

# Lab 11

*Discrete-Time Estimator-Based Controller*



# Exercise

- 주어진 파일들은 샘플링 주파수가 200Hz의 경우에 대한 discrete-time estimator-based controller의 시뮬레이션과 제어기 구현 코드이다. 이 파일들을 이용하여, 아래의 샘플링 주파수에 대한 시뮬레이션과 실험을 진행하여 결과를 제출한다.
- 100Hz, 50Hz, 20Hz
- State feedback과 estimator의 pole 위치는 변경하지 않고 주어진 값 그대로 사용한다.
- 제어기 프로그램은 state feedback 프로그램과 동일하게 세팅한 후, main.c를 복사하거나 편집하면 됨.

# 샘플링 주파수 변경(1)

- MATLAB simulation code: discrete\_state\_estimator.m  
 $T_s = 1/(\text{sampling frequency})$

```
discrete_state_estimator.m
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Discrete-time state estimator based controller
3  % p : desired pole location
4  % pe: estimator pole location
5  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6
7  clear
8  Ts=1/200; % sampling period
9  Gc=ss([0 -10;0 0],[0;-10],[1 0],0)
10 Gd=c2d(Gc,Ts,'zoh')
11 [a,b,c,d,Ts]=ssdata(Gd)
12 pc1= -10+20*i; pc2=-10-20*i;
13 p=[exp(Ts*pc1) exp(Ts*pc2)];
14 pe=[exp(Ts*pc1*5) exp(Ts*pc2*5)];
15 k=acker(a,b,p)
```

# 샘플링 주파수 변경(2)

## ■ Cortex-M code: main.c

```
int main(void)
{
    /* USER CODE BEGIN 1 */
    a11=1.0;a12=-0.05;a21=0.0;a22=1.0;
    b1=0.0013;b2=-0.05;
    K1=4.7532;K2=-2.0221;
    L1=0.6331;L2=-4.7921;
    sampling_frequency=200;
    /* USER CODE END 1 */
```

a =

1.0000	-0.0500
0	1.0000

b =

0.0013
-0.0500

k =

4.7532	-2.0221
--------	---------

l =

0.6331	-4.7921
--------	---------

# 샘플링 주파수 변경(3)

- Cortex-M code: main.c

$\text{htim10.Init.Period} = 200 \cdot N - 1$

$N = 1000 / (\text{desired frequency})$

```
static void MX_TIM10_Init(void)
{
    /* USER CODE BEGIN TIM10_Init 0 */

    /* USER CODE END TIM10_Init 0 */

    /* USER CODE BEGIN TIM10_Init 1 */

    /* USER CODE END TIM10_Init 1 */
    htim10.Instance = TIM10;
    htim10.Init.Prescaler = 839;
    htim10.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim10.Init.Period = 200*5-1;
}
```

# 샘플링 주파수 변경(4)

- MATLAB plotting code: state\_plot.m  
sf: sampling frequency

```
state_plot.m  x  +
1  clear
2
3  sf=200;
4  load data
5  for i=1:4*sf
6      t(i)=data(i,1)/sf;
7      x1(i)=data(i,2)*10/2048;
8      x2(i)=data(i,3)*10/2048;
9      u(i)=data(i,4)*10/2048;
10 end
11
12 figure(2)
13 subplot(2,1,1);
14 plot(t,x1,'-',t,x2,':')
15 axis([0 4 -0.5 1.5])
```