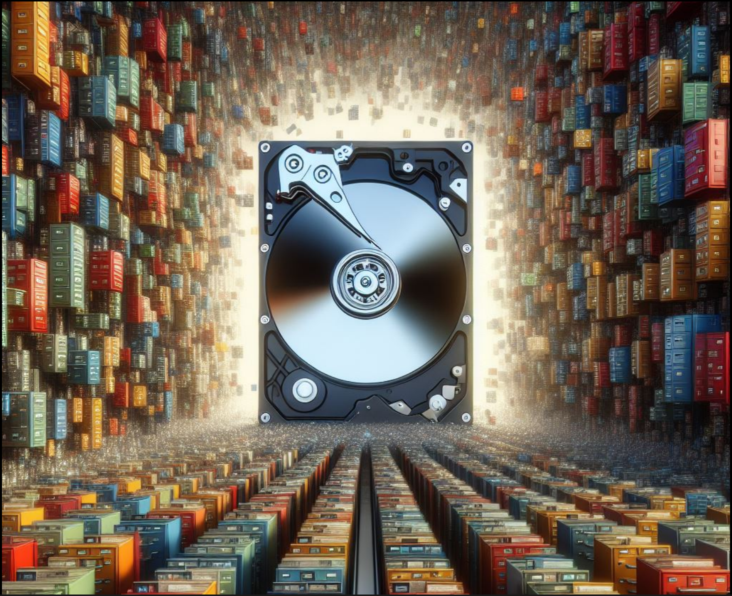
Introduction to Modern Controls

Discretization of State-Space System Models

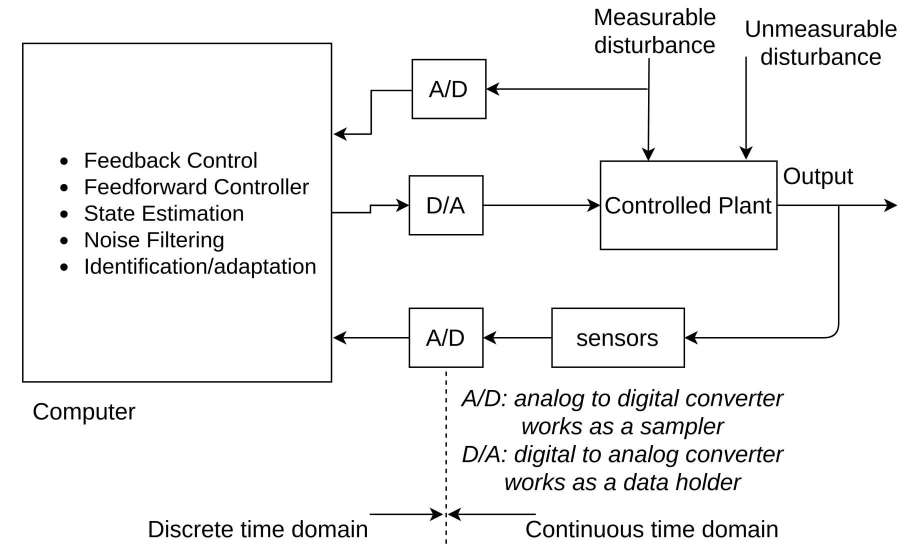
1TB vs 1,300 filing cabinets of paper



Inherent sampling in practice

$$\Delta t = \frac{1}{(\text{rpm}/60) \times \text{sector number}}$$

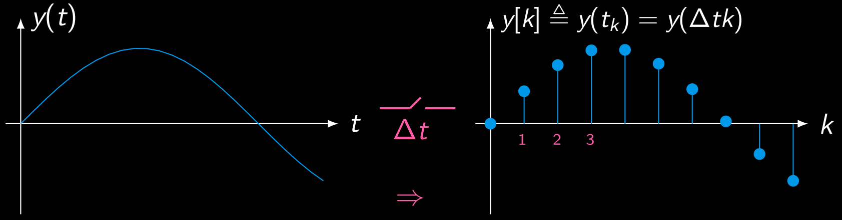
Practical control systems



Sampler

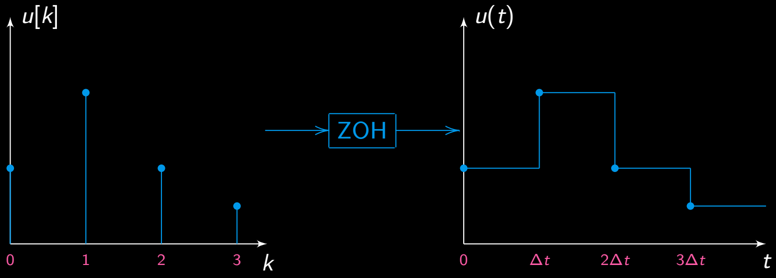
- sampler: converts a time function into a discrete sequence,

e.g.,



Signal holding

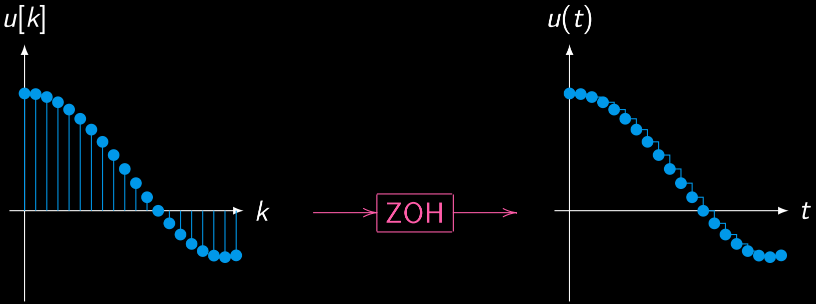
- Zero-order Hold (ZOH): converts a sequence into a “stair-case” time function, e.g.,



- $u(t) = u[k]$ for $t \in [k\Delta t, (k+1)\Delta t)$

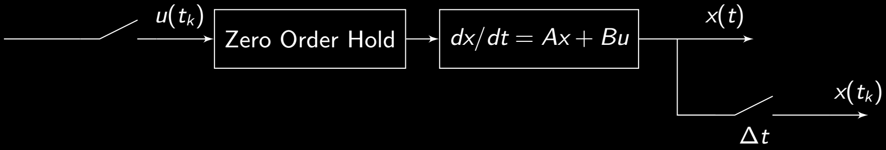
Signal holding

- more faithful presentation with fast sampling



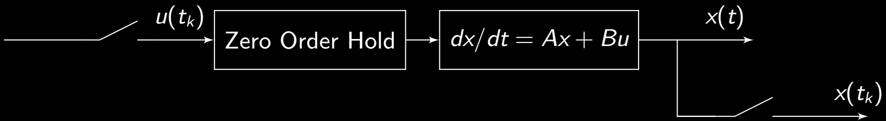
Problem definition

continuous-time system preceded by a ZOH:



- $u(t_k)$: discrete-time input
- $x(t)$: continuous-time output
- $x(t_k)$: sampled discrete-time output
- Δt: sampling time
- goal: to obtain the model between $u(t_k)$ and $x(t_k)$

Solution



- starting from t_k, the solution of $\dot{x} = Ax + Bu$ at time t_{k+1} is

$$
\begin{aligned}
x(t_{k+1}) &= e^{A(t_{k+1}-t_k)}x(t_k) + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau_o)} Bu(\tau_o)d\tau_o \\
&= e^{A(\overbrace{t_{k+1} - t_k}^{\Delta t})}x(t_k) + u(t_k) \underbrace{\int_{t_k}^{t_{k+1}} e^{A(\overbrace{t_{k+1} - \tau_o}^{\eta})} Bd\tau_o}_{=\int_{\Delta t}^{0} e^{A\eta} Bd(-\eta) = -\int_{\Delta t}^{0} e^{A\eta} Bd\eta}
\end{aligned}
$$

- noting $-\int_{\Delta t}^{0} e^{A\eta} Bd\eta = \int_0^{\Delta t} e^{A\tau} Bd\tau$ and denoting t_k as k yield

$$
\boxed{x[k+1] = A_d x[k] + B_d u[k],\ A_d = e^{A\Delta t},\ B_d = \int_0^{\Delta t} e^{A\tau} Bd\tau}
$$

Mapping of eigenvalues

$$\boxed{x[k+1] = A_d x[k] + B_d u[k],\ A_d = e^{A\Delta t},\ B_d = \int_0^{\Delta t} e^{A\tau} B d\tau}$$

- diagonalization / Jordan form: $A = T^{-1}\Lambda T$
- e^{At} has the same eigenvalues as $e^{\Lambda t}$
- $\Rightarrow$ eigenvalues of $A_d = e^{A\Delta t}$ are $e^{\lambda_i \Delta t}$'s where λ_i is an eigenvalue of A

Example

$$\dot{x}(t) = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B} u(t)$$

$$y(t) = \underbrace{\begin{bmatrix} \frac{1}{m} & 0 \end{bmatrix}}_{C} x(t)$$

discretization at a sampling time of $\Delta t \Rightarrow$

$$A_d = e^{A\Delta t} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \; B_d = \int_0^{\Delta t} e^{A\tau} B d\tau = \int_0^{\Delta t} \begin{bmatrix} \tau \\ 1 \end{bmatrix} d\tau = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}$$

$$C_d = C$$

Numerical example in Python

```
import control
import numpy
m = 1
dt = 0.1
A = [[0, 1], [0, 0]]
B = [[0], [1]]
C = [[1/m, 0]]
D = 0

G_s = control.ss(A, B, C, D)
G_z = control.c2d(G_s, dt, 'zoh')
print(G_z.A)

# eigenvalues of continuous-time system
eigA, eigvecA = numpy.linalg.eig(A)
print(eigA)
# eigenvalues of discretized system
eigAd, eigvecAd = numpy.linalg.eig(G_z.A)
print(eigAd)
```

Spectral mapping theorem

- eigenvalues of $A_d = e^{AT}$ are $e^{\lambda_i T}$'s where λ_i is an eigenvalue of A
- more generally: take any $X \in \mathbb{C}^{n\times n}$ and a polynomial function $f(\cdot)$ (more generally, analytic functions)
- e.g.:

$$A = \begin{bmatrix} 99.8 & 2000 \\ -2000 & 99.8 \end{bmatrix} = 99.8I + 2000 \overbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}^{X}$$

- then

$$\text{eig}\,(f(X)) = f(\text{eig}\,(X))$$

- e.g.:

$$A = \begin{bmatrix} 99.8 & 2000 \\ -2000 & 99.8 \end{bmatrix} = 99.8I + 2000 \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$\lambda(A) = 99.8 + 2000\lambda \left\{ \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \right\} = 99.8 \pm 2000i$$

Spectral mapping theorem

$$A = \begin{bmatrix} 99.8 & 2000 \\ -2000 & 99.8 \end{bmatrix}$$

```python
import numpy
A = [[99.8, 2000], [-2000, 99.8]]

eigA, eigvecA = numpy.linalg.eig(A)
print(eigA)
```

```
[99.8+2000.j 99.8-2000.j]
```