

## 1. 상태방정식의 유도

### 1) 배 경

이미 알고 있듯이, 제어이론은 크게 ‘고전적 방법’과 ‘현대적 방법’으로 양분된다. 고전적 방법은 전달함수, 블록 선도 또는 신호 흐름선도를 이용해 접근하는 것이 일반적이며 이산계를 다룰 때는 z-변환을 사용하여 해석한다. 고전적 방법은 쉽게 풀이되고 도해적인 기법들이 잘 발달되어 현장에서 익숙하게 사용되고 있다. 이에 대응되는 현대적 방법은 주로 운동방정식을 상태변수 표현으로 나타내어 접근하는 것이다.

상태변수 표현을 사용하면, 행렬식으로 나타나므로 수치적인 컴퓨터계산에 자연스럽게 적용되고 다양한 샘플링 방법과 다자유도계 시스템에 대해 통일적인 표현이 가능해지며, 선형 및 비선형 시스템에 대해 모두 적용할 수 있게 된다. 상태방정식을 유도하기 위해서는 우선, 시스템을 표현하는 미분방정식을 상태변수들의 집합인 상태벡터를 정의하여 그 상태벡터의 도함수에 관한 형태로 만들어 주면 된다. 여기에 추가하여 시스템 출력을 나타내기 위하여, 상태변수와 입력에 관계되는 출력방정식을 유도해 낸다. 이상의 상태방정식과 출력방정식을 묶어서 동태방정식이라 한다. 보통은 혼용하여 단지 상태방정식으로만 부르는 경우가 많다.

선형 시불변계의 상태방정식은 일반적으로 다음과 같이 쓰여진다.

$$\begin{aligned}\frac{dx(t)}{dt} &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

여기서  $x(t)$ 는  $(n \times 1)$ 행렬이며 상태벡터라 부르고,  $u(t)$ 는  $(p \times 1)$ 행렬이며 입력벡터라 부른다.  $y(t)$ 는  $(q \times 1)$ 행렬이며 출력벡터라 한다. 또한  $A$ 는  $(n \times n)$ ,  $B$ 는  $(n \times p)$ ,  $C$ 는  $(q \times n)$ ,  $D$ 는  $(q \times p)$ 행렬이며 이상의  $A, B, C, D$  행렬의 모든 요소는 상수이다.

이제 본 도립진자 시스템의 상태방정식을 유도할 수 있게 되었다.

### 2) 상태방정식의 유도

우선,  $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta \\ \phi \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix}$ 으로 상태 벡터를 잡으면, 앞서 구했던 운동 방정식은 다음과 같이 쓸 수 있게 된다.

$$J_{tot}\dot{x}_4 + N\dot{x}_3 + C_{\phi}x_4 = K_m u \quad (\text{식 3-6})$$

$$N\dot{x}_4 + (J_b + J_{ex})\dot{x}_3 + C_{\theta}x_3 - Gx_1 = 0 \quad (\text{식 3-7})$$

간략하게 표기하기 위하여  $S = (J_b + J_{ex})J_{tot} - N^2$  이라 치환하고, (식 3-7)을  $\dot{x}_3$ 에 대해 정리한 후 (식 3-6)에 대입하여  $\dot{x}_4$ 에 대해 다시 정리하면,

$$\dot{x}_4 = \left(-\frac{NG}{S}\right)x_1 + \left(\frac{NC_{\theta}}{S}\right)x_3 + \left(\frac{-(J_b + J_{ex})C_{\phi}}{S}\right)x_4 + \left(\frac{(J_b + J_{ex})K_m}{S}\right)u \quad (\text{식 3-8})$$

이고,

(식 3-6)을  $\dot{x}_4$ 에 대해 정리한 후 (식 3-7)에 대입하여  $\dot{x}_3$ 에 대해 다시 정리하면,

$$\dot{x}_3 = \left(\frac{J_{tot}G}{S}\right)x_1 + \left(\frac{-J_{tot}C_{\theta}}{S}\right)x_3 + \left(\frac{NC_{\phi}}{S}\right)x_4 + \left(\frac{-J_{tot}NK_m}{S}\right)u \quad (\text{식 3-9})$$

이다. 또한 상태 벡터의 정의에서

$$\dot{x}_1 = x_3, \quad \dot{x}_2 = x_4 \quad (\text{식 3-10})$$

이므로 (식 3-8, 9, 10)을 정리하여 행렬식으로 쓰면

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{J_{tot}G}{S} & 0 & -\frac{J_{tot}C_\Theta}{S} & \frac{NC_\Phi}{S} \\ -\frac{NG}{S} & 0 & \frac{NC_\Theta}{S} & -\frac{(J_b+J_{ex})C_\Phi}{S} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{J_{tot}NK_m}{S} \\ \frac{(J_b+J_{ex})K_m}{S} \end{bmatrix} u \quad (\text{식 3-11})$$

이고 이를 간략하게 표시하기 위해 각 요소를

$$\begin{aligned} k_1 &= \frac{1}{S}(J_{tot}G) & k_2 &= \frac{1}{S}(-J_{tot}C_\Theta) \\ k_3 &= \frac{1}{S}(NC_\Phi) & k_4 &= \frac{1}{S}(-NG) \\ k_5 &= \frac{1}{S}(NC_\Theta) & k_6 &= \frac{1}{S}[-(J_b+J_{ex})C_\Phi] \\ k_7 &= \frac{1}{S}(-J_{tot}NK_m) & k_8 &= \frac{1}{S}[(J_b+J_{ex})K_m] \end{aligned}$$

으로 치환하여 최종 상태방정식을 구하면,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ k_1 & 0 & k_2 & k_3 \\ k_4 & 0 & k_5 & k_6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ k_7 \\ k_8 \end{bmatrix} u \quad (\text{식 3-12})$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \quad (\text{식 3-13})$$

이 된다. 위 형태는 선형시불변계에 관한 상태방정식의 일반형과 같은 꼴이다.

## 2. 상태방정식의 이산화

### 1) 도 입

본 도립진자 시스템은 시불변의 시스템이라고 할 수 있다. 그리고 사용된 계측장치인 2개의 엔코더는 양자화(quantization)된 값을 주는 디지털 장치이다. 또한 그 값을 연속적으로 받아들이는 것이 아니고 1.0kHz의 인터럽트를 발생시켜 입력값을 샘플링하는 이산계인 것이다. 이렇게 아날로그적인 자연계를 이산화시켜 인식하는 이유는 디지털 컴퓨터를 제어기부로서 사용하기 때문이다. 이에 따른 잇점은 계산시간의 단축, 잡음의 제거, 비용의 절감등이 있다. 여기서 우리는 ‘시간에 관한 이산화’ 과정을 수행한다. 이는 물론 시스템이 시간에 종속적인 특성을 가진 방정식으로 표현되기 때문이다. 이산화 방법은 유일하지 않고 크게 두 가지 방법이 존재한다. 상태천이행렬을 사용한 방법과 이산시간 근사화 방법이 그것이다<sup>1)</sup>. 여기서는 이산시간 근사화 방법에 관하여 풀이를 하여 본다. 그전에 먼저 샘플된 데이터를 취급하는 개념을 잡기 위해 필터링과 데이터 복원에 관한 설명을 한다.

### 2) ZOH의 적용<sup>2)</sup>

보통의 경우, 샘플링 조작으로 획득된 신호 내의 고주파 성분을 없애기 위해 저주파 통과 필터나 데이터 복원장치가 필요해 진다. 사용되는 필터중 가장 일반적인 것은 샘플-홀드 장치이다. 이것은 단순하고 저렴하게 구현할 수 있어 가장 널리 사용되는 필터링 장치이다. 우리는 데이터 복원장치로도 역시 홀드장치를 사용한다. 이산화된 데이터를 원래의 신호로 복원한다는 것은, 주어진 이산치 데이터만을 바탕으로 이루어져야 하므로 단순히 직전에 읽은 값을 유지하는 것, 즉 홀드가 효과적임을 알 수 있다. 이러한 복원 과정은, 모든 과거의 샘플링 순간에 대한 데이터를 기초로 한 먹금수 전개를 바탕으로 한다. 즉 복원된 데이터  $f_n(t)$ 는 다음 식으로 구할 수 있다.

1) 최신 제어시스템 제 7 판 (반도출판사, Richard C. Dorf, Robert H. Bishop 저, 박홍배, 이균경 공역)

2) 디지털 제어시스템 제 2 판 (화성출판사, Benjamin C. Kuo 저, 박진배, 어진우, 백윤수, 최윤호 공역)

즉

$$f_n(t) = f(nT) + f^{(1)}(nT)(t-nT) + \frac{f^{(2)}(nT)}{2!}(t-nT)^2 \dots \quad (\text{식 3-14})$$

이고, 여기서

$$f_n(t) = f(t) \quad \text{for} \quad nT \leq t < (n+1)T \quad (\text{식 3-15})$$

$$f^{(m)}(nT) = \left. \frac{d^{(m)}f(t)}{dt^m} \right|_{t=nT} \quad (m = 1, 2, \dots)$$

으로 한다. 그러나 (식 3-14)를 그대로 적용하여 데이터를 복원하기에는 문제가 있다. 왜냐하면 이산화된 데이터를 기초로 한 도함수  $f^{(m)}(nT)$ 는 극한적인 미분이 아닌 유한히 작은 시간의 차분식이 되고, 이는 필연적으로 시간지연을 야기하기 때문이다. 즉,

$$\begin{aligned} f^{(1)}(nT) &= \frac{1}{T}[f(nT) - f((n-1)T)] \\ f^{(2)}(nT) &= \frac{1}{T^2}[f(nT) - 2f((n-1)T) + f((n-2)T)] \\ &\vdots \end{aligned}$$

이므로, 도함수의 차수가 높아질수록 필요한 지연펄스의 수가 많아짐을 의미하고 이는 곧 실시간의 페루프 제어시스템의 안정성에 악영향을 주게 된다. 일반적으로,  $f^{(m)}(nT)$ 를 위해 필요한 펄스의 데이터 수는  $(m+1)$ 개다.

이상 먹급수 전개를 바탕으로 원래의 정보를 복원하는 조작을 ‘외삽’이라 하는데, 외삽의 정밀도를 높이기 위해 도함수의 차수를 높이려는 시도는 상당히 신중해 질 수 밖에 없다. 왜냐하면 실시간의 페루프 제어 시스템에 고차의 도함수를 포함한 외삽을 실현하려면 고주파의 샘플링과 높은 분해능, 그에 따른 빠른 처리속도가 필요해 지는데 이는 개발 및 제작 비용의 상승을 초래하는 요인이기 때문이다. 그러한 이유로 인해 우리는 ZOH(영차홀드)를 사용한다. 영차홀드란 데이터 복원을 위해 먹급수의 첫째 항인 0차의 도함수만을 사용하는 것을 말한다. 따라서 ZOH를 적용한 데이터의 복원식은 다음과 같이 된다.

즉

$$f_n(t) = f(nT)$$

이고, 여기서

$$f_n(t) = f(t) \quad \text{for} \quad nT \leq t < (n+1)T$$

으로 한다.

영차홀드는 중첩의 원리를 만족하는 선형장치의 특성을 가짐에 유의한다. 참고적으로, 본 시스템은 입력받는 변위 데이터  $\theta$ 와  $\phi$ 는 영차 홀드를 사용하여 시간지연이 발생하지 않지만, 각속도인  $\dot{\theta}$ 와  $\dot{\phi}$ 는 두 번의 변위 데이터를 단위 시간  $T$ 로 나누어 사용하므로 2번의 펄스만큼 시간지연이 발생한다. 또한 C 프로그램 상에서 모터에 출력되는 데이터 값은 인터럽트가 걸릴때마다 출력되는 것이 아니고 무한루프가 한 번 돌때마다 출력되므로 엄밀하게 1.0kHz로 출력되는 것은 아니다. 이는, 컴퓨터 수행속도의 한계로 인한 것으로 부득이한 것이었음을 밝힌다. 한 번의 인터럽트가 걸릴때마다 입력과 제어를 위한 연산, 출력 그리고 디스플레이 등을 수행하는등의 작업을 컴퓨터가 해낼 수 없었던 것이다. 그러나 크게 보아서 제어시스템의 불안정성에 눈에 띄게 영향을 미쳤던 것 같지는 않다.

### 3) 이산시간 근사화 방법에 의한 상태방정식의 이산화

이산시간 근사화(Discrete-time approximation)는 시간축을 충분히 작은 시간의 증가분으로 나눈다고 생각하는 것에 기초한다. 다음의 선형상태방정식을 상기한다.

$$\dot{x} = Ax + Bu$$

이때 미분항  $\dot{x}$ 는 시간의 증가분  $T$ 를 고려한 다음 차분식으로 쓰여질 수 있다.

$$\dot{x} = \frac{x(t+T) - x(t)}{T}$$

이를 상태방정식에 대입하면

$$\frac{x(t+T) - x(t)}{T} \cong Ax(t) + Bx(t)$$

이고, 이를 다시 정리하면

$$x(t+T) \cong TA x(t) + x(t) + TBu(t) \cong (TA + I)x(t) + TBu(t)$$

를 얻는다. 여기에  $t = nT$  라고 하여 대입하면,

$$x[(n+1)T] \cong (TA + I)x(nT) + TBu(nT) \quad (\text{식 3-29})$$

로 된다. 그러므로,

$$A_d = TA + I \quad (\text{식 3-30})$$

$$B_d = TB \quad (\text{식 3-31})$$

로 할 수 있게 된다. 이를 Euler 방법이라 하며 반복적인 연속계산에 간단히 적용할 수 있으므로 디지털 컴퓨터 연산에 적용하기에 편리하다. 실제로 MATLAB의 Control toolbox에 포함된 c2d( ) 함수는 이 방법을 사용한다. 그리고 본 도립진자 시스템의 설계에서도 c2d( ) 함수를 사용하여 제어기를 구하였다.

### 3. 디지털 LQ 최적제어기의 설계

#### 1) 도입

제어시스템의 해석과 설계에 있어 중요한 역할을 하는 요소는 보통 안정도(Stability), 가제어성(Controllability), 가관측성(Observability)의 개념이다. 이러한 부분에 관한 자세한 사항들은 여기서 생략하도록 한다.

도립진자 시스템을 위해 적용하고자 하는 제어기는 일반적이며 실제적으로 적용할 수 있는 기법의 하나인 최적 선형 레귤레이터이다. 레귤레이터란, 여기서는 영의 기준입력을 가진 시스템을 기준으로 정의되며 설계목적은 상태 또는 출력을 평형상태 근처로 이동시키는 것이다. 이때 영입력이라는 조건은 엄격한 제한 사항은 아니라고 한다. 왜냐하면 선형 레귤레이터의 결과는 시스템이 안정한 감쇠특성을 가지도록 보증하므로 요구되는 성능을 만족하기 때문이다.

#### 2) 성능지수의 표시 및 최적제어입력의 유도

최적제어를 위해서는 적당한 성능지수를 선정할 필요가 있다. 성능지수는 일반적으로 시스템의 성능을 나타내는 일반적인 척도이며 시스템의 중요한 규격을 강조할 수 있도록 선정한다. 성능지수가 최소치인 최적치에 도달하도록 시스템의 파라미터가 조정되었을 때 이를 최적제어시스템이라 한다. 대표적인 성능지수로 오차자승적분(ISE : integral of the square of the error)이 있으며, 시스템의 오차를  $e(t)$ 라 하면 다음과 같이 정의된다.

$$J_{ISE} = \int_0^T e^2(t) dt$$

상한의  $T$ 는 적분이 정상상태 값으로 수렴하도록 어느정도 임의로 정한 유한한 시간이다. 예를들어 단위 스텝입력에 대한 응답곡선을 기반으로 정상상태와의 오차를 자승하여 적분하면, 이 역시 어떤 값으로 수렴하게 될 것이다. 자승을 취하는 이유는, 선별도를 높이고 성능지수 자체의 값이 항상 양수를 갖도록 하기 위함이다. 위 정의식에서 영상태에 대한 오차를 우리가 선정한 상태변수로 정한다면 다음과 같이 나타낼

수 있다.

$$\begin{aligned}
 J &= \int_0^T (x_1^2 + x_2^2 + x_3^2 + x_4^2) dt \\
 &= \int_0^T [x_1 \ x_2 \ x_3 \ x_4] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} dt \\
 &= \int_0^T x^T x dt
 \end{aligned}$$

그런데, 본 도립진자 시스템은 제어신호  $u$ 의 크기가 24V로 제한되어 있고 또한 허용전류는 1.5A로 제한되어 있으므로 가해 줄 수 있는 에너지의 양은 한계가 있다. 따라서  $u$ 에 관한 항을 성능지수에 포함시켜 고려해줄 필요가 있다. 만일 그러지 않고 위 성능지수를 그대로 사용한다면 결과로 얻어지는 파라미터가 무한대로 나올 수가 있다. 따라서 다음의 성능지수로 다시 정의한다.

$$J = \int_0^\infty (x^T Q x + R u^2) dt \quad (\text{식 4-1})$$

여기서  $Q$ 와  $R$ 은 각 상태변수를 위한 가중치로 설정할 수 있으며  $Q$ 에 대한  $R$ 의 크기는 시스템 전력소모의 중요도와 대비되어 선택될 수 있다. 역시 대각행렬인  $Q$ 는 각각의 상태변수를 위한 가중치로 선택될 수 있다. 이 가중행렬들은 시뮬레이션을 통해 적당히 조정될 수 있을 것이다. 본 도립진자 시스템의 경우에는 행렬  $Q$ 는 단위행렬로 쓰여져 수식의 복잡성을 피하고 시스템의 전력소모에 관해 관심을 집중시켜 보았다. 왜냐하면  $\pm 24V$ 의 한계로 인해 전력공급능력에 제한이 있기 때문이다. 이제 성능지수가 결정되었으면 본격적인 선형제한제어기의 설계로 들어가 본다. 사용되는 시스템은 다음으로 표현됨을 다시 한 번 상기한다.

$$\dot{x} = Ax + Bu \quad (\text{식 4-2})$$

또한 목표하는 제어기는 다음의 선형 부폐환 함수로 정한다.

$$u = -Kx \quad (\text{식 4-3})$$

여기서 파라미터로 사용되는 Marix gain  $K$ 는

$$K = [k_1 \ k_2 \ k_3 \ k_4]$$

으로 주어지게 된다. 그러므로 (식 4-3)의 선형제어기가 들어간 제어시스템은 (식 4-3)을 (식 4-2)에 대입함으로써 표현할 수 있다. 즉

$$\dot{x} = Ax - BKx = (A - BK)x \quad (\text{식 4-4})$$

여기서 일단

$$D = A - BK \quad (\text{식 4-5})$$

로 한다. 그리고 제어기가 들어간 시스템의 성능지수를 위해 성능지수  $J$ 에 (식 4-3)을 대입해 보면

$$\begin{aligned}
 J &= \int_0^\infty (x^T Q x + R u^2) dt \\
 &= \int_0^\infty (x^T Q x + R u^T u) dt \\
 &= \int_0^\infty [x^T Q x + R (Kx)^T (Kx)] dt
 \end{aligned}$$

$$= \int_0^{\infty} [x^T(Q + RK^TK)x]dt \quad (\text{식 4-6})$$

이 된다. 그리고 여기에서

$$W = Q + RK^TK \quad (\text{식 4-7})$$

이라 치환하면,

$$J = \int_0^{\infty} (x^TWx)dt \quad (\text{식 4-8})$$

로 주어진다. 이제 제어시스템에 관한 천이식(식 4-4)와 성능지수(식 4-8)이 결정되었다고 할 수 있다. 그리고, 이제

$$\frac{d}{dt}(x^TSx) = -x^TWx \quad (\text{식 4-9})$$

로 가정해 본다. 이때 사용된 대칭행렬  $S$ 는 아무런 조건없이 위 대수방정식을 간략히 하기 위해 쓰여진 것이다.  $S$ 가 대칭행렬이므로  $S_{ij} = S_{ji}$ 임을 기억해 둔다.

우선 (식 4-9)의 좌변을 미분해 보면,

$$\frac{d}{dt}(x^TSx) = \dot{x}^TSx + x^T\dot{S}x$$

이다. 여기에 앞서 구했던 천이식 (식 4-4)를 대입하면

$$\begin{aligned} \frac{d}{dt}(x^TSx) &= (Dx)^TSx + x^TS(Dx) = x^TD^TSx + x^TSDx \\ &= x^T(D^TS + SD)x \quad (\because (Dx)^T = x^TD^T) \end{aligned}$$

이고, 여기에서

$$D^TS + SD = -W \quad (\text{식 4-10})$$

이라면, (식 4-9)의 가정에 맞게 되므로 (식 4-9)를 제어시스템의 성능지수식인 (식 4-8)에 대입할 수 있다. 대입하면

$$\begin{aligned} J &= \int_0^{\infty} -\frac{d}{dt}(x^TSx)dt \\ &= -x^TSx \Big|_0^{\infty} \\ &= x^T(0)Sx(0) - x^T(\infty)Sx(\infty) \end{aligned}$$

이 된다. 그런데 이 성능지수는  $x(\infty) \rightarrow 0$  으로 되는 안정한 제어시스템을 나타낸 것이므로

$$J = x^T(0)Sx(0) \quad (\text{식 4-11})$$

으로 주어진다.

이제 (식 4-8)에서  $J$ 를 최소화하는  $S$ 를 찾으면 제어시스템의 최적시스템을 설계할 수 있게 된다. 우선 적당한  $Q$ 와  $R$ 을 결정한 후에 앞에서 구한 (식 4-7)과 (식 4-10)을 이용하면 미지의 파라미터  $K$ 를 포함한 행렬  $S$ 를 유도할 수 있을 것이다. 이  $S$ 를 제어시스템의 성능지수인 (식 4-8)에 대입하면 최종적으로 성능지수  $J$ 는 파라미터  $K$ 에 관한 방정식으로 된다.  $K$ 는 4개의 요소를 가진 행렬이므로 실제로는 4개의 변수를 가진 성능지수가 결정되는 것이고,  $J$ 를 최소화하는 4개의 변수를 결정하면 될 것이다. 최소치를

구하려면  $J$ 를  $K$ 에 관해 미분한 값이 0인 지점을 찾으면 된다. 그런데 4개의 변수를 가진 방정식을 푸는 것은 컴퓨터를 이용하는 것이 현명하며 이를 위해 MATLAB의 Control toolbox에 `lqr()`과 `dlqr()` 함수가 준비되어 있다. 이산계를 위한 MATLAB의 `dlqr()` 함수는 이를 위한 성능지수로

$$J = \sum_{n=0}^{\infty} [x^T Q x + u^T R u + 2x^T N] \quad (\text{식 4-12})$$

를 사용한다. 행렬  $N$ 은 무시될 경우, 영행렬로 취급된다. 이후에 다음의 Riccati equation의 해인  $S$ 와 페루프의 고유치  $E$ , 그리고 Matrix gain값  $K$ 를 구한다.

$$A^T S A - S - (A^T S B + N)(R + B^T S B)^{-1} (B^T S A + N^T) + Q = 0 \quad (\text{식 4-13})$$

$$E = \text{EIG}(A - BK) \quad (\text{식 4-14})$$

사용 형식은 다음과 같다.

`[K,S,E]=dlqr(Ad,Bd,Q,R,N)`

### 3) 독립진자 시스템에의 적용결과

MATLAB 프로그램을 이용하여 LQ 제어를 설계하였다. 독립진자의 길이는 1m 이고, 질량은 30g이다.

또한 LQR을 위한 가중행렬은 각각  $Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ ,  $R = 0.05$ 로 잡았다. 또한 샘플링 시간은

0.001sec 로 하였다.

아래에 사용된 MATLAB 프로그램의 리스트를 보였다.

독립진자 시스템의 상태방정식을 이산화하고 LQR의 Matrix gain K를 찾는 프로그램

```
% LQ 제어기 설계 프로그램
clear
g=9.81;
mb=30e-3;
L=1000e-3;
r=122e-3;
km=0.0189;
Cphi=3.15e-3;
Ctheta=0.0;
Jtot=0.0026;
Jb=(mb*L^2)/3;
Jex=9.1875e-6;
N=-0.5*mb*L*r;
G=0.5*L*mb*g;
S=(Jb+Jex)*Jtot-N^2;
k1=(Jtot*G)/S;
k2=(-Jtot*Ctheta)/S;
k3=(N*Cphi)/S;
k4=(-N*G)/S;
k5=(N*Ctheta)/S;
k6=(-(Jb+Jex)*Cphi)/S;
k7=(-Jtot*N*km)/S;
k8=((Jb+Jex)*km)/S;
A=[zeros(2) eye(2); k1 0 k2 k3; k4 0 k5 k6];
B=[0; 0; k7; k8];
C=[eye(4)];
D=zeros(4,1);
pzmap(A,B,C,D);
Q=eye(4);
R=0.05;
[Ad,Bd]=c2d(A,B,0.001);
Cd=C;
Dd=D;
[K,S,E]=dlqr(Ad,Bd,Q,R);
printsys(Ad,Bd,Cd,Dd)
```

프로그램의 실행 결과는 다음과 같다.

### 프 로 그 램 실행 결 과

K =  
-809.6896    -4.3723   -197.5417    5.4039

S =  
1.0e+006 \*  
4.1731    0.0451    1.0159   -0.0054  
0.0451    0.0015    0.0110    0.0000  
1.0159    0.0110    0.2475   -0.0013  
-0.0054    0.0000    -0.0013    0.0000

E =  
0.9634  
0.9958  
0.9960  
0.9990

a =

	x1	x2	x3	x4
x1	1.00001	0	0.00100	-1.27053e-007
x2	5.93518e-006	1.00000	1.97862e-009	0.00100
x3	0.01687	0	1.00001	-0.00025
x4	0.01187	0	5.93518e-006	0.99861

b =

	u1
x1	1.62957e-009
x2	4.16948e-006
x3	2.90591e-006
x4	0.00834

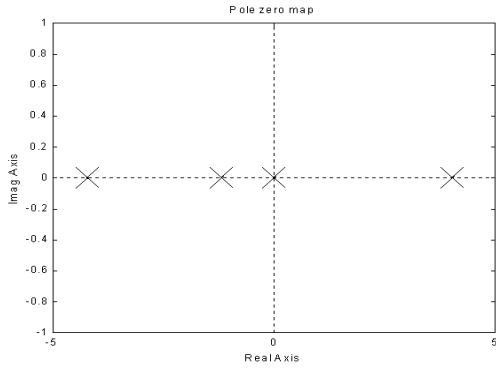
c =

	x1	x2	x3	x4
y1	1.00000	0	0	0
y2	0	1.00000	0	0
y3	0	0	1.00000	0
y4	0	0	0	1.00000

d =

	u1
y1	0
y2	0
y3	0
y4	0



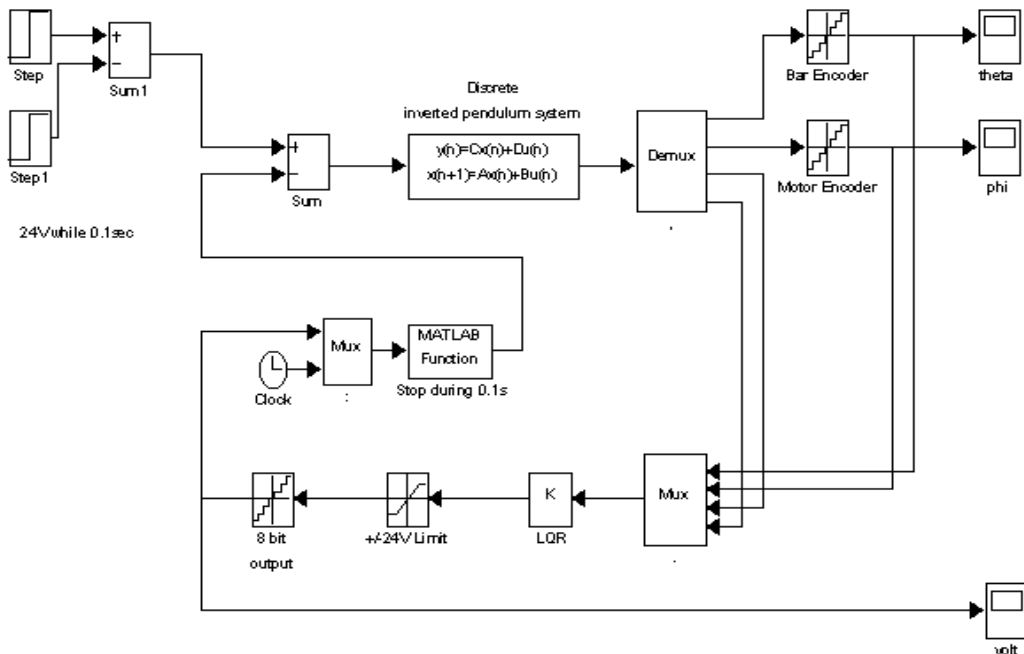


도립전자시스템의 PZMAP

PZMAP 그래프를 이용해 모델링된 상태공간의 점을 보면, 좌표계의 우반부에 극점이 존재하고 있으므로 그 자체로 불안정한 시스템임을 알 수 있다. 제어기의 역할은 불안한 위치에 존재하는 극점을 안정권에 들어가도록 옮겨주는 것이고, 극점의 이동이 성공적이라면 잘 설계된 제어기라 할 수 있을 것이다. 물론 실제 물리계에서 잘 동작할 것이 기대된다. 이제 앞 페이지에서 출력된 Matrix gain값  $K$ 를 이용하여 시뮬레이션 및 실제 실험을 할 수 있게 되었다.

#### 4. 시뮬레이션

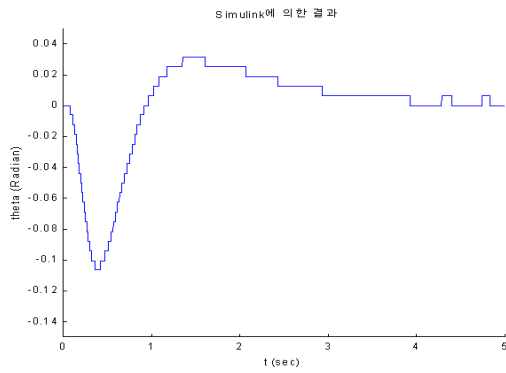
다음 그림은 다른 외란은 무시된 상태에서 0 sec 에서 0.1 sec 동안 24 Volt 의 외란을 가했을 경우의 시뮬레이션 모델이다. 상태변수들과 Matrix Gain값  $K$ 는 앞에서 구한 값을 사용하였고, 샘플링 주기는 0.001 sec 로 하였다. Scope 에 출력되는 데이터들은 MATLAB Work space의 변수 theta 와 phi 로 각각 저장되도록 하여, 그 데이터를 이용해 그래프를 그릴 수 있다. 될 수 있는대로 실제 시스템과 비슷하게 만들기 위해 양자화(Quantizer)블록을 사용하여 각 엔코더와 출력 전압의 분해능대로 설정하였다. 또한 간단한 Saturation 블록을 사용하여 출력 전압의 한계를 24V로 제한하여 주었다. Clock에 연결된 MATLAB Function 블록은 초기외란이 들어올 때 부제환입력신호를 0으로 강제 출력하여(제어기가 동작하지 않음) 실제 실험과 같은 조건을 주도록 하였다.



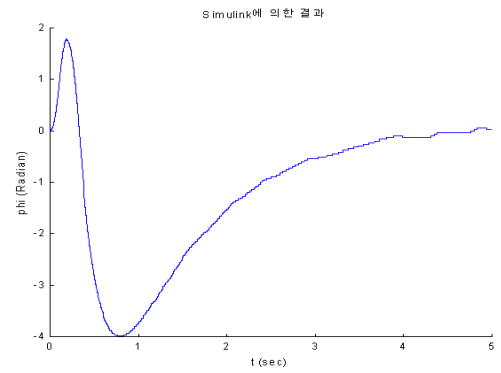
도립전자 제어시스템의 시뮬링크 모델

위의 Simulink 모델로 수행한 시뮬레이션 결과는 다음과 같다.

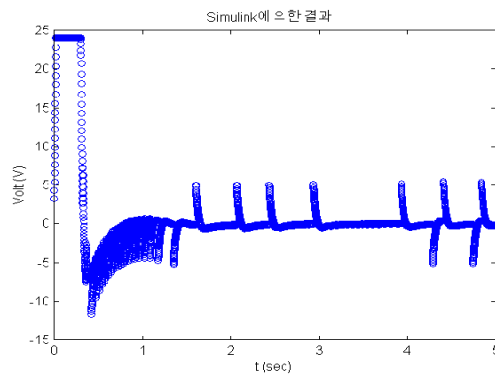
## < 시뮬레이션 결과 >



Bar의 변위



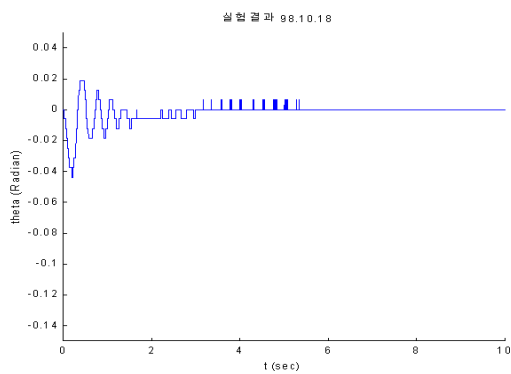
Motor의 변위



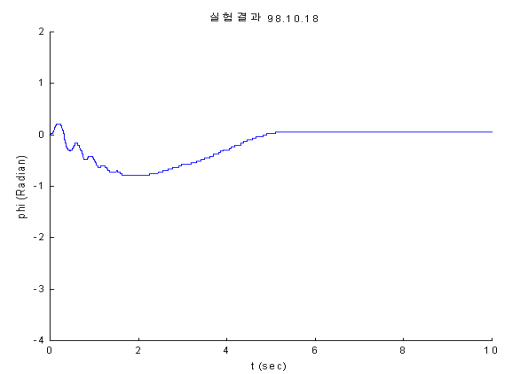
출력전압

## 5. 실제 실험결과

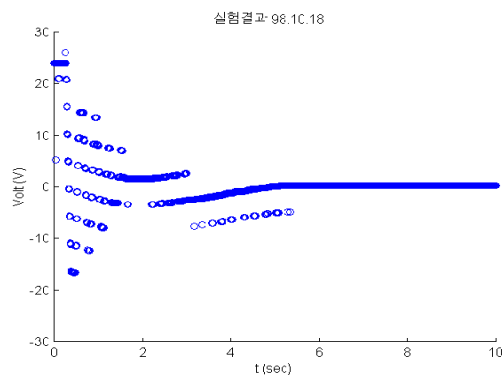
C-프로그램으로 제어와 동시에 저장한 데이터를 MATLAB을 이용해 그린 결과이다. 데이터를 획득한 제어용 C-프로그램은 첨부에 수록되어 있다. 2회 실험한 결과이다.



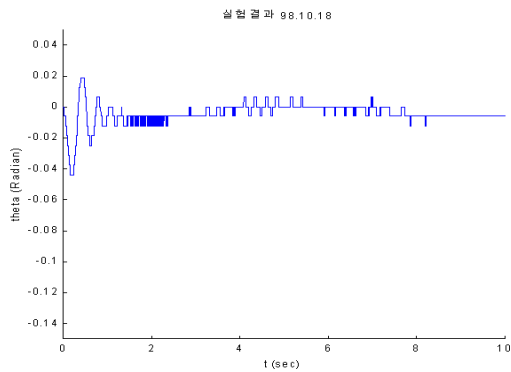
Bar의 변위



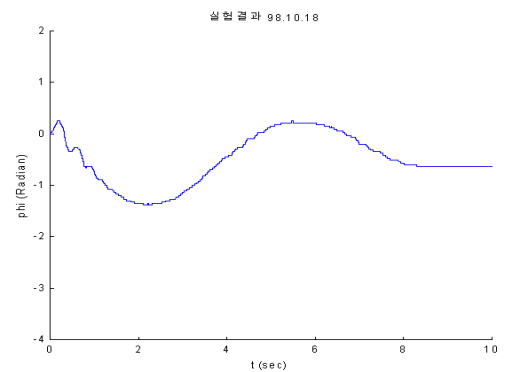
Motor의 변위



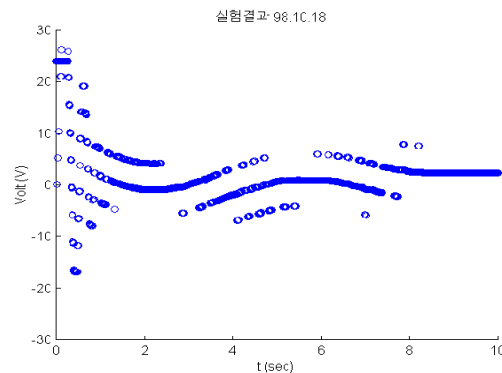
출력전압



Bar의 변위



Motor의 변위



출력전압

실험결과를 대략 분석하자면, 시뮬레이션 결과에서 보다 실제 실험의 완전수렴이 더 빠를 때가 있음을 알 수 있다. 그러나 여러번의 실험을 통해 보았을 때 ‘운이 좋았던’ 경우였음을 알 수 있다. 실험실 조건이나 시스템 자체의 조건이 약간 변화하였을 때 이 최종 수렴시간은 매우 크게 차이가 났다. 그리고, 전반적으로 실제 계에서의 불안정성이 시뮬레이션에서보다 더 큼을 알 수 있었다. 이는 모델링 되지 못했던 여러 요인들에 의한 결과로 생각된다. 그리고 같은 외란에 대해 시뮬레이션이 실제 실험에서 보다 더욱 큰 응답을 보이고 있는데 이 역시 모델링의 부정확성에 기인한다고 보여진다. 또한, 각변위  $\theta$ 의 실제 실험그래프를 보면 큰 오버슈트에 의한 부족감쇠 현상이 보이고 있는데, 이는 각속도 데이터  $\dot{\theta}$ ,  $\dot{\phi}$ 를 엔코더의 변위로 부터 측정된 값을 1차로 차분하여 얻음으로써 시간지연이 발생하였고, 출력전압에 대한 DC-Motor의 엄밀한 선형성이 보장되지 않은 것, 부정확한 파라미터에 의한 것등으로 생각된다. 그러나 이러한 조건들을 충분히 만족시키지 못했음에도 불구하고 처음 목표했던 ‘손으로 툭툭 치는’ 정도의 외란에 대해 상당히 효과적으로 제어하는 성능에 도달하였다. 실험에 사용한 외란은 안정한 상태에서 24V를 0.1s 동안 가한 것인데, 이로 인해  $\theta$ 가  $3^\circ$  정도 움직였고,  $\dot{\theta}$ 도 존재하고 있는 비교적 큰 외란이다. 1m 길이의 Bar 상단의 변위는 약 2.6cm 움직인다는 뜻이다.

사용된 LQR의 특징이라면, LQR이 만들어내는 출력전압의 파형이 2개의 주파수대로 이루어져 있음이 보인다. 하나는  $\theta$  입력에 대한 보상이고, 나머지 하나는  $\phi$  입력에 대한 보상임을 알 수 있다. 그리고 그 진동수의 차이가 곧 각 상태변수의 질량관성모멘트와 관계되는 고유진동수 및 댐핑계수에 연관됨을 추론할 수 있다. 설계된 LQR은 영의 기준입력을 기반으로 오차가 생겼을 때 보상해주는 역할을 충실히 수행하고 있다. 결론적으로, 어떤 제어해야할 시스템의 물성치들을 알고 있고 그 시스템에 관한 운동방정식을 기술할 수 있으며 이것이 선형적이라면(또는 선형화가 가능하다면) 이와같은 LQ 최적제어기법을 적용해도 좋고 결론지을 수 있다. 나아가 입출력 변수의 개수가 어느정도 늘어나더라도 이러한 제어기 설계에 크게 어려움이 없다고 생각된다. 물론, 다변수 시스템의 경우 비선형성이 강해지고 물성치들을 미리 측정하기가 곤란하며 시간에 따라 물성치가 변화하는 시변시스템일 경우 보다 개선된 제어기법의 적용 및 연구가 필요할 것이다.

## 6. 결론

이상 PID와 LQ 최적 제어기를 실제 시스템에 적용하여 본 결과 어느 정도의 외란에도 불구하고 도립된 상태를 유지하는데 만족스러운 결과를 얻었다. 그러나 어떤 경우에는 매우 불안한 때가 있었는데 다음 경우에 더욱 민감한 반응을 보였다.

1) 시스템의 프레임이 좌우로 기울어져 있을 때 도립진자가 기울어진 방향으로 급속히 쏠리는 현상이 일어났다. 이는 모델링된 시스템에 많은 영향을 주는 변수의 하나로 간주할 수 있을 것이다.

2) 사용한 도립진자는 원통형의 속이 빈 알루미늄 프레임을 사용하였는데, 길이가 긴 관계로 제어동작시 종종 공진 현상이 발생하였다. 모델링 과정에 포함되지 않은 Bar의 변형과 탄성력 등이 제어기의 오동작에 영향을 주었을 가능성이 있다.

3) Bar에 연결된 엔코더에서 나온 케이블이 모터의 심한 회전으로 인해 감겨버리는 일이 잦았는데, 감기었을 때 케이블의 장력으로 인해 모터의 토크가 극복하지 못하는 경우 제어에 실패하였다. Rotating Inverted Pendulum은 Boundary의 제약을 극복할 수 있는 구조이나 이러한 케이블 등의 문제로 인해 예기치 못한 Boundary가 발생했다고 본다.

4) 모터에 인가되는 전압은 24V가 한계이고 전류는 1.5A로 제한시켰다. 때문에 모터가 낼 수 있는 최대 토크 역시 제한받는다. 그러나 LQ 제어기는 진자의 각도와 속도에 따라 이론적으로 무한대의 범위를 갖는 제어 입력 값을 발생시킨다. 그러므로 진자가 지나치게 많이 기울어 졌거나 쓰러지는 속도가 너무 빠를 때 이를 보정하는데 실패하고 만다. 이를 해결하기 위한 방법은 모터의 토크를 충분히 인가할 수 있는 방법, 즉 강력한 모터를 사용하는 것이라고 생각한다.

5) 최적 제어기법은 시스템의 파라미터들을 정확하게 파악하는 것이 중요한데, 모터의 토크상수와 댐핑 계수 등을 측정하여 결정함에 있어 부정확한 부분이 많았다. 이는 모터자체가 미리 모델링한 것과 같은 운동을 보여주지 않은 비선형성이 있었고, 계측과정에 있어 정확도에 문제점이 있었기 때문이다 (모터가 최종속도 근처에 이르면 편심 하중으로 인하여 프레임이 심하게 진동하였음). 또한 모터에 부과된 질량들과 그 질량 관성모멘트를 결정함에 있어 생략되거나 측정 오차가 큰 것들이 있었다. 이 역시 부정확한 파라미터를 만드는 요인이 되었다.

6) 더욱 우수한 성능을 가진 시스템을 위해 다음의 개선점을 들 수 있다.

- (1) Bar의 탄성변형, 자체진동에 관한 동특성을 운동방정식에 포함시켜 시스템을 해석하는 것
- (2) 무시된 엔코더의 회전축방향의 질량 관성모멘트와 댐핑 계수를 측정하여 운동방정식에 포함하는 것
- (3) Motor와 구동회로의 출력 범위를 더욱 상승 시킬 것  
(더욱 강한 Motor와 구동회로를 사용할 것)
- (4) 가벼운 부품을 사용하여 Motor에 부과된 부하를 줄일 것
- (5) 엔코더의 분해능을 더욱 높일 것
- (6) 더욱 빠른 처리속도를 가진 컴퓨터와 소프트웨어를 사용할 것
- (7) 조립된 전자회로의 신뢰성을 더욱 높일 것(상용의 DSP 보드 사용 등)
- (8) 비선형 제어기법을 적용하여 볼 것(FUZZY-PD 제어기법등)