Advanced Dynamical Systems Control

動的システム制御論

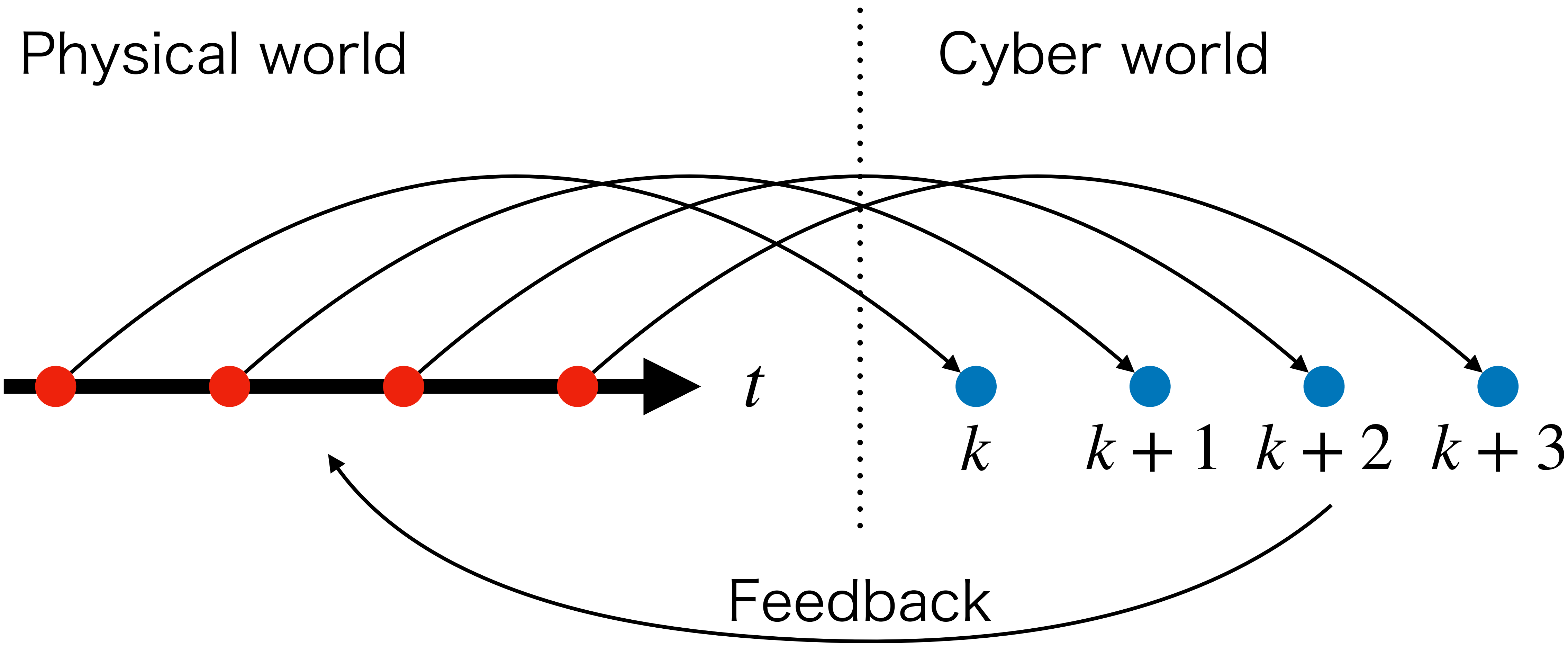
#2

ver. 0.1

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

Sample-data control

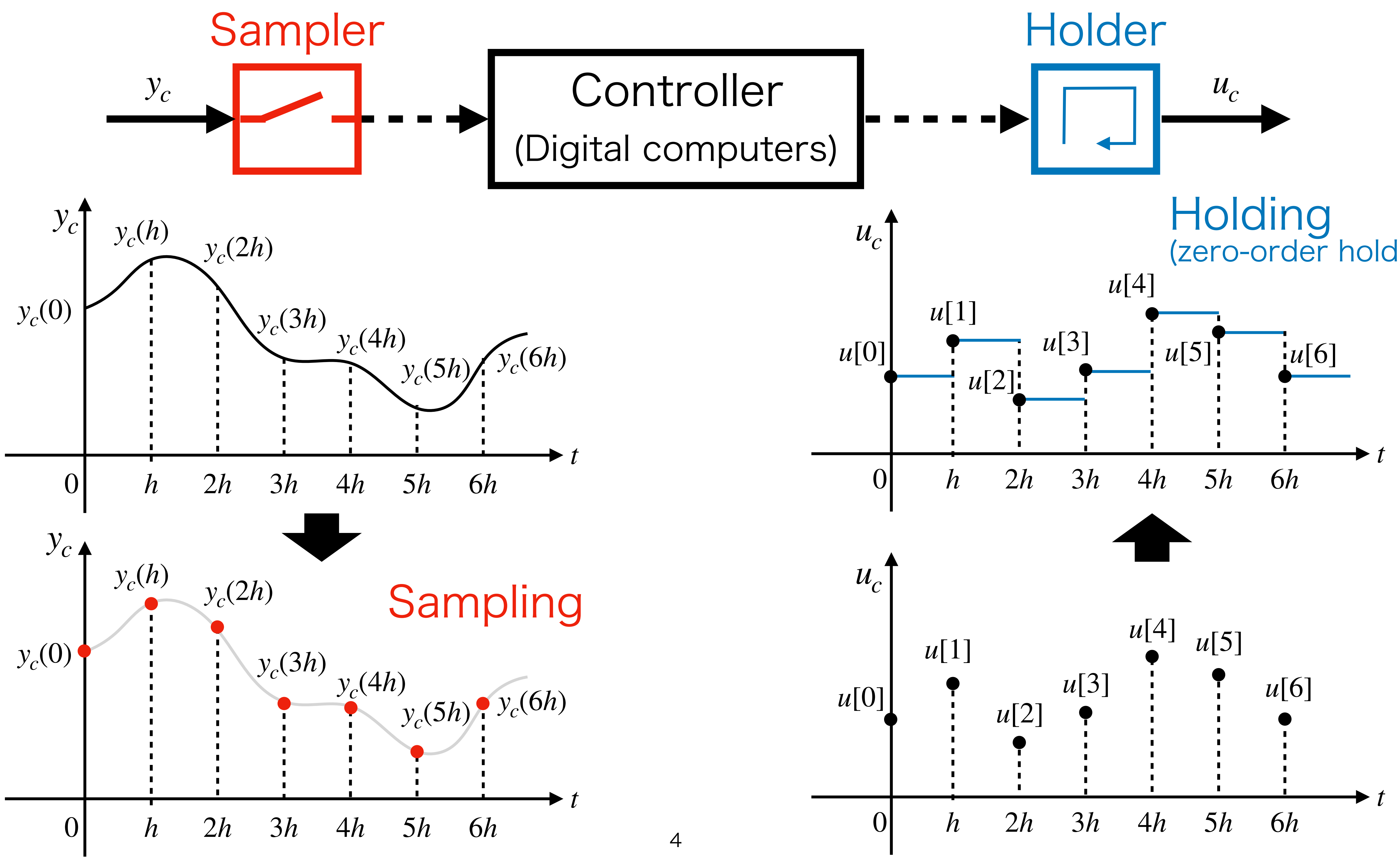


Continuous-time system

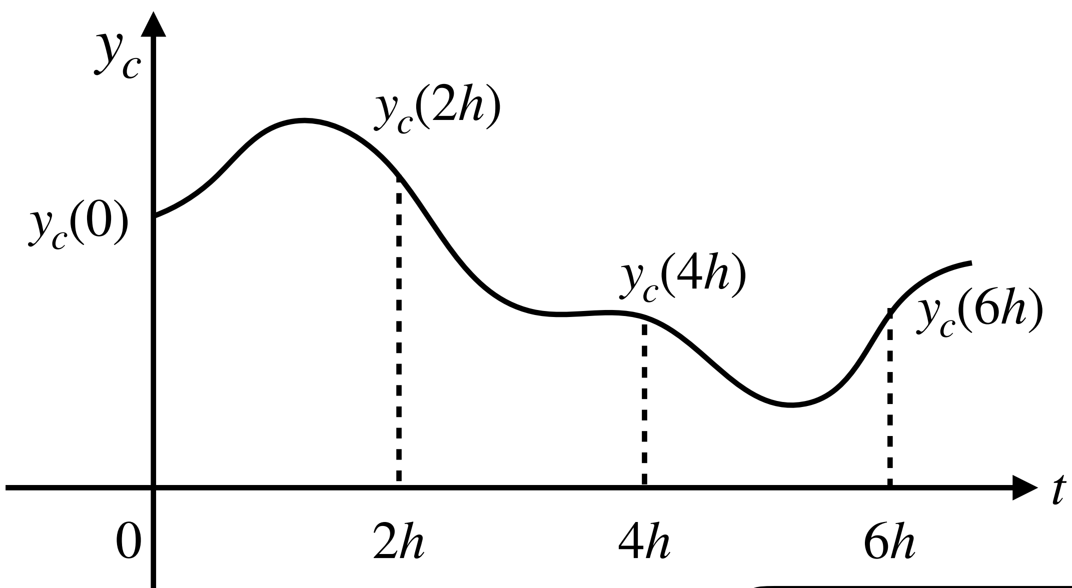
Discrete-time system

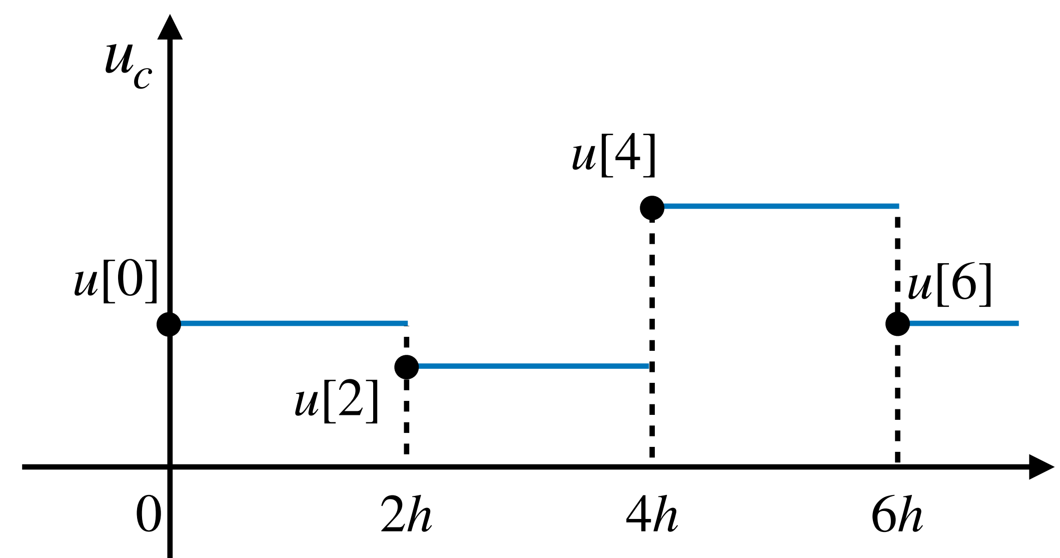
To sample data of physical phenomena and control it by calculating converted discrete data.

Sample-data control

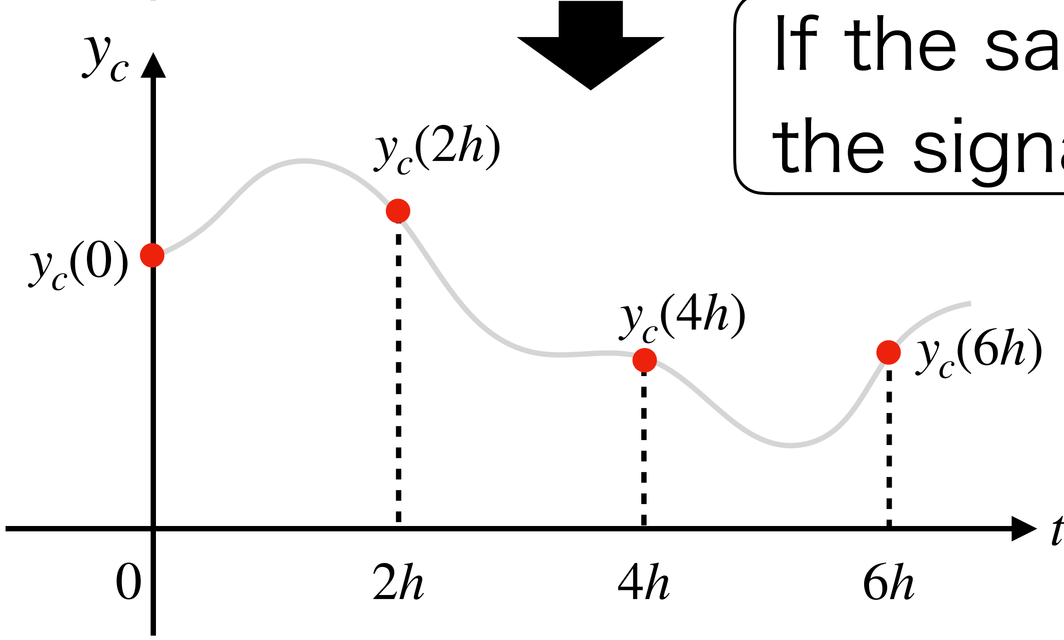


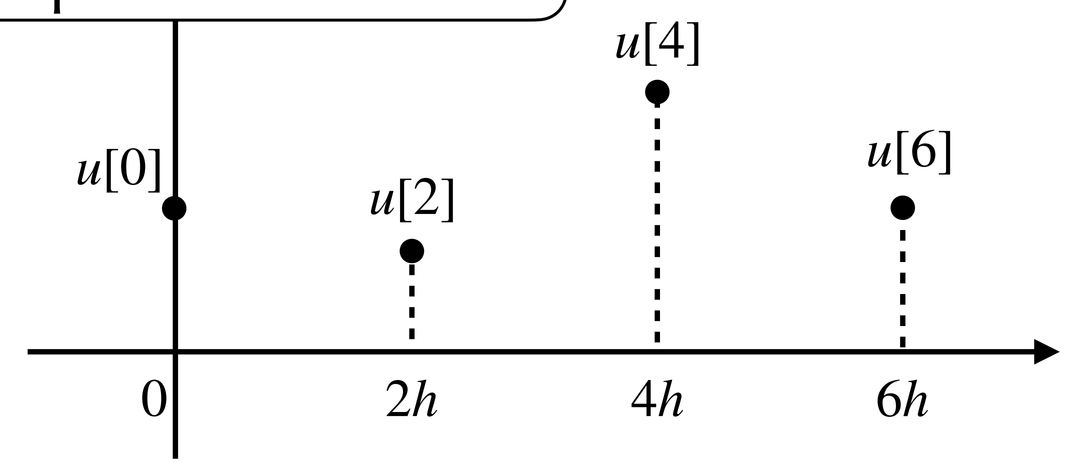
Sample-data control





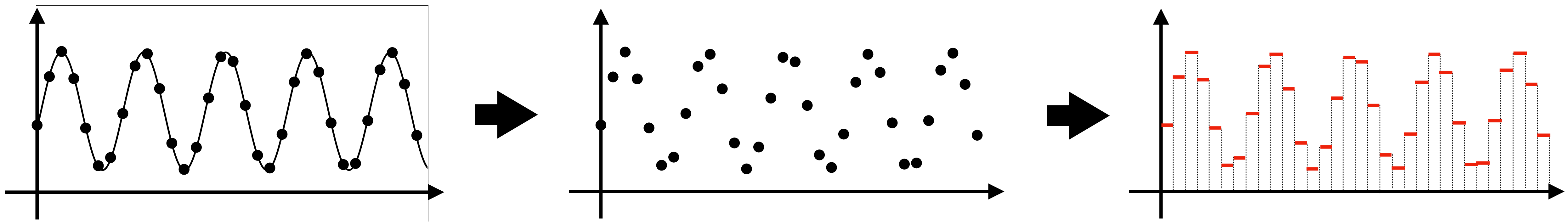
If the sampling period h is larger,
the signals become quite different



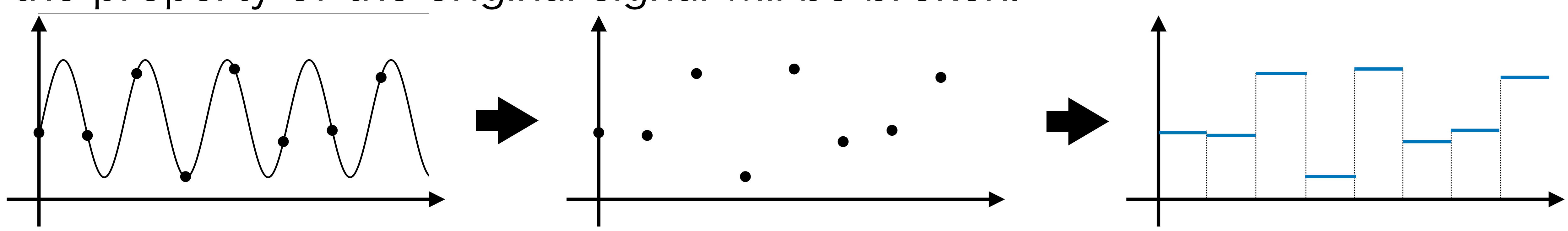


Sampling theorem

If sampling period h is small enough,
we can reconstruct almost the same signal in the discrete-time domain



If sampling period h is large,
the property of the original signal will be broken.



Nyquist frequency f_N: maximum frequency of reconstructable signal.

The sampling frequency should be more than twice as large as the original signal frequency.

If sampling period is h (sampling frequency is $\frac{1}{h}$), $f_N = \frac{1}{2h}$.

Continuous-time system
vs
Discrete-time system

Discrete-time linear system

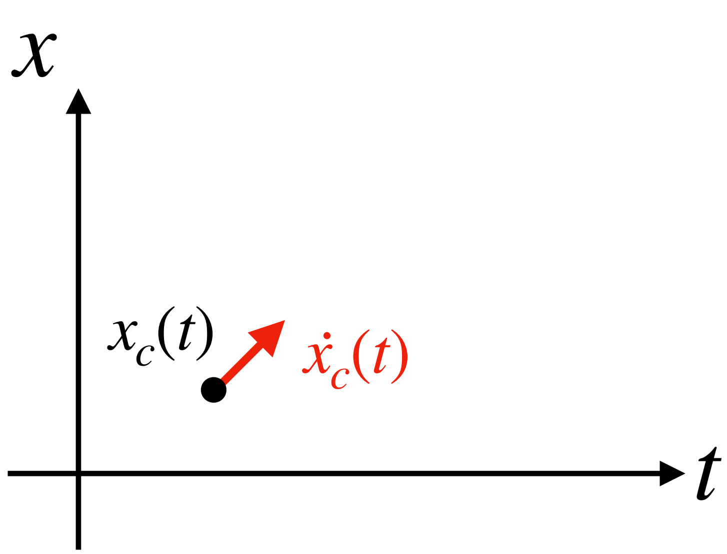
The state-space forms are similar to each other,
but the their meaning are little different

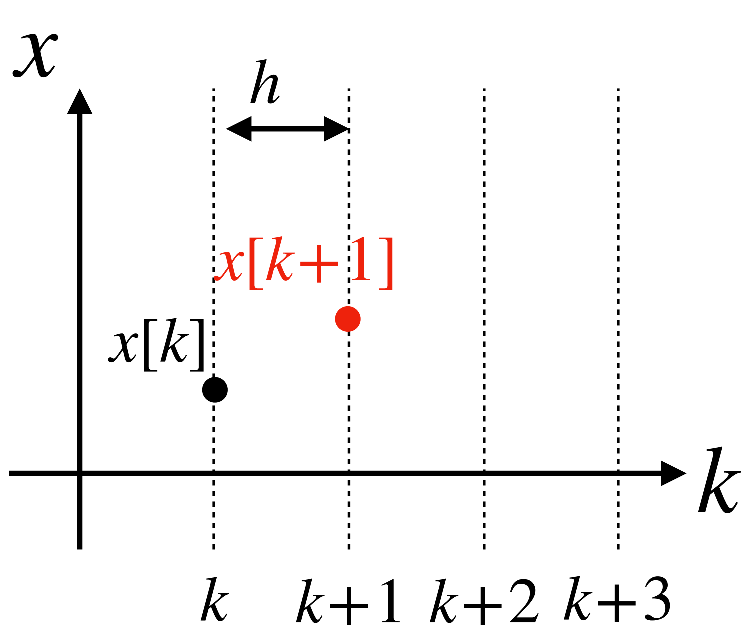
Continuous-time system

$$\dot{x}_c(t) = A_c x_c(t) + B_c u_c(t)$$
$$y_c(t) = C_c x_c(t) + D_c u_c(t)$$

Discrete-time system

$$x[k+1] = Ax[k] + Bu[k]$$
$$y[k] = Cx[k] + Du[k]$$





h : Sampling period

$$x[k] = x_c(th)$$

Discrete-time linear system

Continuous-time system

$$\dot{x}_c(t) = A_c x_c(t) + B_c u_c(t)$$

$$\Downarrow$$

$$x_c(t) = e^{A_c(t-t_0)} x_c(t_0) + \int_{t_0}^{t} e^{A_c(t-\tau)} B_c u_c(\tau) d\tau$$

replace $t \to th + h$ and $t_0 \to th$,

$$x_c(th+h) = e^{A_c h} x_c(th) + \int_{th}^{th+h} e^{A_c(th+h-\tau)} B_c u_c(\tau) d\tau$$

In discrete-time systems, $u(t)$ is fixed in a sampling period.

So,

$$A = e^{A_c h} \quad B = \int_0^h e^{A_c \tau} B_c d\tau$$

for the discrete-time system $x[k+1] = Ax[k] + Bu[k]$

Example

Social systems are often expressed by discrete-time models

Bank deposit model (extremely simplified)

- Interest rates: 0.1% per year

balance of this month

$$x[k+1] = \frac{0.1}{100 \times 12} x[k] + u[k]$$

balance of next month

deposit of this month

Some properties of discrete-time system

Stability

Continuous-time system

$$\dot{x}_c(t) = A_c x_c(t) + B_c u_c(t)$$

$$\Downarrow$$

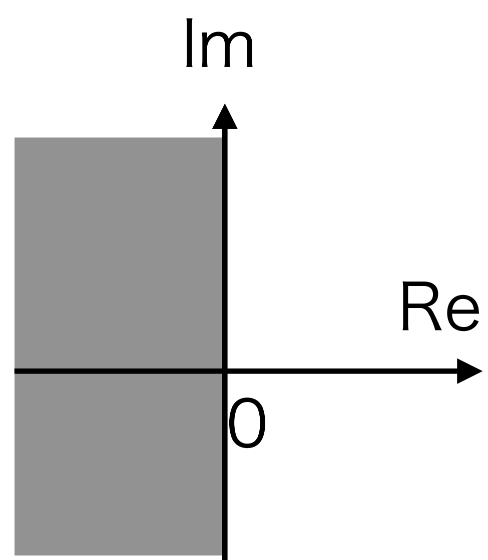
$$x_c(t) = e^{A_c t} x_c(0) + \int_0^t e^{A_c(t-\tau)} B_c u_c(\tau) d\tau$$

If $u(t) \equiv 0$

$$x_c(t) = e^{A_c t} x_c(0)$$

Stable $\Leftrightarrow \lim_{t \to \infty} e^{A_c t} \to 0$

All the eigenvalues of A_c are on the left-half plane



Discrete-time system

$$x[k+1] = Ax[k] + Bu[k]$$

$$\Downarrow$$

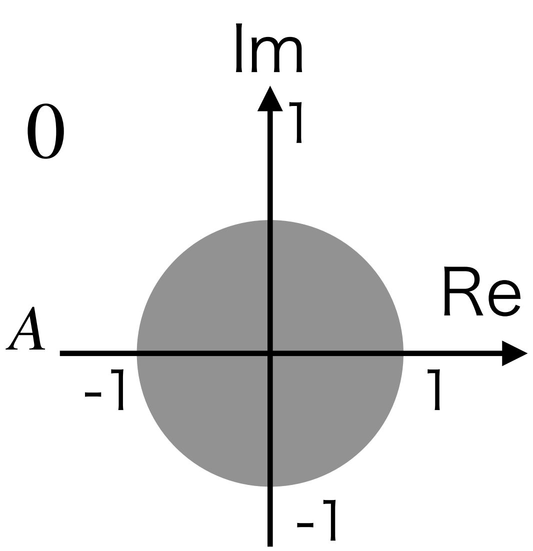
$$x[k] = A^k x[0] + \sum_{i=0}^{k} A^{k-i} Bu[i]$$

If $u(t) \equiv 0$

$$x[k] = A^k x[0]$$

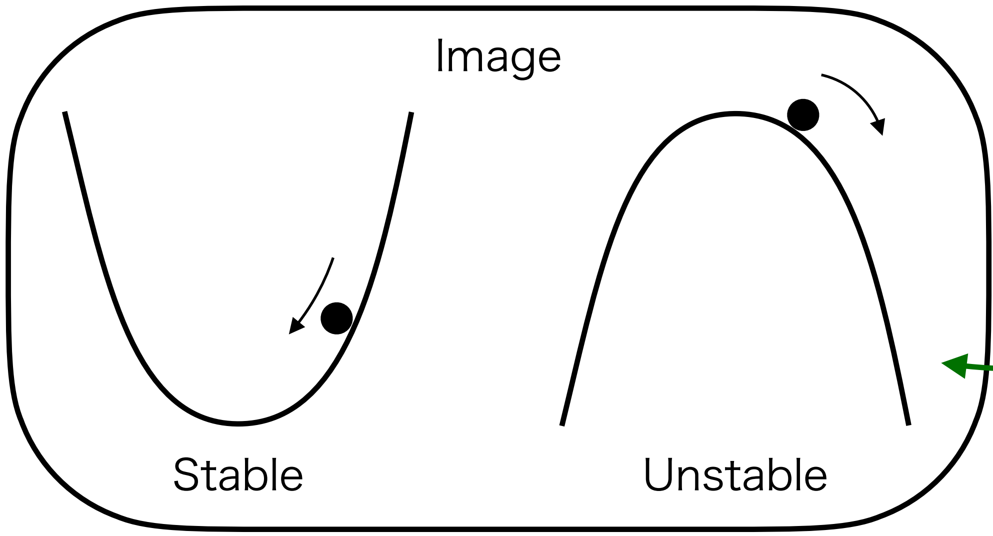
Stable $\Leftrightarrow \lim_{k \to \infty} A^k \to 0$

All the eigenvalues of A are in the unit circle



Stability

Lyapunov stability criterion



$$x[k+1] = Ax[k]$$

Lyapunov function candidate

$$P(x) = x^T P x$$

P : Positive definite $(x^T P x > 0 \ (x \neq 0))$

$$P(x[k+1]) - P(x[k]) = x^T[k](A^T P A - P)x[k]$$

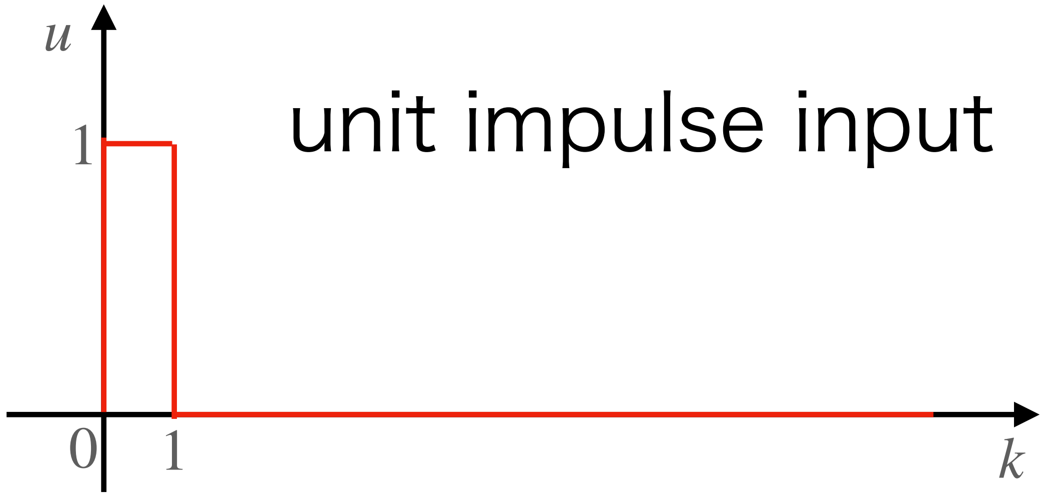
If positive definite matrices P, Q,
which satisfy

$$P = A^T P A + Q,$$

are exist, A is called **stable**

Impulse response

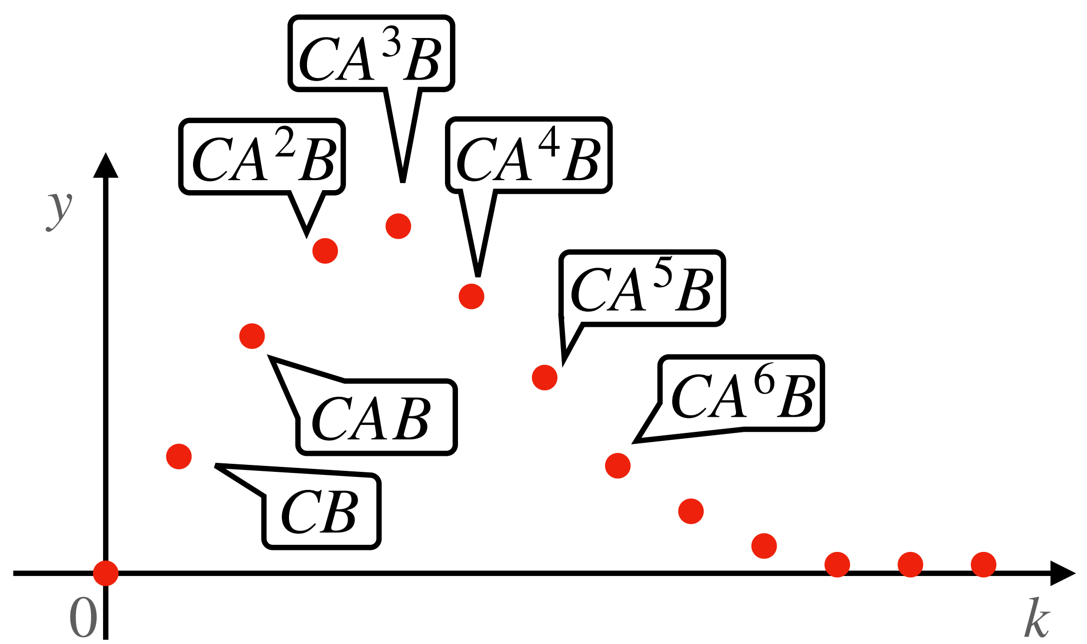
$$x[k+1] = Ax[k] + Bu[k]$$
$$y[k] = Cx[k]$$



$$y[k] = CA^k x[0] + \sum_{i=0}^{k} \boxed{CA^{k-i-1}B} u[i]$$

Impulse response matrix
or
Marcov parameter

If u is unit impulse input,
the output is written by only
the impulse response matrix

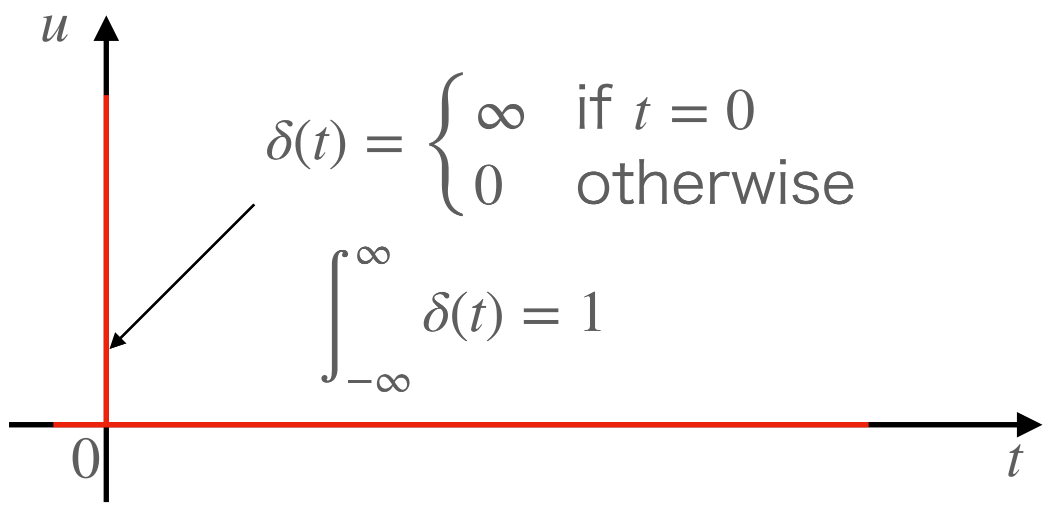


Impulse response

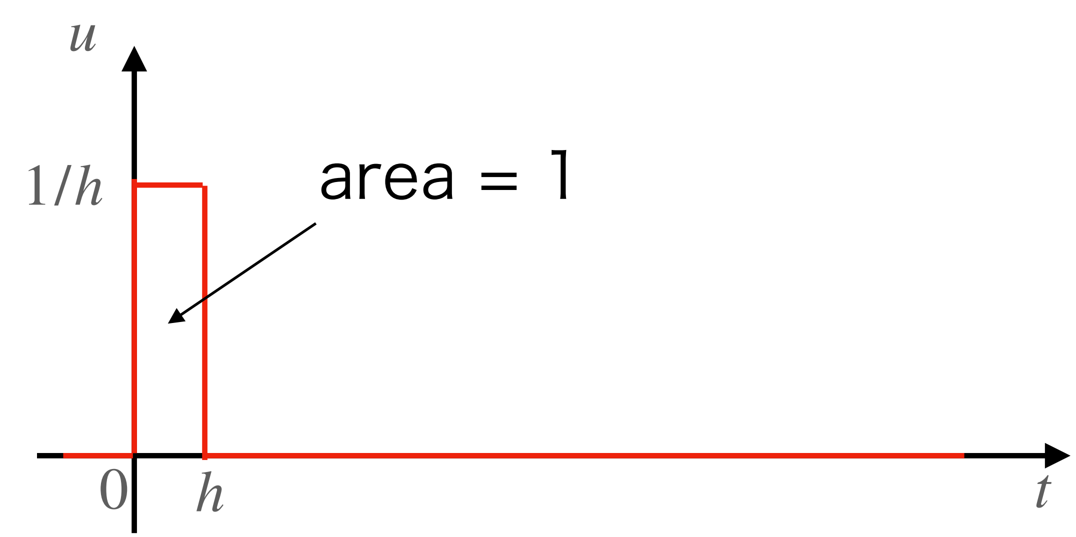
Note

To obtain output which which corresponds to the impulse response in continuous-time system,
the input should be a unit area pulse.

continuous-time



discrete-time



Code Example

MATLAB

```
clear

A=[
    0 4;
    -6 -4;
    ];
B=[1;0];
C=[1 0];
sys1=ss(A,B,C,0);
sys2=c2d(sys1,0.1);
sys3=c2d(sys1,1);
disp('Eigenvalues of A')
disp(eig(A))
disp('Absolute eigenvalues of A (h=0.1)')
disp(abs(eig(sys2.A)))
disp('Absolute eigenvalues of A (h=1)')
disp(abs(eig(sys3.A)))


markov2=zeros(1,51);
markov3=zeros(1,51);
for k=2:51
    markov2(k)=sys2.C*sys2.A^(k-2)*sys2.B;
    markov3(k)=sys3.C*sys3.A^(k-2)*sys3.B;
end

figure(1)
impulse(sys1,sys2,sys3,5)
hold on
% plot Markov parameter divided by sampling period
stairs((0:50).*0.1,markov2./0.1,'Marker','o','LineStyle','none')
stairs((0:5),markov3(1:6),'Marker','o','LineStyle','none')
grid on
legend('continuous','h=0.1','h=1',...
    'markov parameter (h=0.1)','markov parameter (h=1)')
hold off
```

Python (Jupyter Notebook)

```
# !pip install control
# %matplotlib inline

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

```
A=' 0 4 ; -6 -4 '
B=' 1 ; 0 '
C=' 1 0 '

sys1=ss(A,B,C,0)
print(sys1)
print(np.linalg.eigvals(sys1.A))
```

```
sys2=c2d(sys1,0.1)
sys3=c2d(sys1,1)

print(sys2)
print(abs(np.linalg.eigvals(sys2.A)))

print(sys3)
print(abs(np.linalg.eigvals(sys3.A)))
```

```
markov2=[]
markov3=[]
for k in range(50):
  markov2.append(sys2.C*np.linalg.matrix_power(sys2.A,k)*sys2.B)
  markov3.append(sys3.C*np.linalg.matrix_power(sys3.A,k)*sys3.B)

markov22=(np.array(markov2).reshape(-1))/0.1
markov33=(np.array(markov3).reshape(-1))
```

```
t1=np.arange(0,5,0.001)
t2=np.arange(0,5,0.1)
t3=np.arange(0,5,1)
y1,T1=impulse(sys1,t1)
y2,T2=impulse(sys2,t2)
y3,T3=impulse(sys3,t3)
```

```
fig, ax = plt.subplots()
ax.plot(T1,y1,label='continuous')
ax.plot(T2,y2,label='h=0.1')
ax.plot(T3,y3,label='h=1')
ax.plot(np.arange(0.1,5.1,0.1),markov22,linestyle='none',marker='o',label='markov parameter (h=0.1)')
ax.plot(np.arange(1,6,1),markov33[0:5],linestyle='none',marker='o',label='markov parameter (h=1)')
ax.legend()
```