Advanced Dynamical Systems Control
動的システム制御論
#1

ver. 0.1

Goal of this course
この授業の目標

Rough understanding sample-data control

Undergraduate course

- Classical control theory
  (written in transfer function)
- Modern control theory
  (written in state space)

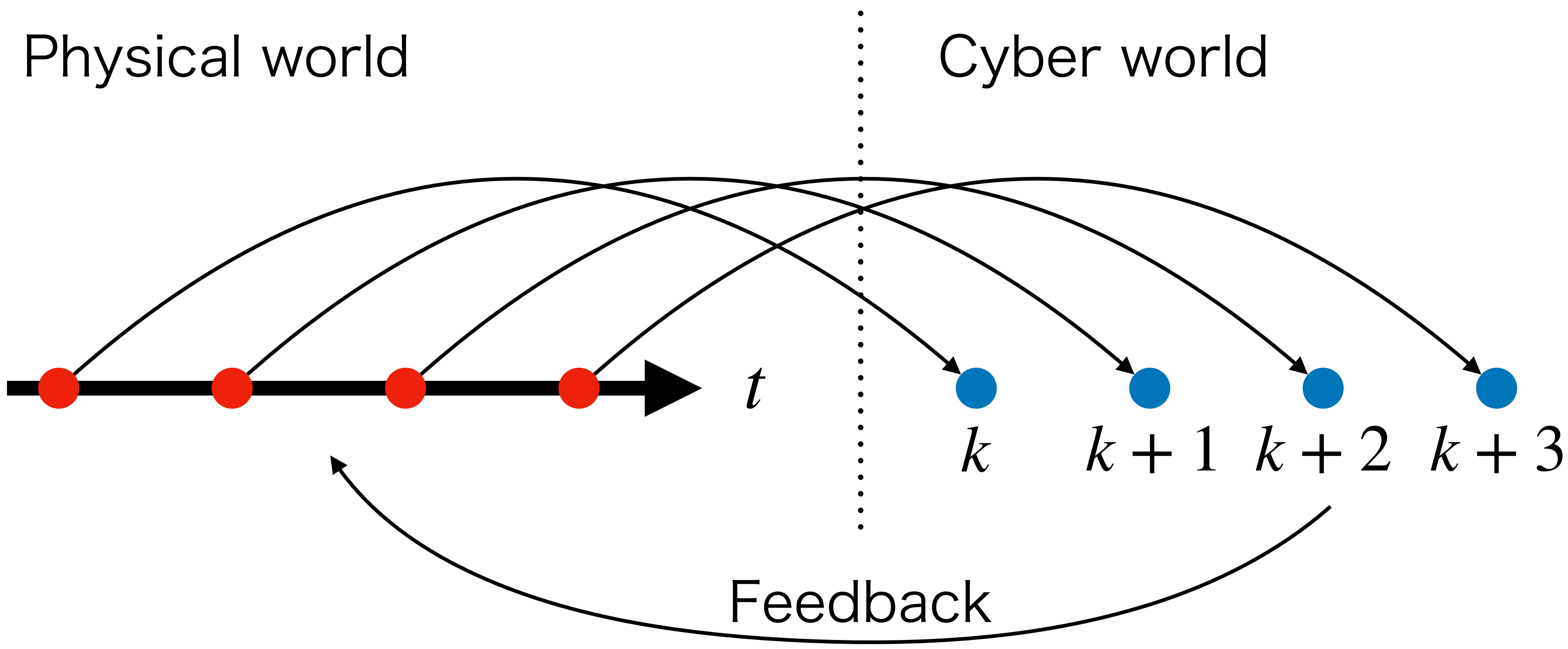
Supposed to be continuous-time system
(based on differential equation)

Considering digital implementation (by computers),
we cannot strictly apply these theories.
because digital computers cannot calculate exact derivatives.

Sample-data control

サンプル値制御とは

Details are the next time



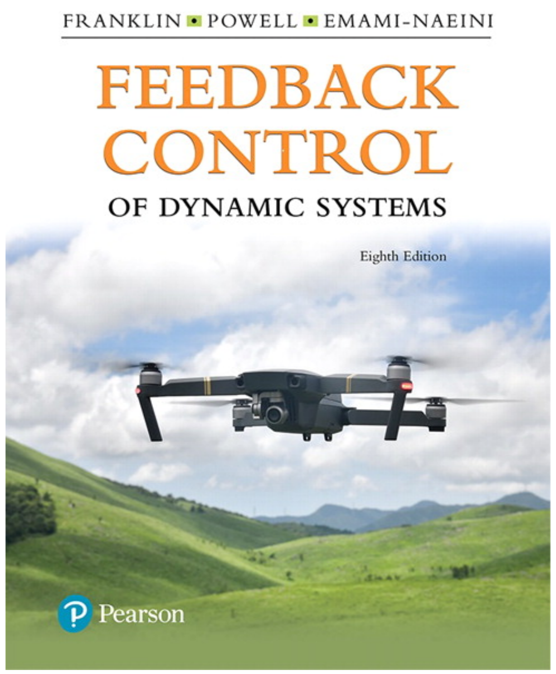
To sample data of physical phenomena and control it by calculating converted discrete data.

Notice about syllabus

シラバスに関する注意

Reference listed in the syllabus

G. F. Franklin, J. D. Powell, and A. F. Emami-Naeini,
Feedback Control of Dynamic Systems, 8th Edition,
Pearson, 2019



This is one of the basic textbook of feedback control
NOT for sample-data control

No textbook has been organized on sample-data control, to my knowledge.

Schedule
授業の予定

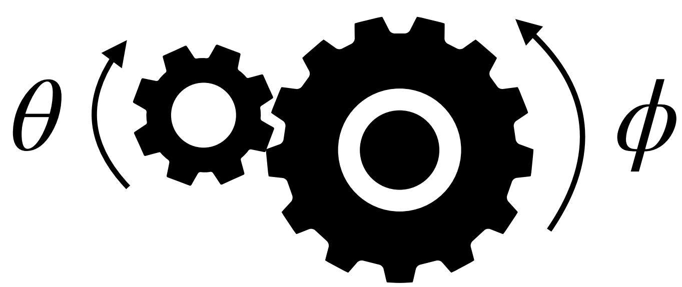
- Brief review of “modern control theory”
  現代制御総復習
- Basic idea of sample-data control
  サンプル値制御理論の考え方
- Continuous-time systems and discrete-time systems
  連続時間システムと離散時間システム
- Stability of discrete-time linear systems
  離散時間線形システムの安定性
- Multi-rate sampling systems
  マルチレートサンプリング系
- Design example of sample-data control systems
  サンプル値制御系の設計例
- Quantization errors and their solution
  量子化誤差とその対策
- Implementation of sample-data systems
  サンプル値制御系の実装

Some other contents may be added or omitted depending on the situation.
状況により追加・省略を行なうことがある.

Brief review of modern control theory

What's "dynamical systems"?

Static system



$$\phi = K\theta$$

Gear ratio

The output is determined
by the input at just that moment

Dynamical system

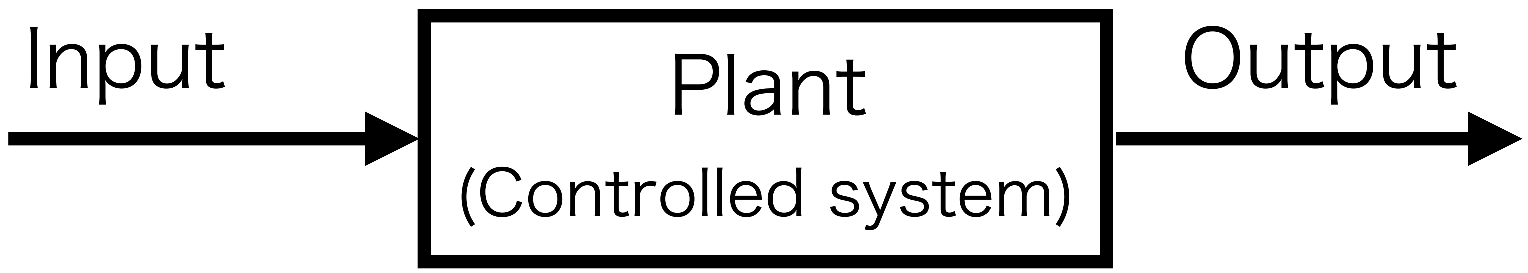
Cars cannot suddenly stop

$$m\ddot{x} + d\dot{x} + kx = F$$

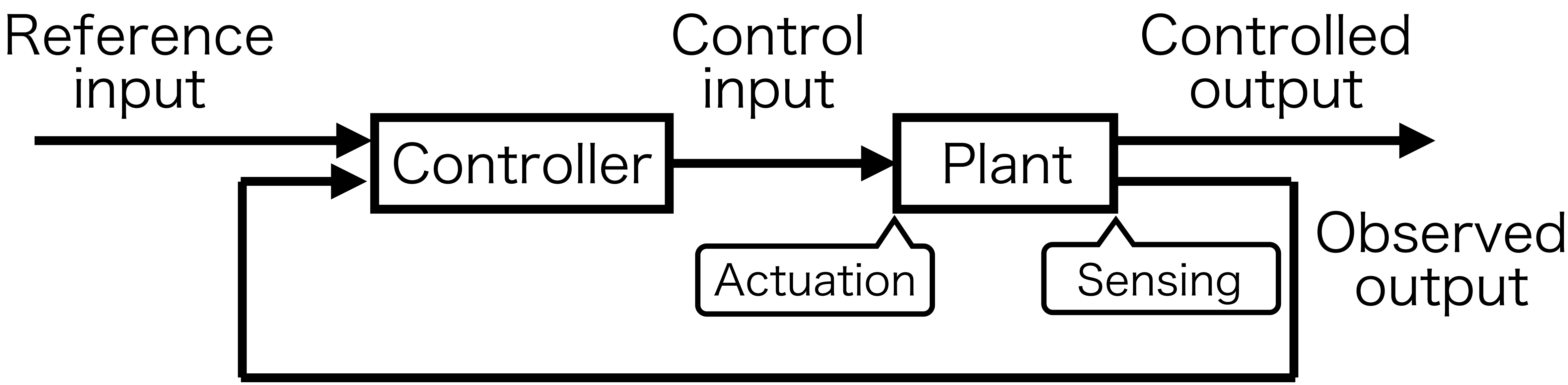
The output is determined
by the past input series

Usually written in differential equations

Block diagram

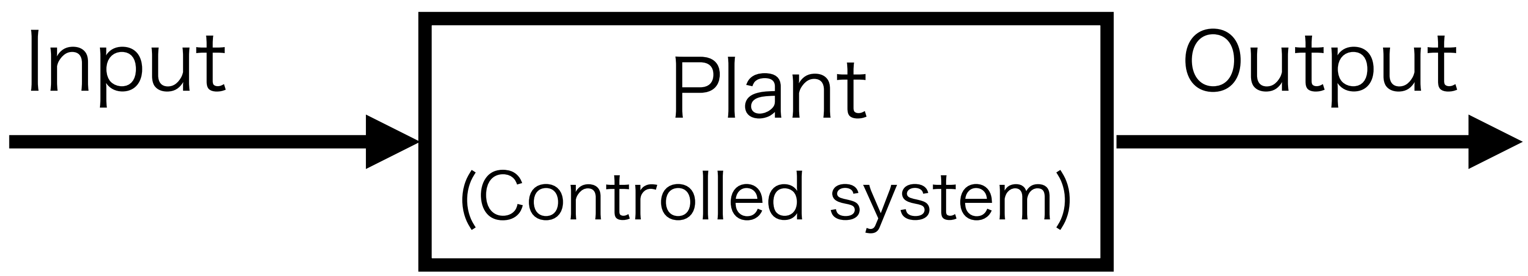


Basic feedback control



State space

one of the system expression



$$\dot{x}(t) = Ax(t) + B\overbrace{u(t)}^{\text{Input}}$$

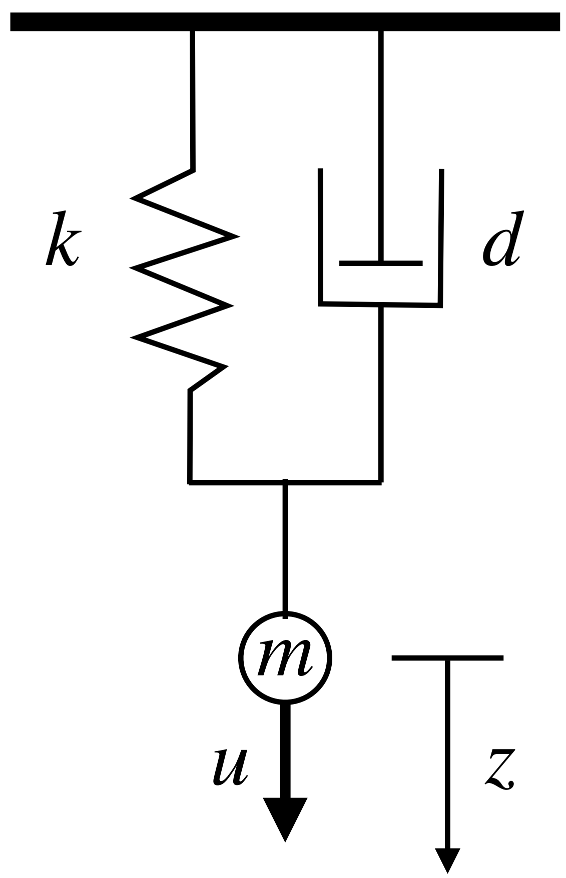
$$\underbrace{y(t)}_{\text{Output}} = C\underbrace{x(t)}_{\text{State}} + Du(t)$$

$x(t),\ y(t),\ u(t)$: Vectors

$A,\ B,\ C,\ D$: Matrices

Example

Spring-damper-mass system



$$m\ddot{z}(t) + d\dot{z}(t) + kz(t) = u(t)$$

Rewrite the above with state space

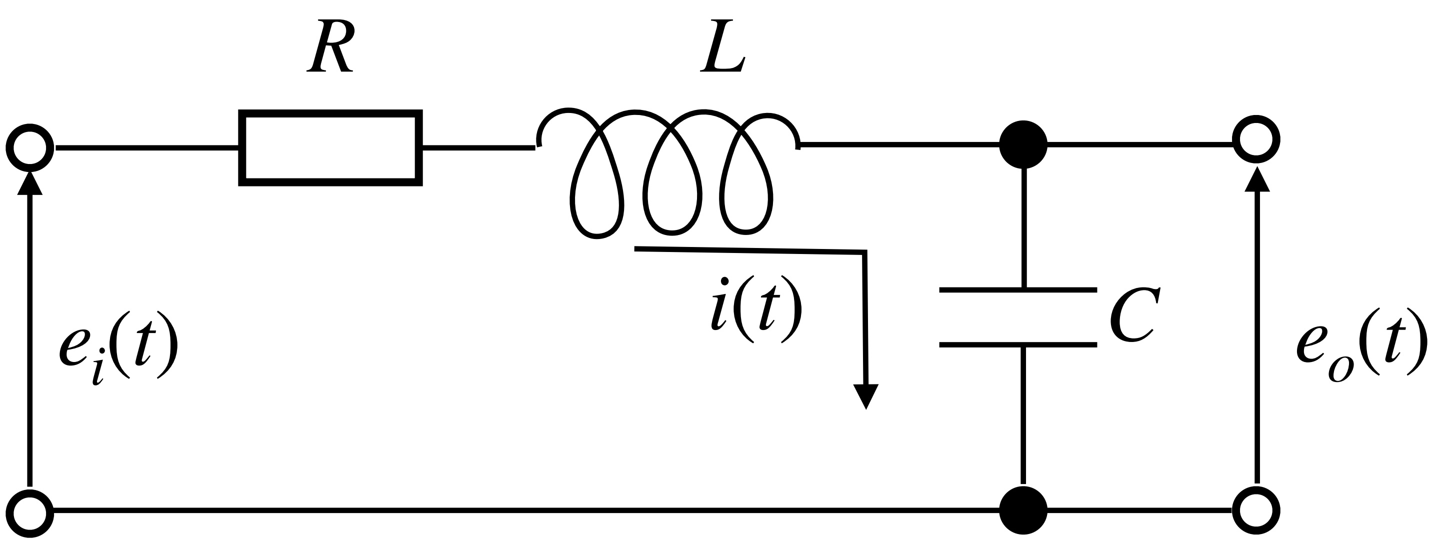
$$\begin{cases} \begin{bmatrix} \ddot{z}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} -\frac{d}{m} & -\frac{k}{m} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{z}(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \\ 0 \end{bmatrix} u(t) \\ z(t) = [0 \quad 1] \begin{bmatrix} \dot{z}(t) \\ z(t) \end{bmatrix} \end{cases}$$

$$A = \begin{bmatrix} -\frac{d}{m} & -\frac{k}{m} \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{m} \\ 0 \end{bmatrix}, C = [0 \ 1], D = 0$$

$$x(t) = \begin{bmatrix} \dot{z}(t) \\ z(t) \end{bmatrix}, y(t) = z(t)$$

Example

RLC circuit



$$i(t) = C\frac{de_o(t)}{dt}$$

$$e_i(t) = Ri(t) + L\frac{di(t)}{dt} + e_o(t)$$

$$\Downarrow$$

$$e_i(t) = RC\frac{de_o(t)}{dt} + LC\frac{d^2e_o(t)}{dt^2} + e_o(t)$$

Rewrite the above with state space

$$\begin{cases} \begin{bmatrix} \ddot{e}_o(t) \\ \dot{e}_o(t) \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{LC} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{e}_o(t) \\ e_o(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{LC} \\ 0 \end{bmatrix} e_i(t) \\ e_o(t) = [0 \quad 1] \begin{bmatrix} \dot{e}_o(t) \\ e_o(t) \end{bmatrix} \end{cases}$$

$$A = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{LC} \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{LC} \\ 0 \end{bmatrix},$$

$$C = [0 \quad 1], D = 0$$

$$x(t) = \begin{bmatrix} \dot{e}_o(t) \\ e_o(t) \end{bmatrix}, y(t) = e_o(t)$$

$$u(t) = e_i(t)$$

Stability

Consider $u(t) = 0 \ (\forall t)$ ...

$$\dot{x}(t) = Ax(t)$$

➡ $\dot{x}(t) = e^{At}x(0)$

if $A < 0$, $x(t) \to 0 \ (t \to \infty)$

$A < 0$: All the eigenvalues of A is smaller than 0

The stability of the system can be checked by the eigenvalues of the matrix A

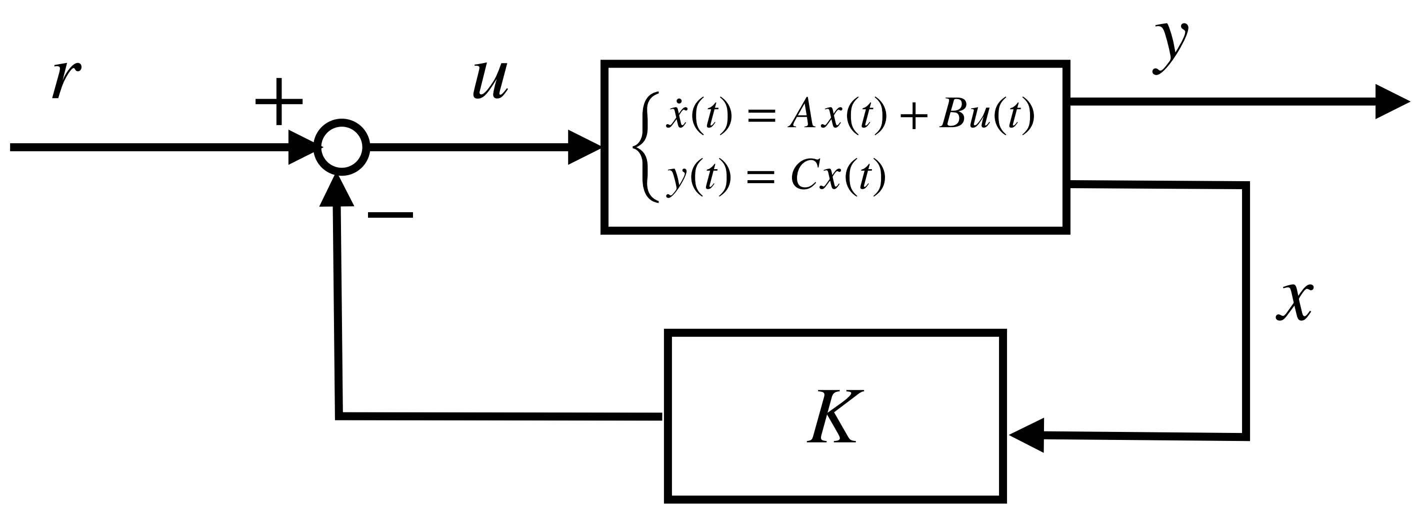
Similar to the case $\dot{x}(t) = ax(t)$ (a is a scalar value)

➡ $\dot{x}(t) = e^{at}x(0)$

if $a < 0$, $x(t) \to 0 \ (t \to \infty)$ ➡ diverge

if $a > 0$, $x(t) \to \infty \ (t \to \infty)$ ➡ converge

Feedback control



K : State feedback (matrix)

$u(t) = r(t) - Kx(t)$

substitute to state space...

$\dot{x}(t) = (A - BK)x(t) + Br(t)$

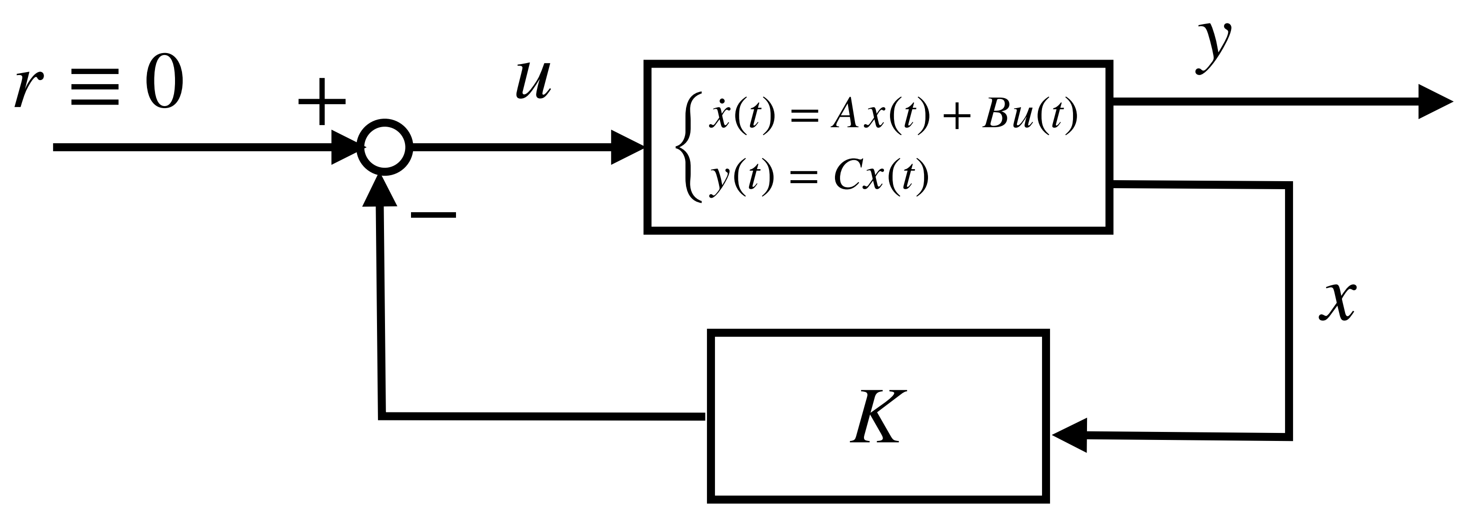
if $r(t) = 0 \ (\forall t)$ ...

$$\dot{x}(t) = e^{(A-BK)t}x(0)$$

If the plant is unstable (not $A < 0$), design K to be

$A - BK < 0$

Practice



Find state feedback controller $u = -Kx(t)$

which stabilize the following plant

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

Practice

Pole assignment

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

$$|\lambda I - A| = \lambda^2 - 3\lambda + 2 = (\lambda - 1)(\lambda - 2) = 0$$

Eigenvalues: 1, 2 ➡ Unstable

if $K = [k_1 \ k_2]$

➡ $$A - BK = \begin{bmatrix} 0 & 1 \\ -2 - k_1 & 3 - k_2 \end{bmatrix}$$

$$|\lambda I - (A - BK)| = \lambda^2 + (-3 + k_2)\lambda + 2 + k_1 = 0$$

$K = [4 \ 8]$

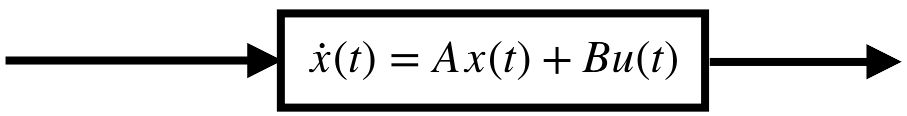
$$\begin{aligned} |\lambda I - (A - BK)| &= \lambda^2 + 5\lambda + 6 \\ &= (\lambda + 2)(\lambda + 3) = 0 \end{aligned}$$

Eigen values: -2, -3

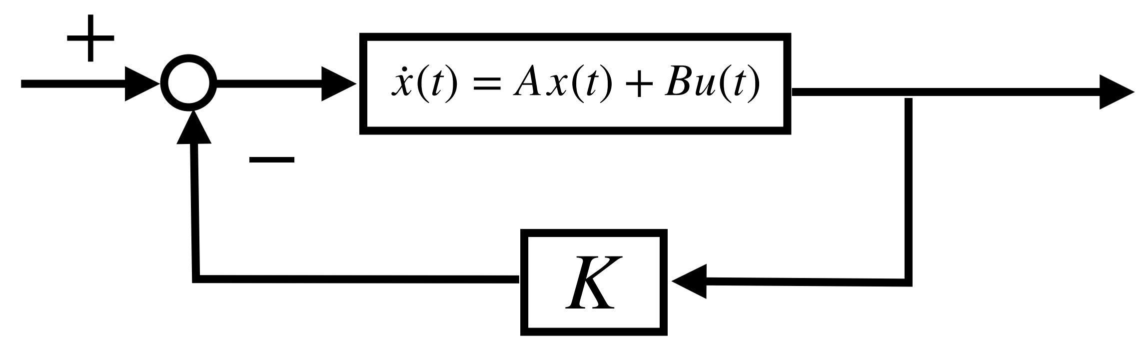
➡ Stable

Answer: $K = [4 \ 8]$

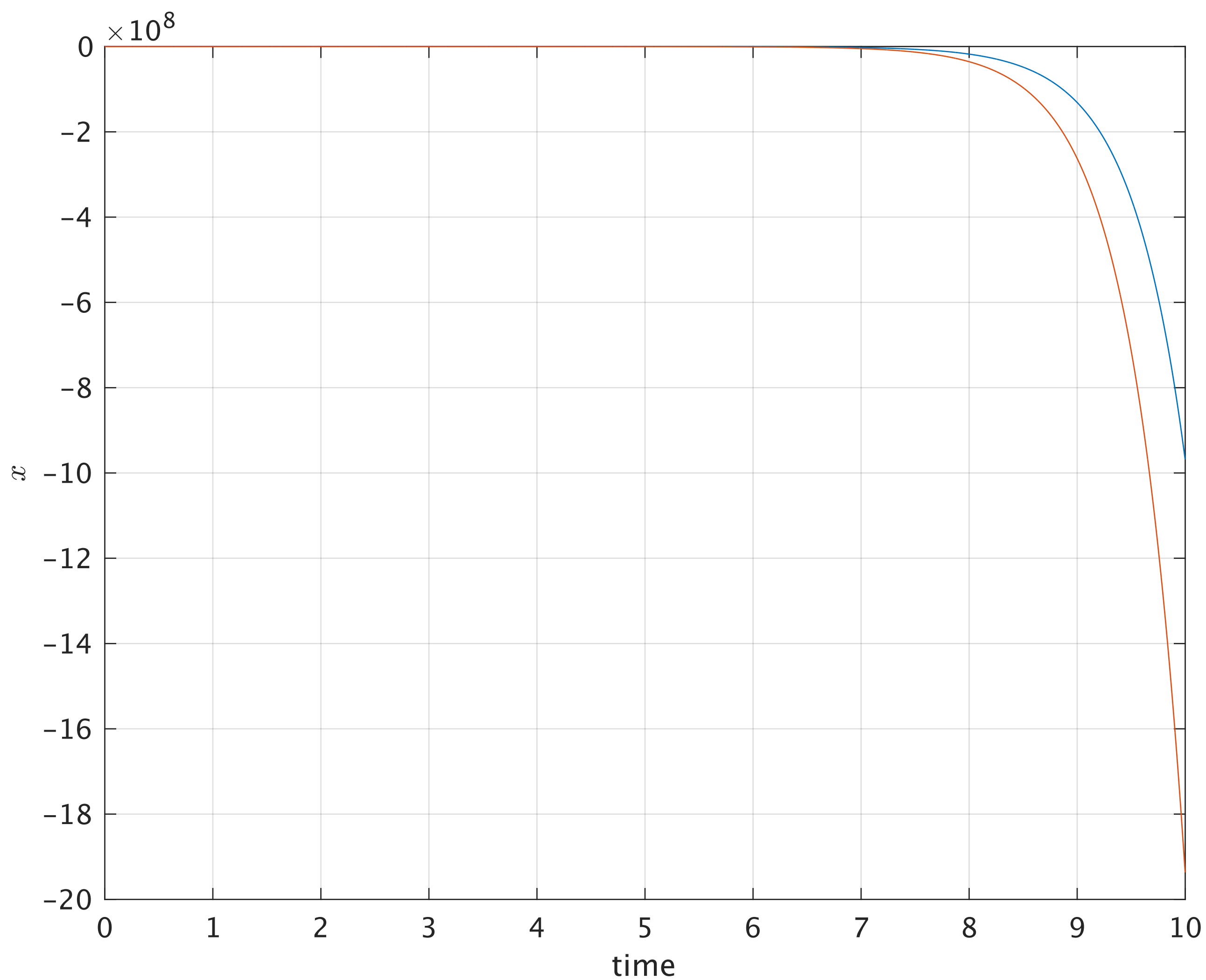
Practice



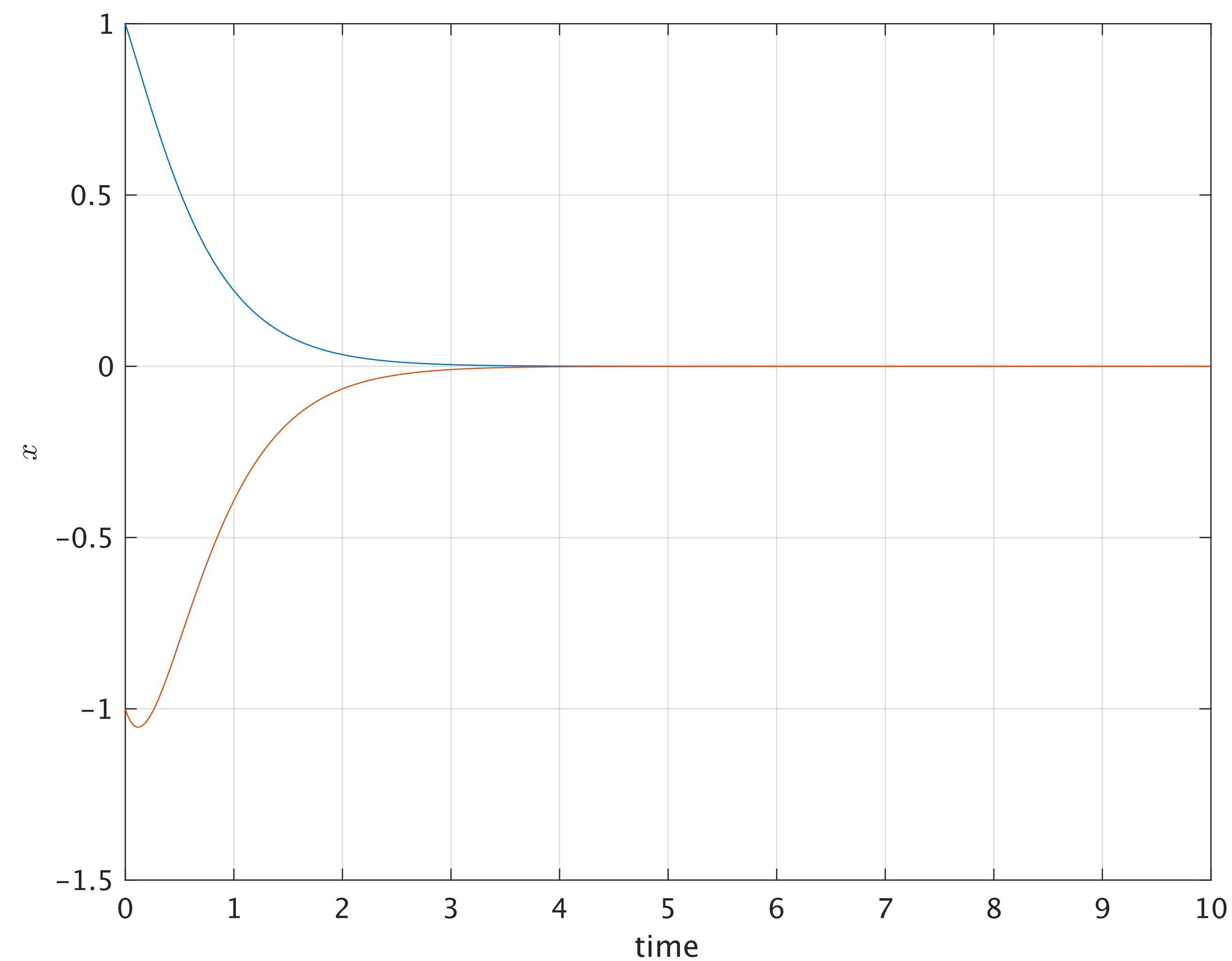
$\dot{x}(t) = Ax(t) + Bu(t)$



+
−
$\dot{x}(t) = Ax(t) + Bu(t)$
K



$\times 10^8$
0
−2
−4
−6
−8
−10
−12
−14
−16
−18
−20
x
0 1 2 3 4 5 6 7 8 9 10
time



1
0.5
0
−0.5
−1
−1.5
x
0 1 2 3 4 5 6 7 8 9 10
time

Code Example

MATLAB

```
clear

A=[
     0 1;
    -2 3;
    ];
B=[
    0;
    1;
    ];
C=eye(2); % identity matrix

P1=ss(A,B,C,0);

% show eigenvalues of A (system poles)
disp('Eigenvalues of A')
disp(eig(A))

K=place(A,B,[-2 -3]);
P2=feedback(P1,K);

% show eigenvalues of A-BK (system poles)
disp('Eigenvalues of A-BK')
disp(eig(A-B*K))

t=0:0.001:10;      % time series
u=zeros(size(t)); % input (always 0)
x0=[1 -1];         % initial states

% simulation
y1=lsim(P1,u,t,x0);
y2=lsim(P2,u,t,x0);

figure(1)
plot(t,y1)
grid on
xlabel('time')
ylabel('$x$','Interpreter','latex')

figure(2)
plot(t,y2)
grid on
xlabel('time')
ylabel('$x$','Interpreter','latex')
```

Python (Jupyter Notebook)

```
# !pip install control       # If you use Goole Colab, install these packages by
# !pip install matplotlib    # enabling these two lines once

# %matplotlib inline        # If graphs do not appear, enable this line

import numpy as np
from control.matlab import *
import matplotlib.pyplot as plt
```

```
A='0 1 ; -2 3 '
B='0 ; 1'
C='1 0; 0 1'

P1=ss(A,B,C,0)
print(P1)
print(np.linalg.eigvals(P1.A))
```

```
K=place(P1.A,P1.B,[-2,-3])
print(K)
```

```
P2=feedback(P1,K)
print(P2)
print(np.linalg.eigvals(P2.A))
```

```
t=np.arange(0,10,0.001)
u=np.zeros(t.size)
x0=np.array([[1],[-1]])
y1, t, x1out=lsim(P1,u,t,x0)
y2, t, x2out=lsim(P2,u,t,x0)
```

```
plt.plot(t,y1)
plt.xlabel('time')
plt.ylabel('x')
```

```
plt.plot(t,y2)
plt.xlabel('time')
plt.ylabel('x')
```

For non-MATLAB user

You can use python for control simulation

Required modules (All of them are available by PIP)

- numpy
- python-control (“control” in PyPI)
- matplotlib

If you are also non-python user, you can use Google Colaboratory

(It needs Google account)

https://colab.research.google.com/