

자코비안을 이용한 LQR 제어기 학습법

임윤규*, 정병목#

A Learning Method of LQR Controller Using Jacobian

Yoon Kyu Lim* and Byeong Mook Chung#

ABSTRACT

Generally, it is not easy to get a suitable controller for multi variable systems. If the modeling equation of the system can be found, it is possible to get LQR controller as an optimal solution. This paper suggests an LQR learning method to design LQR controller without the modeling equation. The proposed algorithm uses the same cost function with error and input energy as LQR is used, and the LQR controller is trained to reduce the function. In this training process, the Jacobian matrix that informs the converging direction of the controller is used. Jacobian means the relationship of output variations for input variations and can be approximately found by the simple experiments. In the simulations of a hydrofoil catamaran with multi variables, it can be confirmed that the training of LQR controller is possible by using the approximate Jacobian matrix instead of the modeling equation and this controller is not worse than the traditional LQR controller.

Key Words : LQR(Linear Quadratic Regulator), Learning Control (학습제어), Jacobian (자코비안), Hydrofoil Catamaran (수중익 쌍동선)

1. 서론

현대 제어에서는 다변수 시스템의 제어기를 설계하는 방법으로 상태 변수 피드백 제어기를 많이 사용한다. 상태 변수 피드백 제어기의 대표적인 방법으로는 LQR 제어기가 있으며 이 제어기는 정확한 시스템의 모델식을 이용하여 최적의 제어기를 설계한다.^{1,3} 그러나 산업 현장과 같이 시스템의 모델식을 구하기 어려운 경우에 LQR 제어기를 설계하기란 매우 힘들다. 이러한 경우에는 적응제어를 이용하여 시스템을 모사하는 모델을 구한 다음에 이 모델을 토대로 제어기를

설계하는 방법이 알려져 있지만 시스템을 모사하는 모델과 실제 모델 사이의 에러를 줄이지 못하면 설계된 제어기의 성능이 기대치에 못 미치는 경우가 많다.^{4,8} 따라서 본 논문에서는 모델식을 구하기 어려운 시스템의 제어기를 설계할 때 모델식이나 모사 모델을 사용하지 않고 학습에 의해 LQR 제어기를 설계하는 방법을 제안하고자 한다.

제안한 학습법은 LQR 제어기와 같이 상태변수의 목표값과 결과값 간의 에러와 입력을 목적함수로 정의하고, 이 목적함수를 최소가 되도록 제어기를 수정한다. 이때 목적함수를

접수일: 2004년 10월 25일; 게재승인일: 2005년 5월 13일

* 울산산업진흥재단 전략기획단

교신저자: 영남대학교 공과대학 기계공학부

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

최소화하기 위해서는 시스템의 입력 변화에 대한 출력 변화의 관계를 나타내는 자코비안을 알아야 되므로 실험적으로 구한 값을 사용한다.^{9,12} 따라서 시스템의 자코비안만 구하면 목적함수를 최소화하는 LQR 제어를 실시간으로 구할 수 있다. 시뮬레이션에서는 다변수 시스템인 수중익 쌍동선을 대상으로 하였고 이 시스템을 대상으로 시스템의 근사 자코비안을 실험적으로 대략 구한 다음, 제안한 학습법을 이용하여 LQR 제어를 구한다. 그리고 제안한 학습법의 성능을 객관적으로 평가하기 위해 이심용¹² 이 시스템의 모델식으로부터 구한 LQR 제어기와 본 논문에서 제안하는 학습법에 의한 LQR 제어기의 성능을 서로 비교하여 평가한다.

2. LQR 제어기

LQR(Linear Quadratic Regulator) 제어기는 시스템의 상태방정식을 구속조건으로 하여 주어진 목적함수(cost function)를 최소화하는 최적제어 이론에 기반을 둔 상태 피드백 기법이다.

선형 시스템의 상태 방정식은 식(1)과 같이 표현한다.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (1)$$

여기서 초기 상태 $\mathbf{x}(0)$ 는 주어져 있고 $\mathbf{x}(t) \in R^p$, $\mathbf{u}(t) \in R^m$ 이며 시스템 $\mathbf{u}(t) \in R^m$ 는 안정 가능하다고 가정한다. 그리고 LQR 의 목적함수 C 를 다음과 같이 정의한다.

$$C = \int_0^{\infty} \left[\mathbf{x}(t)^T \mathbf{Q}\mathbf{x}(t) + \mathbf{u}(t)^T \mathbf{R}\mathbf{u}(t) \right] dt \quad (2)$$

여기서 \mathbf{Q} 는 $p \times p$ 인 양의 반한정 대칭행렬인 상태가중행렬이고, \mathbf{R} 은 $m \times m$ 인 양의 한정 대칭행렬인 제어가중행렬이다. 최적제어 문제는 상태방정식의 구속조건에 대해 목적함수를 최소화하는 제어법칙을 찾는 문제로써 최적제어가 유일하게 존재하고, 실시간에서 모든 상태변수를 측정할 수 있다면 최적 제어 법칙을 다음과 같이 피드백 형태로 나타낼 수 있다.

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t) \quad (3)$$

$$\text{여기서 제어게인행렬 } \mathbf{K} = \begin{bmatrix} k_{11} & \dots & k_{1l} \\ \vdots & \ddots & \vdots \\ k_{m1} & \dots & k_{ml} \end{bmatrix} \text{ 는}$$

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} \quad (4)$$

그리고 \mathbf{S} 는 양의 반한정 대칭행렬로서 다음과 같은 제어대수 Riccati 식의 유일해이다.

$$\mathbf{S}\mathbf{A} + \mathbf{A}^T\mathbf{S} + \mathbf{Q} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} = 0 \quad (5)$$

그러나 이 Riccati 식을 푸는 것이 컴퓨터의 도움없이 간단한 일이 아니다.

3. LQR 제어기의 학습법

다변수 시스템의 LQR 제어를 설계하기 위해서는 식(5)의 Riccati 식으로부터 구할 수 있지만 시스템의 모델식을 모르는 경우에는 불가능하다. 따라서 제안하는 학습법은 식(2)의 LQR 제어기법과 같은 목적함수를 정의하고 이를 최소화하도록 LQR 제어기의 게인을 학습한다.

추적 문제(Tracking control)를 고려하여 LQR 제어를 상태변수 대신에 Fig.1 에서처럼 상태 변수의 목표치와 현재 값의 차인 에러를 이용하여 나타내면 이산시간에서 제어 입력은 다음과 같다.

$$\mathbf{u}(k) = -\mathbf{K}\mathbf{e}(k) \quad (6)$$

여기서 $\mathbf{e}(k) = \mathbf{x}(k) - \mathbf{d}(k)$ 상태 변수 에러이다.

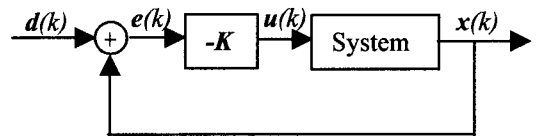


Fig. 1 Block Diagram of LQR control

목표치 $\mathbf{d}(k)$ 가 영이면 식(3)과 같이 안정화 문제에 적용할 수 있으며, 목표치가 시간에 따라 변하거나 임의의 상수벡터이면 추적문제에 적용할

수 있다. 이산시간에서 LQR 의 목적함수를 입력과 상태변수 에러의 함으로 나타내면 다음과 같다.

$$C = \frac{1}{2} \sum_{k=0}^{\infty} [\mathbf{e}^T(k) \mathbf{Q} \mathbf{e}(k) + \mathbf{u}^T(k) \mathbf{R} \mathbf{u}(k)] \quad (7)$$

이 비용함수에서 에러를 최소화하기 위해서는 LQR 제어게인이 다음과 같이 수정되어야 한다.

$$\Delta \mathbf{K} \propto -\frac{\partial C}{\partial \mathbf{K}} \quad (8)$$

실시간 학습을 위해서는 전체의 비용을 한꺼번에 최소화하는 것이 어려우므로 현재의 입력과 그에 따른 출력만을 고려하는 것이 바람직하다. 따라서 k 번째 스텝에서 작용한 입력이 $k+n$ 번째 출력에 가장 큰 영향을 끼쳤다면 k 번째 스텝에서 줄여야 할 비용함수는

$$c(k) = \frac{1}{2} [\mathbf{e}^T(k+n) \mathbf{Q} \mathbf{e}(k+n) + \mathbf{u}^T(k) \mathbf{R} \mathbf{u}(k)] \quad (9)$$

이며 이때, k 번째 스텝에서 LQR 제어게인은 다음과 같이 수정될 수 있다.

$$\Delta \mathbf{K}(k) \propto -\frac{\partial c(k)}{\partial \mathbf{K}} \quad (10)$$

비용함수는 상태변수 에러와 입력의 함수이므로 다음과 같이 편미분할 수 있다.

$$\begin{aligned} \Delta \mathbf{K}(k) &\propto -\left[\frac{\partial c(k)}{\partial \mathbf{e}(k+n)} \frac{\partial \mathbf{e}(k+n)}{\partial \mathbf{K}} + \frac{\partial c(k)}{\partial \mathbf{u}(k)} \frac{\partial \mathbf{u}(k)}{\partial \mathbf{K}} \right] \\ &= -\left[\frac{\partial c(k)}{\partial \mathbf{e}(k+n)} \frac{\partial \mathbf{e}(k+n)}{\partial \mathbf{x}(k+n)} \frac{\partial \mathbf{x}(k+n)}{\partial \mathbf{u}(k)} + \frac{\partial c(k)}{\partial \mathbf{u}(k)} \right] \frac{\partial \mathbf{u}(k)}{\partial \mathbf{K}} \end{aligned} \quad (11)$$

위 식은 식(6)과 (7)을 이용하여 정리하면

$$\Delta \mathbf{K}(k) \propto -\left[\mathbf{e}^T(k+n) \mathbf{Q} \frac{\partial \mathbf{x}(k+n)}{\partial \mathbf{u}(k)} + \frac{\partial c(k)}{\partial \mathbf{u}(k)} \right] \frac{\partial \mathbf{u}(k)}{\partial \mathbf{K}} \quad (12)$$

이 된다. 위 식에서 시스템의 입력과 상태 변수의 관계를 나타내는 자코비안을 정의한다.

$$\begin{aligned} \mathbf{J} &\equiv \frac{\partial \mathbf{x}(k+n)}{\partial \mathbf{u}(k)} \\ &= \begin{bmatrix} \frac{\partial x_1(k+n)}{\partial u_1(k)} & \dots & \frac{\partial x_1(k+n)}{\partial u_m(k)} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_p(k+n)}{\partial u_1(k)} & \dots & \frac{\partial x_p(k+n)}{\partial u_m(k)} \end{bmatrix} \end{aligned} \quad (13)$$

식(12)를 자코비안을 이용하여 다시 표현하면 피드백 게인의 학습 식을 다음과 같이 구할 수 있다.

$$\Delta \mathbf{K}(k) = \eta [\mathbf{J}^T \mathbf{Q} \mathbf{e}(k+n) + \mathbf{R} \mathbf{u}(k)] \mathbf{e}^T(k) \quad (14)$$

여기서 η 는 학습율이다. 그런데 일반적으로 LQR 제어기에서는 입력의 범위를 제한하기 위한 방법으로 입력의 가중치를 이용하지만 학습에 의해 LQR 제어기를 구하는 경우에는 학습 시에 이미 입력이 최대값으로부터 제한을 받기 때문에 \mathbf{R} 에 대해서는 고려할 필요가 없으므로 다음과 같이 수정된 피드백 게인의 학습식을 사용하는 것이 가능하다.

$$\Delta \mathbf{K}(k) = \eta \mathbf{J}^T \mathbf{Q} \mathbf{e}(k+n) \mathbf{e}^T(k) \quad (15)$$

여기서 학습의 안정성을 해석하기 위해서는 수정된 제어게인에 의한 출력에러가 수렴하는지 여부를 알아보는 것으로 충분하다. 식 (6)에 의해 다음 번의 입력은 식 (16) 만큼 차이가 나고 이로 인해 생기는 오차의 변화량은 학습이 무한히 진행되면 에러는 식 (17)과 같이 나타남을 알 수 있다.¹²

$$\Delta \mathbf{u}(k) = \eta \mathbf{J}^T \mathbf{Q} \mathbf{e}(k+n) \mathbf{e}^T(k) \mathbf{e}(k) \quad (16)$$

$$\Delta \mathbf{e}(k+n) = [1 - \eta \mathbf{J} \mathbf{J}^T \mathbf{Q} \mathbf{e}(k) \mathbf{e}^T(k)]^{\infty} \mathbf{e}(k+n) \quad (17)$$

따라서 $e(k+n)$ 이 수렴하기 위해서는 $\Delta e(k+n)$ 이 영이 되어야 하는데 $\mathbf{J}\mathbf{J}^T\mathbf{Q}e(k)e^T(k)$ 가 양의 한정 (positive definite) 대칭행렬이므로 충분히 작은 학습율(η)를 사용하면 학습의 수렴성은 보장된다.¹²

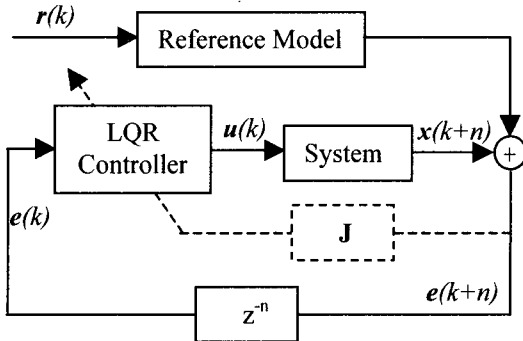


Fig. 2 Block Diagram of LQR learning controller using Jacobian

Fig. 2 는 LQR 제어기 학습의 블록선도를 나타내는데 시스템의 출력지연이 n 스텝이므로 n 스텝 후의 에러와 자코비안을 이용하여 LQR 제어제인을 수정한다. 실제로 에러를 이용하여 입력량을 조정하기 위해서는 역 자코비안을 사용하는 것이 바람직하지만 학습에서는 한 번만에 정확한 값을 찾아가는 것이 아니라 여러 번 반복해서 찾아가기 때문에 역 자코비안을 사용할 필요는 없다. 즉, 여기서는 자코비안을 통해 에러가 줄어드는 방향에 대한 정보만을 이용한다.⁹

4. 자코비안의 추정

수중익 쌍동선은 공기 부양선에 비해 거친 해상 상태에서도 운동 성능이 우수하기 때문에 차세대 고속선으로써 지속적인 연구 개발이 수행되고 있다.¹³ Fig. 3 에서 보는 것처럼 수중익 쌍동선은 선두부의 수중익 입사각 u_1 과 선미부의 수중익 입사각 u_2 를 조정하여 쌍동선의 상하 진동 변위 y_1 와 피치 각도 y_2 를 제어한다.

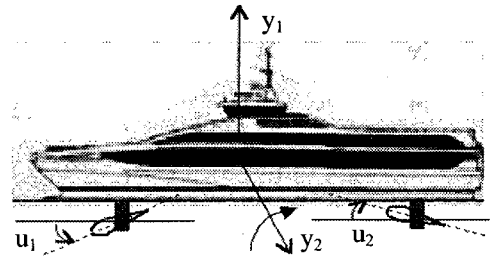
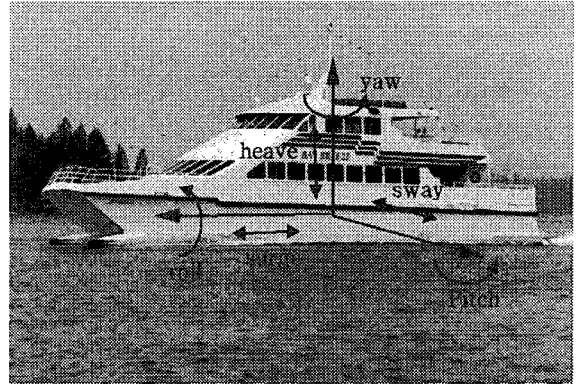


Fig. 3 Hydrofoil catamaran and Schematic diagram

시스템의 모델식을 알면 모델식을 이용하여 자코비안을 구할 수 있지만, 모델식을 모르는 경우에는 실험을 통하여 자코비안을 구할 수밖에 없다. 식(13)의 입력에 대한 출력의 자코비안을 다음의 식(18)을 이용하여 실험적인 방법으로 구한다.

$$\hat{\mathbf{J}} = \begin{bmatrix} \frac{y_1'(k_1+n) - y_1(k_1)}{\Delta u_1(k_1)} & \frac{y_1'(k_2+n) - y_1(k_2)}{\Delta u_2(k_2)} \\ \frac{y_2'(k_1+n) - y_2(k_1)}{\Delta u_1(k_1)} & \frac{y_2'(k_2+n) - y_2(k_2)}{\Delta u_2(k_2)} \end{bmatrix} \quad (18)$$

여기서 $\hat{\mathbf{J}}$ 는 근사 자코비안이고, $y_1'(k_1+n)$ 는 k_1 스텝에서 입력 변화량 $\Delta u_1(k_1)$ 을 가하였을 때 k_1+n 스텝에서의 출력이다. 본 논문에서는 실험이 여의치 않기 때문에 편의상 시뮬레이션을 이용하여 다음과 같이 자코비안을 구했다.

초기 입력을 $u_1 = 10^\circ$ 와 $u_2 = 10^\circ$ 을 주고, 시스템이 어느 정도 정상상태에 이르렀을 때 Fig. 4 와 5 에서처럼 4 초에서 $\Delta u_1(k_1) = 20^\circ$ 를, 7 초에서 $\Delta u_2(k_2) = 30^\circ$ 를 주었다. 출력 y_1 과 y_2 의 응답에서 n 스텝 후의 응답을 측정하는데 그림과 같이

진동하는 경우에는 초기에 변화한 양이나 정상상태에서의 변화량을 측정한다. 즉, Fig. 4 에서 $y_1(k_1) = 0.2675[m]$ 이고, Δu_1 에 의한 출력은 $y_1^1(k_1+n) = 0.2692[m]$ 이다. 그리고 Δu_2 에 의한 출력은 $y_1^2(k_2+n) = 0.2694[m]$ 였다. Fig. 5 에서는 $y_2(k_1) = 0.1055^\circ$ 이고, Δu_1 의 영향으로 변한 출력 $y_2^1(k_1+n) = 0.0997^\circ$ 과 Δu_2 의 영향으로 변한 출력 $y_2^2(k_2+n) = 0.1165^\circ$ 를 얻을 수 있었다.

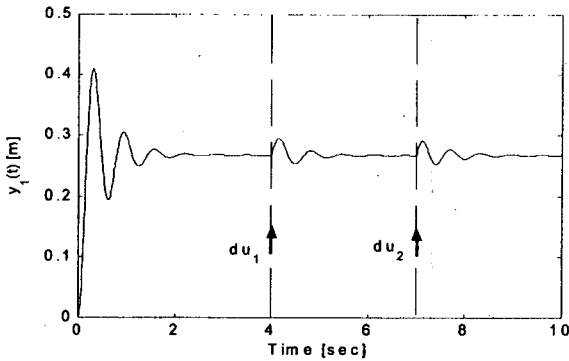


Fig. 4 Vertical distance for step inputs of Hydrofoil catamaran

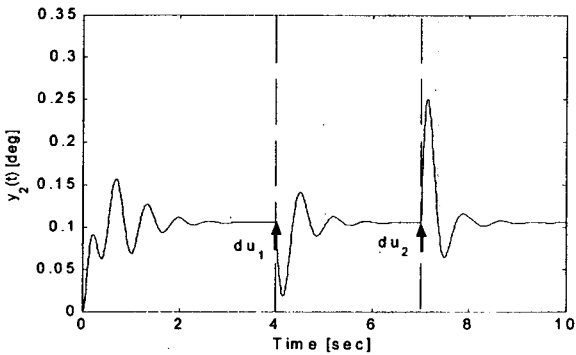


Fig. 5 Pitch angle for step inputs of Hydrofoil catamaran

이렇게 실험적으로 구한 값을 식(18)에 대입하여 근사 자코비안의 값을 구하고 이를 10^4 을 곱하여 정량화하면 식(19)와 같이 나타낼 수 있다.

$$\hat{\mathbf{j}} = \begin{bmatrix} 0.83 & 0.63 \\ -2.90 & 3.69 \end{bmatrix} \quad (19)$$

5. 시뮬레이션

수중의 쌍동선은 2 개의 입력, 2 개의 출력과 각 출력의 속도를 가진 4 개의 상태 변수를 가진 시스템이다. 식(20)과 (21)은 시뮬레이션에서 사용하게 될 수중의 쌍동선의 상태방정식을 나타낸다.¹³

$$\begin{bmatrix} \dot{x}_1 \\ \ddot{x}_1 \\ \dot{x}_2 \\ \ddot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -109.21 & -4.19 & -0.24 & -5.69 \\ 0 & 0 & 1 & 0 \\ -26.45 & -3.54 & -6.43 & -94.10 \end{bmatrix} \begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1.68 & 1.30 \\ 0 & 0 \\ -5.91 & 7.61 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 10\sin(10t) \\ 0 \\ -60\sin(10t) \end{bmatrix} \quad (20)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 & \dot{x}_1 & x_2 & \dot{x}_2 \end{bmatrix}^T$$

(21)

수중의 쌍동선은 파도, 바람, 등의 해상 환경에 따라 많은 외란(disturbance)을 받게되므로, 이러한 외란에 대해 이를 안정화하는 문제에 제안한 LQR 제어기 학습을 적용하였다. 학습에 의해 설계된 제어기와 LQR 제어기의 성능을 객관적으로 비교하는 것이 중요하므로 LQR 제어기는 본 논문에서 별도로 구하지 않고 이심용의 논문¹³에서 설계된 바 있는 LQR 제어기 값을 그대로 사용하였다. 여기서 제시한 LQR 제어의 상태 가중행렬과 이 상태 가중치를 이용하여 구한 피드백 제어 게인 값은 다음과 같다.

$$\mathbf{Q} = \begin{bmatrix} 500 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 1.0 & 0 \\ 0 & 0.8 \end{bmatrix}$$

(22)

$$\mathbf{K} = \begin{bmatrix} 3.06 & 4.76 & -3.16 & -1.68 \\ 4.39 & 3.59 & 3.66 & 2.42 \end{bmatrix} \quad (23)$$

식 (23)의 피드백 계인을 사용하여 제어한 LQR 제어결과를 Fig. 6 과 7 에 나타내었다. 그림에서 보는 바와 같이 비록 LQR 제어를 적용했지만 배의 중심 위치와 피치각도에서의 진동이 상당히 크게 나타나고 있음을 알 수 있다¹²⁻¹³.

수중의 쌍둥선은 4 개의 상태 변수를 가지고 있는 시스템이므로 LQR 제어기의 제어 입력은 식 (24)와 같이 나타낼 수 있다.

$$\begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} = - \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \end{bmatrix} \begin{bmatrix} x_1(k) \\ \dot{x}_1(k) \\ x_2(k) \\ \dot{x}_2(k) \end{bmatrix} \quad (24)$$

$$C = \frac{1}{2} e^T(k) Q e(k) \quad (25)$$

목적함수에 대해서는 식 (7)을 사용하지 않고 식 (25)에서 보는 바와 같이 상태변수 에러의 제곱 항만을 사용했다. 이것은 일반적인 제어 문제에서 입력의 최대값 ($\pm 15^\circ$) 이 제한되어 있으므로 목적함수에 입력비용을 별도로 고려할 필요가 없었기 때문이다. 먼저, LQR 제어기에서 각 상태변수에 대한 가중치를 식 (26)과 같이 선정하여 학습을 시켜보았다. 이때, 학습률 $\eta = 0.01$ 로 두었고 자코비안은 식(18)에서 구한 근사 자코비안 \hat{J} 를 사용하였다.

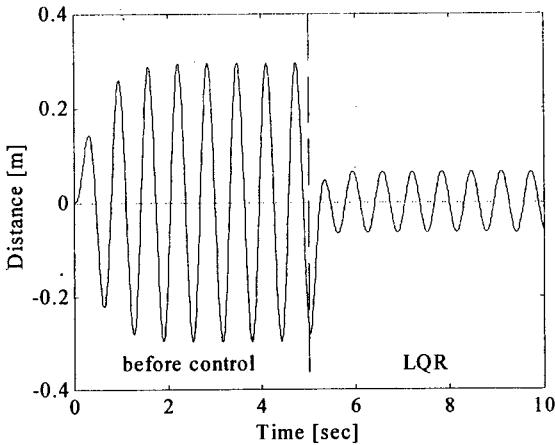


Fig. 6 Vertical distance of LQR

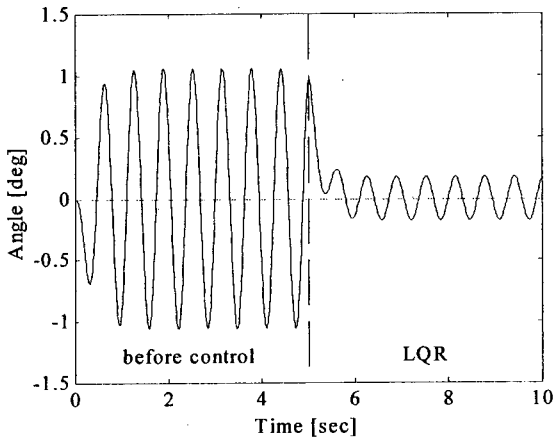


Fig. 7 Pitch angle of LQR

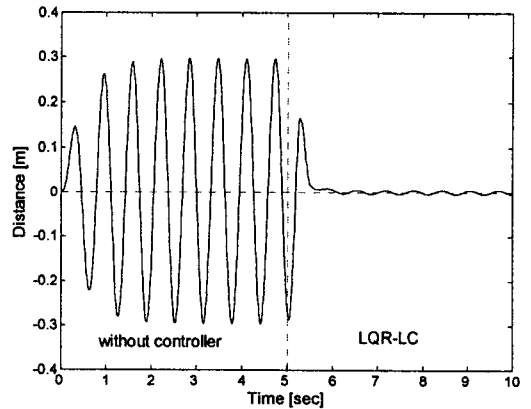


Fig. 8 Vertical distance of LQR-LC by Q_1

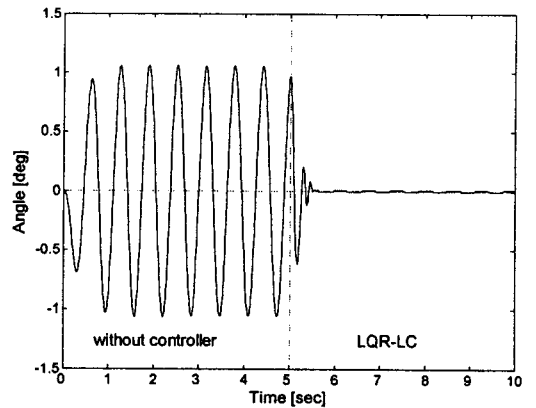


Fig. 9 Pitch angle of LQR-LC by Q_1

$$Q_1 = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (26)$$

과도응답이 LQR 제어에 비해 급격하게 변하는 것을 볼 수 있었고 이를 개선하기 위해 다음과 같이 가중치를 조정하여 다시 학습하였다.

$$Q_2 = \begin{bmatrix} 200 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

이렇게 수직거리에 대한 에러는 가중치를 더 크게 하고 피치각도에 대한 에러에 대해서는 가중치를 더 줄였을 때 식 (27)의 가중치를 이용하여 구한 LQR 제어기는 다음의 식 (28)과 같이 구해졌다.

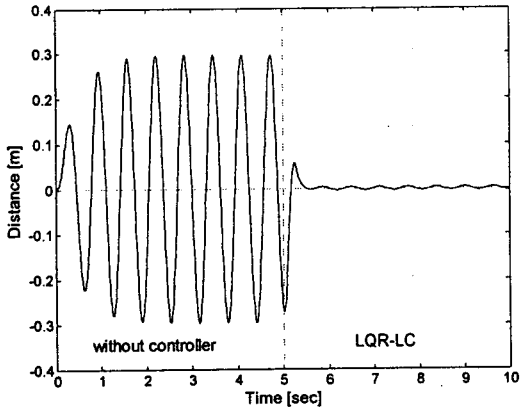


Fig. 10 Vertical distance of LQR-LC by Q_2

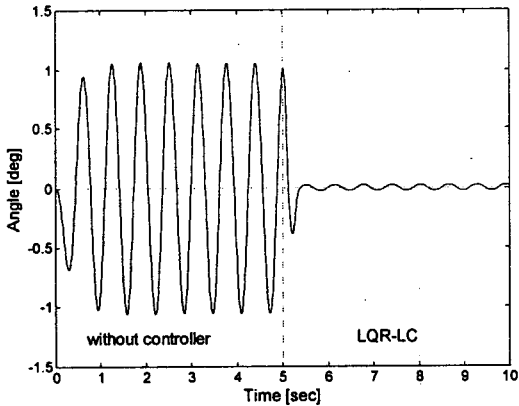


Fig. 11 Pitch angle of LQR-LC by Q_2

학습은 약 70 여 회에서 종료되었고 학습후 LQR 제어기(LQR-LC : LQR Learning Controller)의 응답은 Fig. 8 과 9 에 각각 나타내었다. 그림에서 보는 것처럼 LQR 제어기의 결과와 비교할 때 두 응답 모두 정상상태에서의 진동이 크게 줄었는데 그 이유는 앞에서 언급한 바와 같이 본 학습에서는 LQR 제어와는 달리 입력 에너지를 최소화하지 않고 입력의 최대값에 대한 조건만을 적용한 결과로 해석할 수 있다. 그러나 이 결과에서는 피치각도의

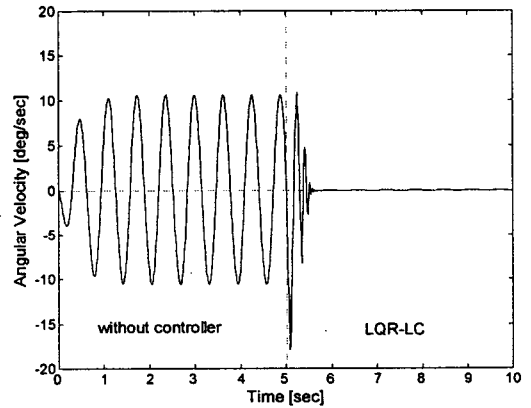


Fig. 12 Angular velocity of LQR-LC by Q_1

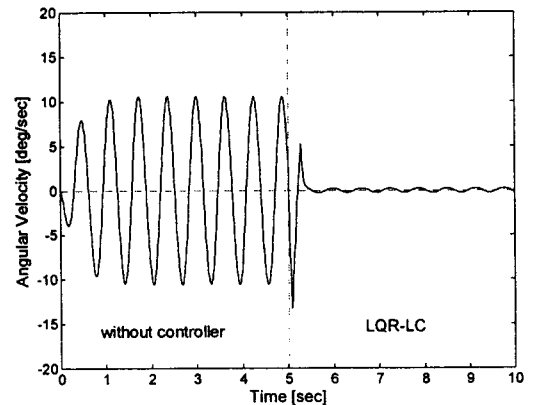


Fig. 13 Angular velocity of LQR-LC by Q_2

$$K = \begin{bmatrix} 753.56 & 74.20 & -140.34 & -6.22 \\ 586.10 & 56.88 & 179.05 & 7.95 \end{bmatrix} \quad (28)$$

Fig. 10 과 11 의 결과를 보면 Q_2 를 사용하여 학습한 제어 결과가 Q_1 을 사용한 것에 비해 전반적으로 나은 결과를 보여주는 것을 알 수 있다. 배의 중심위치 제어에서 오버슈트가 줄어들었고 피치각도도 한번의 진동 후에 안정됨을 알 수 있다. 비록 정상상태에서 약간의 진동이 나타나고 있지만 Fig. 12 와 13 을 비교해 보면 피치각도에 대한 제어 결과가 크게 개선되었음을 보다 분명히 확인할 수 있다. 이러한 결과를 볼 때, LQR-LC 의 제어 결과는 Fig. 6 과 7 에서 보여주는 LQR 결과와 단순 비교를 한다고 해도 LQR-LC 를 통해 전반적으로 아주 좋은 제어기를 설계할 수 있다고 볼 수 있다.

6. 결론

모든 제어기 설계자들은 최적의 제어기를 설계하기 원한다. 그러나 LQR 제어기를 설계하기 위해서는 기본적으로 시스템의 모델식을 알고 있어야만 가능하다. 따라서 본 논문에서는 산업 현장과 같이 모델식을 구하기 어려운 경우에 실시간 학습을 통하여 최적의 LQR 제어기를 설계하는 학습법을 제안하였다. 다변수 시스템에 대한 LQR 제어기를 학습하기 위해서는 시스템의 자코비안 메트릭스를 우선적으로 알아야 하므로 실험적으로 시스템의 자코비안을 구하는 방법을 제시하였다. 그리고 이 방법을 수중익 쌍동선에 대해 적용한 결과 모델식을 알고 구한 LQR 제어기보다 더 좋은 결과를 얻었다. 하지만 이것은 입력 에너지에 대한 조건이 다르기 때문에 절대적으로 낫다고 볼 수는 없다. 하지만 모델식 때문에 LQR 제어기를 사용할 수 없었던 시스템의 제어에 본 논문의 LQR 학습 제어기는 크게 도움이 되리라고 본다.

후 기

이 논문은 2004 년도 두뇌한국 21 사업에 의하여 지원되었음.

참고문헌

1. Maciejowski, J. M., "Multivariable feedback design," Addison-Wesley Inc., 1989.
2. Franklin, G. F., Powell, J. D. and Emami-Naeini, A., "Feedback Control of Dynamic Systems," Addison-Wesley Inc., 2002.
3. Dorf, R. C. and Bishop, R. H., "Modern Control Systems," Addison-Wesley Inc., 1995.
4. Zhang, Y.M. and Kovacevic, R., "Neurofuzzy model-based predictive control of weld fusion zone geometry," IEEE Trans. on Fuzzy Systems, Vol. 6, No. 3, pp. 389-401, 1998.
5. Narendra, K. S. and Parthasarathy, K., "Identification and control of dynamical systems using neural networks," IEEE Trans. Neural Networks, Vol. 1, pp. 4-26, 1990.
6. Sanner, R. M. and Slotine, J. E., "Gaussian networks for direct adaptive control," IEEE Trans. Neural Networks, Vol. 3, pp. 837-863, 1992.
7. Chen, F. C., "Back-propagation neural networks for nonlinear self-tuning adaptive control," IEEE Control System Magazine(Special issue on Neural Networks for Control Systems), Vol. 10, pp. 44-48, 1990.
8. Chen, F. C. and Khalil, H. K., "Adaptive control of nonlinear systems using neural networks," Int. J. of Control, Vol. 55, pp. 1299-1317, 1992.
9. Chung, B. M., "Control of nonlinear multivariable Systems using direct fuzzy learning method," Int. J. of Intelligent & Fuzzy Systems, Vol. 5, pp. 297-310, 1998.
10. Lim, Y.K. and Chung, B.M., "PID learning method using gradient approach for optimal control," J. of the KSPE, Vol. 18, No. 1, pp. 180-186, 2001.
11. Lim, Y.K., Chung, B.M. and Cho, C.S., "Optimal neural network controller design using Jacobian," J. of the KSPE, Vol. 20, No. 2, pp.85-93, 2002.
12. Lim, Y.K. and Chung, B.M., "A learning method of PID controller by Jacobian in multi variable system," J. of the KSPE, Vol. 20, No. 2, pp.112-119, 2003.
13. Lee, S.Y., "Theoretical and experimental study on the attitude control system of foil-catamaran," Ph. D. thesis, Seoul National University, Korea, 1999.