1995.
 9. T. J. Procyk, "A Self-Organizing Control for Dynamic Processes," Ph. D. Thesis, Queen Mary College, Univ. of London, 1977.
 10. B. Chung, "Control of nonlinear multivariable Systems using direct fuzzy learning method," Journal of Intelligent & Fuzzy Systems, Vol. 5, pp. 297-310, 1998.
 11. 임윤규, 정병목, "기울기법을 이용한 최적의 PID 제어 학습법," 한국정밀공학회지, 제18권, 제1호, 2001.
 12. 이심용, "수중익 쌍동선의 자세제어 시스템에 관한 이론적 및 실험적 연구," 서울대학교, 박사학위논문, 1999.