

데이터 주도(Data-Driven) 제어 기초:

시스템 식별을 하지 않고 실험 데이터로 제어기 설계하기

최근 공학, 자연과학 등 광범위한 분야에서 데이터로부터 직접 인식, 판단 등의 문제를 해결하는 기법들이 각광받고 있다. 제어공학 분야에서도 강화학습과 같은 데이터 주도 제어 기법이 활발하게 연구되고 있다. 본 고에서는 데이터 주도 제어 기법에 관한 기초 지식을 다룬다. 특히, 실험을 통해 얻은 데이터로부터 시스템을 식별한 후 제어기를 설계하는 전통적인 제어 방식에서 시스템 식별을 배제하고 실험 데이터와 제어기 설계를 직접 연결하였을 때 새로운 설계 인자를 얻을 수 있다는 것을 설명한다. 또한 제어기 설계 관점에서 의미있는 실험 데이터를 얻기 위한 조건에 대해 고찰한다.

백주훈 (광운대학교 로봇학부)

1. 시스템 식별을 통한 제어기 설계

전동기, 로봇, 쿼드콥터 등 우리에게 어떤 동적 시스템이 주어졌고 이에 대한 제어기를 설계해야 한다고 생각하자. 당연히 계도 가장 처음 할 일은 뉴턴 법칙, 공기역학, 키르히호프의 법칙 등 각종 지식을 사용하여 시스템을 표현하는 것이다. 시스템의 거동, 즉 시스템의 운동방정식을 구한 후에는 운동방정식 내의 계수를 알아내는 과정을 거친다. 질량이나 저항 등 특정한 계수를 구하기 해 이를 잘 구할 수 있는 실험을 고안하기도 하고 다양한 시스템 식별 방법을 동원할 수도 있다. 상당한 노력을 거쳐 시스템을 완벽하게 알아낸 후에야 제어 목적에 맞는 전략을 수립하고 설계를 진행하게 된다.

예를 들어, 아래와 같이 표현되는 제어 가능한 진자 시스템을 생각하자.

$$\ddot{y} = -\omega_0^2 y + u. \quad (1)$$

여기서 y 는 각도 변위, ω_0 는 고유 진동수, u 는 제어 토크라고 하자. 상태변수를 $x_c(t) = [y(t); \dot{y}(t)]^T$ 라 정의하면 다음과 같은 상태공간 방정식으로 표현할 수 있다.

$$\dot{x}_c = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} x_c + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u =: A_c x_c + B_c u. \quad (2)$$

이 시스템을 디지털 제어기로 제어하고자 하여 샘플링 시간을 T_s 로 두고, 등가의 이산 시간 시스템을 구하여 다음을 얻는다고 하자.

$$x(k+1) = Ax(k) + Bu(k). \quad (3)$$

여기서 $x(k) = x_c(kT_s)$, $u(k) = u(kT_s)$ 이고, A , B 는 각각 이산 시간 시스템의 시스템 행렬과 입력 행렬이며 연속 시간의 A_c , B_c 와는 다른 값을 갖는다.

앞에서 설명한 바와 같이 제어기 설계를 위해 시스템의 계수를 알아내는 것이 중요한 일이며 이를 위해 연속 시간에서의 실험을 통해 고유진동수 ω_0 를 알아낸 후 계산을 거쳐 A 와 B 를 구하는 것이 하나의 방법이다. 또 다른 방법으로 특정한 이산 입력 신호 $u(k)$ 를 시스템에 인가하여 $x(k)$ 를 구한 후 A 와 B 를 구할 수 있을 것이다. 최근 대부분의 시스템이 디지털화되어 있으므로 후자의 방법이 더 적합한 경우가 많다고 할 수 있다.

이제, 실험을 통해 그림 1과 같은 데이터를 얻었다고 하자. 데이터는 총 $T+1$ 쌍으로 다음과 같이 표현하자.

1) 여기서 MATLAB 기호를 사용하였다. 즉, 두 벡터 x , y 가 있을 때 $[x;y] = [x^T \ y^T]^T$ 로 정의한다.

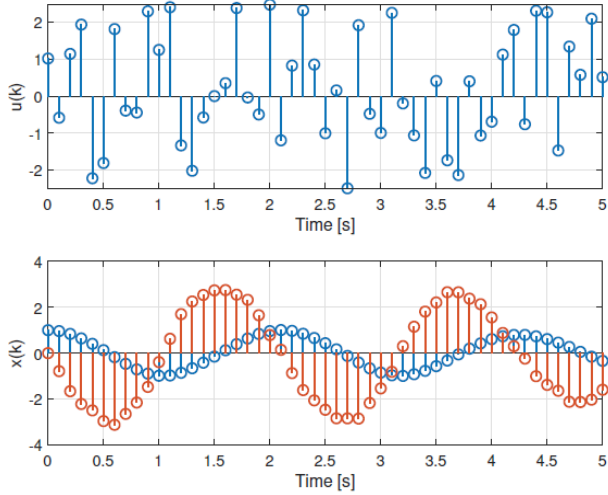


Fig. 1. 랜덤 입력을 인가하여 얻은 상태 벡터 거동. 관련 코드는 부록 참고.

$$\{(u_d(0), x_d(0)), \dots, (u_d(T), x_d(T))\}.$$

여기서 $u(k)$ 대신 $u_d(k)$ 와 같이 아래첨자 d 를 사용했다는 것에 유의한다. 이는 실험을 통해 얻은 데이터를 의미하는 것으로 이후에도 같은 의미로 사용할 것이다. 위 데이터로부터 다음과 같은 행렬을 정의하자.

$$\begin{aligned} U_{0,T} &= [u_d(0) \quad u_d(1) \quad \dots \quad u_d(T-1)] \\ X_{0,T} &= [x_d(0) \quad x_d(1) \quad \dots \quad x_d(T-1)] \\ X_{1,T} &= [x_d(1) \quad u_d(2) \quad \dots \quad x_d(T)]. \end{aligned} \quad (4)$$

$u_d(k), x_d(k), x_d(k+1)$ 은 실제 시스템 (3)의 동역학을 만족해야 하므로 다음을 만족한다.

$$\begin{aligned} x_d(1) &= [B \quad A][u_d(0); x_d(0)] \\ x_d(2) &= [B \quad A][u_d(1); x_d(1)] \\ &\vdots \\ x_d(T) &= [B \quad A][u_d(T-1); x_d(T-1)]. \end{aligned}$$

위 식들은 다음과 같이 쓸 수 있다.

$$X_{1,T} = [B \quad A] \begin{bmatrix} U_{0,T} \\ X_{0,T} \end{bmatrix}. \quad (5)$$

만일 행렬 $[U_{0,T}; X_{0,T}]$ 의 모든 행벡터가 선형 독립이라면 행렬 A, B 는 다음과 같이 구할 수 있다.

$$[B \quad A] = X_{1,T} \begin{bmatrix} U_{0,T} \\ X_{0,T} \end{bmatrix}^\dagger. \quad (6)$$

여기서 $[U_{0,T}; X_{0,T}]^\dagger$ 는 $[U_{0,T}; X_{0,T}]$ 의 의사역행렬이다. 즉, 행렬 $[U_{0,T}; X_{0,T}]$ 의 모든 행벡터가 선형 독립이라면 시스템의 행렬들을 실험 데이터로부터 얻을 수 있다. 이 사실은 이미 알려져 있으며 시스템 식별(system identification) 기법이라 할 수 있다. 이와 같이 구한 시스템 모델을 바탕으로 비로소 제어를 설계할 수 있게 된다.

2. 데이터 주도 제어기 설계

시스템 식별을 통한 제어기 설계는 실험 데이터로부터 시스템을 표현하는 행렬들을 구하고 이것을 바탕으로 제어기를 설계한다. 최근 시스템 식별 과정을 거치지 않고 실험 데이터로부터 직접 제어기를 설계하는 기법이 활발하게 연구되고 있다 [1, 2]. 이는 공학 분야 전반에서 모델을 사용하지 않고 실험을 통해 얻어진 데이터를 직접 사용하여 문제를 해결하는 흐름과 그 궤를 같이한다.

본 고에서는 참고문헌 [1]의 결과를 바탕으로 선형 시스템에 대한 데이터 주도 제어기 설계의 의미 및 그 장점을 살펴보려 한다. 주요 결과 및 증명, 그리고 그 해석도 모두 이 참고문헌에서 언급한 것임을 밝혀둔다.

다음과 같은 시스템을 고려하자.

$$x(k+1) = Ax(k) + Bu(k), \quad x(k) \in \mathbb{R}^n, \quad u(k) \in \mathbb{R}^m. \quad (7)$$

정리 1 [1]: 실험을 통해 얻은 데이터 $(u_d(0), x_d(0)), \dots, (u_d(T), x_d(T))$ 가 다음 조건을 만족한다고 가정하자.

$$\text{rank} \begin{bmatrix} U_{0,T} \\ X_{0,T} \end{bmatrix} = n + m. \quad (8)$$

시스템 (7)과 제어 입력 $u = Kx$ 로 이루어진 폐루프 시스템은 다음과 같이 쓸 수 있다.

$$x(k+1) = X_{1,T} L_K x(k). \quad (9)$$

여기서 $L_K \in \mathbb{R}^{T \times n}$ 는 다음을 만족한다.

$$\begin{bmatrix} K \\ I_n \end{bmatrix} = \begin{bmatrix} U_{0,T} \\ X_{0,T} \end{bmatrix} L_K. \quad (10)$$

또한 제어 입력은 다음과 같다.

$$u(k) = U_{0,T} L_K x(k). \quad (11)$$

정리 1의 증명은 매우 간단하다. 제어 입력 $u(k) = Kx(k)$ 를 생각하자. 식 (10)은 선형 방정식이라는 것에 주목하자. 가정 (8)에 의해 주어진 임의의 행렬 K 에 대해 식 (10)을 만족하는 L_K 가 반드시 존재한다. 이 식에서 $K = U_{0,T}L_K$ 이므로 제어 입력은 (11)로 주어진다. 마지막으로 페루프 시스템은 다음과 같이 쓸 수 있다.

$$x(k+1) = [B \quad A] \begin{bmatrix} u(k) \\ x(k) \end{bmatrix} = [B \quad A] \begin{bmatrix} K \\ I_n \end{bmatrix} x(k).$$

마지막으로 식 (10)을 대입하면 식 (9)를 얻는다.

$$x(k+1) = [B \quad A] \begin{bmatrix} U_{0,T} \\ X_{0,T} \end{bmatrix} L_K x(k) = X_{1,T} L_K x(k).$$

정리 1로부터 제어 이득 행렬 K 를 $U_{0,T}L_K$ 와 같이 L_K 로 바꾸어 쓸 수 있다는 것을 확인할 수 있다. 여기서 더 나아가 식 (9)로부터 페루프 시스템의 시스템 행렬 $X_{1,T}L_K$ 역시 L_K 로 표현된다. 따라서 이 방법은 제어기 설계 조건을 실험데이터 $X_{0,T}$, $U_{0,T}$, $X_{1,T}$ 와 이득 L_K 로 다시 쓸 수 있어 시스템 식별 과정을 통해 제어기를 설계하는 것과는 전혀 다른 설계를 갖는 문제가 된다.

실험 데이터를 통해 시스템 행렬을 식 (6)과 같이 구하고 이를 바탕으로 제어기를 설계하는 방법을 간접적인 데이터 주도 제어기 설계 방법이라 한다면, 이 방법은 제어 이득 및 페루프 시스템의 시스템 행렬을 L_K 로 표현하는 직접적인 데이터 주도 설계 방법이라 부를 수 있다. 결정해야 하는 변수의 개수를 살펴보면 전자의 설계 인자 K 는 $m \times n$ 행렬로 시스템의 구조에 의해서 그 크기가 정해지고 후자의 설계 인자 L_K 는 $T \times n$ 행렬로 실험 데이터의 크기에 의존한다.

또 하나 언급할 만한 점은 주어진 K 가 시스템을 안정화하는 제어기 이득인지를 쉽게 확인할 수 있다는 것이다. 이는 페루프 시스템의 시스템 행렬이 바로 $X_{1,T}L_K$ 로 주어지므로 A , B 를 구하여 $A+BK$ 를 계산할 필요가 없기 때문이다.

이제, 실험 데이터를 사용하여 안정화 제어기를 설계해보도록 하자. 만일 A , B 를 알고 있다면 $A+BK$ 가 슈어(Schur) 안정하도록 이득 행렬 K 를 구하는 것이 필요하다. 이를 위한 한 가지 방법으로 이 문제를 양한정 행렬 R 에 대해 리아프노프 방정식 $(A+BK)^T P(A+BK) - P + R = 0$ 을 만족하는 양한정 행렬 P 를 구하는 문제로 바꾸는 방법이 있다. 이 경우 선형 행렬부등식을 얻게 되며 수치적으로 그 해를 구할 수 있다. 데이터 주도 제어기 설계방법의 경우 페루프 시스템의 시스템 행렬

$X_{1,T}L_K$ 가 L_K 에 선형이므로 유사한 결과를 기대할 수 있을 것이다.

정리 2 [1]: 실험을 통해 얻은 데이터 $(u_d(0), x_d(0))$, ..., $(u_d(T), x_d(T))$ 가 (8)을 만족한다고 가정하자. 다음을 만족하는 임의의 행렬 Q 를 생각하자.

$$\begin{bmatrix} X_{0,T}Q & X_{1,T}Q \\ Q^T X_{1,T}^T & X_{0,T}Q \end{bmatrix} > 0. \quad (12)$$

이 행렬 Q 로부터 다음 제어 이득 행렬을 정의하자.

$$K = U_{0,T}Q(X_{0,T}Q)^{-1}. \quad (13)$$

그러면, K 는 시스템 (7)을 안정화한다. 반대로, 만일 K 가 시스템 (7)을 안정화하는 제어 이득 행렬이라면 이는 (13)의 형태로 쓸 수 있으며 Q 는 (12)의 해이다.

위 정리의 증명을 생각해보자. 식 (9)를 상기하면 어떤 K 에 대해 $u = Kx$ 가 적용된 페루프 시스템이 안정하다는 것과 어떤 $P > 0$ 가 다음 조건을 만족하는 것은 서로 동가이라는 것을 알 수 있다.

$$P - X_{1,T}L_K P L_K^T X_{1,T}^T > 0.$$

여기서 L_K 는 (10)을 만족한다. 이제 $Q = L_K P$ 라 하자. 그러면 시스템의 안정성과 다음 식을 만족하는 두 행렬 Q 와 $P > 0$ 의 존재성이 동가이다.

$$\begin{aligned} P - X_{1,T}Q P^{-1} Q^T X_{1,T}^T &> 0 \\ X_{0,T}Q &= P \\ U_{0,T}Q &= KP. \end{aligned}$$

여기서 $P = X_{0,T}Q$ 를 행렬 부등식에 대입하고 슈어 여수정리 (Schur Complement)를 적용하면 부등식 (12)를 얻을 수 있다. 완벽한 증명은 [1]을 참고한다.

예제 1: 정리 2를 이용하여 제어기를 설계해보자. 이를 위하여 제어 가능한 진자 시스템 (2)를 생각하고 이를 이산화한 후 그림 1과 같은 데이터를 얻어 보자. 이 데이터를 얻기 위한 코드는 부록을 참고한다. 제어기 설계를 위한 식 (12)는 Q 에 대한 선형 행렬부등식이며 그 해를 구해주는 다양한 소프트웨어가 있다. 여기서는 CVX [3]를 사용하겠다. CVX에서는 $P > 0$ 과 같은 제약조건을 사용할 수 없으며 이를 적절한 양수 ϵ 을 사용하여 $P \geq \epsilon I$ 와 같이 바꾼다. 부록의 2번 항목을 참고한다. 그림 2는 위에서 얻은 상태 궤환 제어기를 사용했을 때의 시스

템 응답이다. 두 상태 모두 영으로 수렴함을 알 수 있다. ◇

여기서 잠깐 모델을 아는 경우 상태 궤환 제어를 설계하는 방법을 회상해보자. 만일 (A, B) 가 제어가능하다면 페루프 시스템 행렬 $A+BK$ 가 설계자가 원하는 고유치를 갖도록 하는 행렬 K 를 찾을 수 있다. 데이터 주도설계 방법에서도 이와 같은 얘기를 할 수 있을까?

정리 1을 돌이켜보면 페루프 시스템이 원하는 고유치를 갖도록 하는 문제는 $X_{0,T}L_K = I_n$ 을 만족하면서 $X_{1,T}L_K$ 가 원하는 고유치를 갖도록 하는 L_K 를 구하는 문제로 환원된다는 것을 알 수 있다. 첫 번째 식으로부터 L_K 를 다음과 같이 얻을 수 있다.

$$L_K = X_{0,T}^\dagger + (I_T - X_{0,T}^\dagger X_{0,T})w$$

여기서 $w \in \mathbb{R}^T$ 는 임의의 벡터이다. 이로부터 페루프 시스템의 시스템 행렬을 다음과 같이 쓸 수 있다.

$$\begin{aligned} X_{1,T}L_K &= X_{1,T}X_{0,T}^\dagger + X_{1,T}(I_T - X_{0,T}^\dagger X_{0,T})w \\ &=: \bar{A} + \bar{B}w. \end{aligned}$$

만일 (\bar{A}, \bar{B}) 가 제어 가능하다면 페루프 시스템의 시스템 행렬 $X_{1,T}L_K$ 의 고유치를 원하는 곳에 둘 수 있게 된다. 설계 인자 측면에서 모델 기반 문제와 데이터 주도 문제의 차이를 생각해보면, 전자는 설계 인자 K 가 $m \times n$ 행렬인 반면, 후자의 설계 인자 w 는 $T \times n$ 행렬이다.

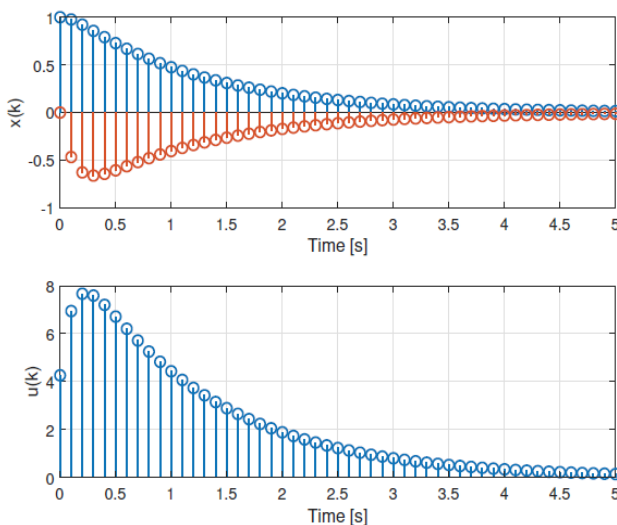


Fig. 2. 데이터 주도 상태 궤환 제어를 사용한 페루프 시스템 응답. $x(0) = [1; 0]$.

3. 지속 여기 조건

앞 절에서 데이터 주도 제어기 설계 기법에 대해서 알아보았다. 정리 1과 2 모두 조건 (8)을 가정하고 있다. 언제 이 조건이 만족되는 것일까? 참고문헌 [4]에 의하면 실험 데이터를 얻을 때 사용한 입력 $u_d(0), \dots, u_d(T-1)$ 이 지속 여기 조건 (Persistently Exciting Condition)을 만족하면 (8)이 자동으로 만족된다. 물론 이는 (8)을 보장하는 하나의 충분조건이므로 보다 완화된 조건이 있을 수 있겠다. 최근 이에 대한 연구도 활발히 진행되고 있다.

이 절에서는 입력 신호의 지속 여기 조건에 대해 알아보고 이것이 어떻게 (8)을 보장하는지 간단하게 설명하도록 하겠다. 정수 t 에 대하여 l 차원의 벡터를 갖는 신호 $f(t)$ 에 대하여 다음을 정의하자.

$$f_{[0,T]} = [f(0) \quad f(1) \quad \dots \quad f(T)].$$

또한 이로부터 $(lL) \times (T-L)$ 크기의 한켈(Hankel) 행렬 $F_{0,L,T-L+1}$ 을 다음과 같이 정의하자.

$$F_{0,L,T-L+1} = \begin{bmatrix} f(0) & f(1) & \dots & f(T-L) \\ f(1) & f(2) & \dots & f(T-L+1) \\ \vdots & \vdots & \ddots & \vdots \\ f(L-1) & f(L) & \dots & f(T-1) \end{bmatrix}.$$

만일 $L=1$ 이면 $F_{0,1,T}$ 를 $F_{0,T}$ 라 하겠다.

정의 1: $f_{[0,T]}$ 의 한켈 행렬 $F_{0,L,T-L+1}$ 을 생각하자. 만일 $F_{0,L,T-L+1}$ 의 모든 행벡터가 선형 독립이라면 $f_{[0,T]}$ 는 차수 L 의 지속 여기 신호라고 정의한다.

$F_{0,L,T-L+1}$ 의 모든 행벡터가 선형 독립이기 위해서는 적어도 이 행렬의 열의 수가 행의 수보다 크거나 같아야 한다. 즉 $T \geq (l+1)L-1$ 이어야 한다.

보조정리 1: 다음과 같이 표현되는 제어가능한(controllable) 시스템을 생각하자.

$$x(k+1) = Ax(k) + Bu(k), \quad x(k) \in \mathbb{R}^n, \quad u(k) \in \mathbb{R}^m.$$

만일 입력 $u_{d,[0,T-1]}$ 이 차수 $n+t$ 의 지속 여기 신호라면 다음이 성립한다.

$$\text{rank} \begin{bmatrix} U_{0,T-t+1} \\ X_{0,T-t+1} \end{bmatrix} = n + tm.$$

일반적인 경우에 대한 증명 [4]은 상당히 복잡하고 난해하므로 입력이 하나이고 상태 변수가 2개인 시스템 즉, $n=2$,

$m = 1$ 인 경우를 생각하자. 추가적으로 정리 1과 2에서 필요한 조건은 $t = 1$ 인 경우이므로 이를 가정하자. $u_{d,[0, T-1]}$ 이 차수 $3(=n+t)$ 의 지속 여기 신호라는 것은 정의 1에 의해 $U_{0,3, T-2}$ 의 모든 행벡터가 선형 독립이라는 것을 의미한다. 이 조건을 만족하기 위한 최소한의 T 는 5이다. ($T \geq 2(n+t) - 1$) 이제 $T = 5$ 라 두자. 사실 $T = 5$ 인 경우에 보조정리 1이 성립한다면 $T \geq 5$ 인 경우에도 당연히 보조정리가 성립하므로 $T = 5$ 인 경우에 대해서만 증명해도 충분하다.

이후에는 $T = 5, n = 2, m = 1, t = 1$ 이라 하자. 다음 행렬을 생각하자.

$$U_{0,3,3} = \begin{bmatrix} u_d(0) & u_d(1) & u_d(2) \\ u_d(1) & u_d(2) & u_d(3) \\ u_d(2) & u_d(3) & u_d(4) \end{bmatrix}$$

$$U_{0,5} = [u_d(0) \quad u_d(1) \quad u_d(2) \quad u_d(3) \quad u_d(4)]$$

$$X_{0,5} = [x_d(0) \quad x_d(1) \quad x_d(2) \quad x_d(3) \quad x_d(4)].$$

이제 보여야 할 것은 $U_{0,3,3}$ 의 차수가 3이면 행렬 $[U_{0,5}; X_{0,5}]$ 의 차수가 3이라는 것이다. 여기서 잠깐 $u_{[0,4]}$ 가 차수 3의 지속 여기 신호라는 것의 의미를 살펴보자. 이것은 $U_{0,3,3}$ 의 세 행벡터가 선형 독립이라는 것이므로 총 5초 동안의 입력 신호 $u_{d,[0,4]}$ 의 일부분으로 이루어진 3초 동안의 입력 신호들 $u_{d,[0,2]}, u_{d,[1,3]}, u_{d,[2,4]}$ 가 서로 독립이라는 것을 의미한다. ($u_{d,[k_0, k_1]} = [u_d(k_0) \dots u_d(k_1)]$ 이라는 기호를 사용하였다.)

행렬 $[U_{0,5}; X_{0,5}]$ 의 차수가 3이라는 것을 보이기 위해 이 행렬의 행벡터들의 선형 조합으로 영벡터를 만드는 경우를 생각해 보자. 즉, 어떤 행벡터 $[w_1 \quad w_x^\top]$, $w_1 \in \mathbb{R}, w_x \in \mathbb{R}^2$ 가 다음을 만족한다고 하자.

$$w_1 U_{0,5} + w_x^\top X_{0,5} = 0.$$

위 식으로부터 등가의 다음 식을 얻을 수 있다.

$$\begin{aligned} w_1 [u_d(0) \quad u_d(1) \quad u_d(2)] + w_x^\top [x_d(0) \quad x_d(1) \quad x_d(2)] &= 0 \\ w_1 [u_d(1) \quad u_d(2) \quad u_d(3)] + w_x^\top [x_d(1) \quad x_d(2) \quad x_d(3)] &= 0 \\ w_1 [u_d(2) \quad u_d(3) \quad u_d(4)] + w_x^\top [x_d(2) \quad x_d(3) \quad x_d(4)] &= 0. \end{aligned} \tag{14}$$

관계식 $x_d(1) = Ax_d(0) + Bu_d(0), x_d(2) = A^2x_d(0) + ABu_d(0) + Bu_d(1)$ 을 위 식의 첫 번째 식에 적용하면 다음을 얻을 수 있다.

$$\begin{aligned} w_1 u_d(0) + w_x^\top x_d(0) &= 0 \\ w_1 u_d(1) + w_x^\top (Ax_d(0) + Bu_d(0)) &= 0 \\ w_1 u_d(2) + w_x^\top (A^2x_d(0) + ABu_d(0) + Bu_d(1)) &= 0 \end{aligned}$$

케일라-해밀턴(Cayley-Hamilton) 정리에 의해 $A^2 = -\alpha_0 I - \alpha_1 A$ 를 만족하는 α_0, α_1 이 존재하며 이를 이용하여 위 식에서 $x_d(0)$ 를 소거할 수 있다. 즉, 다음을 얻는다.

$$\begin{aligned} (w_1 \alpha_0 + w_x^\top (AB + \alpha_1 B)) u_d(0) \\ + (w_1 \alpha_1 + w_x^\top B) u_d(1) + w_1 u_d(2) &= 0. \end{aligned}$$

식 (14)의 두 번째 식과 세 번째 식으로부터 위와 유사한 식을 얻을 수 있으며 이를 아래와 같이 모아 쓸 수 있다.

$$\begin{aligned} [w_1 \alpha_0 + w_x^\top (AB + \alpha_1 B) \quad w_1 \alpha_1 + w_x^\top B \quad w_1] \\ \times \begin{bmatrix} u_d(0) & u_d(1) & u_d(2) \\ u_d(1) & u_d(2) & u_d(3) \\ u_d(2) & u_d(3) & u_d(4) \end{bmatrix} &= 0. \end{aligned}$$

여기서 $u_d(0), \dots, u_d(4)$ 로 이루어진 행렬이 $U_{0,3,3}$ 임에 유의하고 $U_{0,3,3}$ 의 차수가 3임을 이용하면 다음을 얻는다.

$$[w_1 \alpha_0 + w_x^\top (AB + \alpha_1 B) \quad w_1 \alpha_1 + w_x^\top B \quad w_1] = 0.$$

$w_1 = 0$ 이라는 점과 제어가능성($[B \quad AB]$ 의 차수가 2)을 이용하면 $w_x = 0$ 임을 얻을 수 있다. 이로부터 행렬 $[U_{0,5}; X_{0,5}]$ 의 차수가 3이라는 결론을 얻는다.

결국, 지속 여기 조건을 만족하는 충분히 긴 입력 신호를 제어 가능한 시스템에 인가하여 얻은 데이터는 식 (8)을 만족하게 된다.

4. 결론

본 고에서는 실험을 통해 얻은 입력 및 상태 데이터로부터 전통적인 안정화 제어를 직접적으로 얻는 방법에 대해 알아 보았다. 이 과정에서 시스템 행렬을 사용하였을 때와는 매우 다른 설계 인자가 등장하고 이를 사용하여 극점 배치나 선형 행렬부동식을 사용한 제어기 설계기법을 다시 쓸 수 있다는 것을 설명하였다. 이 기법은 대표적인 데이터 주도 기법인 강화학습에 기초하는 제어 기법들에 비해 모델 기반의 제어기 설계 기법 자체를 데이터로 다시 쓰는 기법이라고 할 수 있겠다. 즉, 이미 알려진 검증된 모델 기반 제어기 설계 기법이 있다면 시스템 식별 과정을 거치지 않고 실험 데이터로 직접 표현되는 유사한 제어를 얻을 수 있을 것이다. 실제로 이러한

시도가 활발하게 진행되고 있으며 대표적으로 모델 예측 제어 기법의 아이디어를 데이터 주도 제어로 전환한 결과가 있다. 관심 있는 독자는 [5]를 참고하기 바란다.

부록

1. 그림 1을 얻기 위한 시뮬레이션 코드.

```
% continuous time system
omega_0=3;
Ac= [0 1; -omega_0^2 0];
Bc= [0;1];
Cc=eye(size(Ac)); % output: all states
Dc= zeros(size(Cc,1), size(Bc,2));
Sysc=ss(Ac,Bc,Cc,Dc);

% discrete time system
Ts=0.1;
Sysd= c2d(Sysc,Ts);
[Ad, Bd, Cd, Dd]=ssdata(Sysd);
Tfinal=5; %Final time (continuous time)
Ud_rand=rand(1,Tfinal/Ts+1)*5-2.5;

% simulate with initial condition x0 and input Ud
Tspan=0:Ts:Tfinal;
Ud=Ud_rand;
x0=[1;0];
[YY, TT, XX] =lsim(Sysd, Ud, Tspan, x0);

X0=XX(1:(size(XX,1)-1),:);
X1=XX(2:size(XX,1),:);
U0=Ud(:, 1:size(X0,2));
```

2. 정리 2의 식 (12)를 구현한 코드.

```
cvx_begin sdp
variable Q(size(X0,2), size(X0,1));
[X0*Q, X1*Q; Q'*X1' X0*Q]>=0.01*eye(2*n);
X0*Q == (X0*Q)';
cvx_end
```

참고: 네 번째 행은 생략 가능하다.

REFERENCES

- [1] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909-924, 2020.
- [2] H. J. van Waarde, J. Eising, H. L. Trentelman, and M. K. Camlibel, "Data informativity: A new perspective on data-driven analysis and control," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4753-4768, 2020.
- [3] Inc. CVX Research. CVX: Matlab software for disciplined convex programming, version 2.0. <http://cvxr.com/cvx>, August 2012.
- [4] J. C. Willems, P. Rapisarda, I. Markovskiy, and B. L. M. de Moor, "A note on per- sistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325-329, 2005.
- [5] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," *Proc. of 18th European Control Conference (ECC)*, pp. 307-312, 2019.

저 자 약 력



백주훈

2004년 서울대학교 전기공학부에서 박사학위를 받고 영국 런던임페리얼 칼리지에서 박사후 연구원으로 재직하였으며 2008년부터 현재까지 광운대학교 로봇학부에 재직 중이다. 선형 및 비선형 시스템을 위한 강인 제어 기법을 주로 연구하고 있으며 다개체 시스템, 로봇, 자동차 등에 관한 응용 연구를 수행하고 있다.