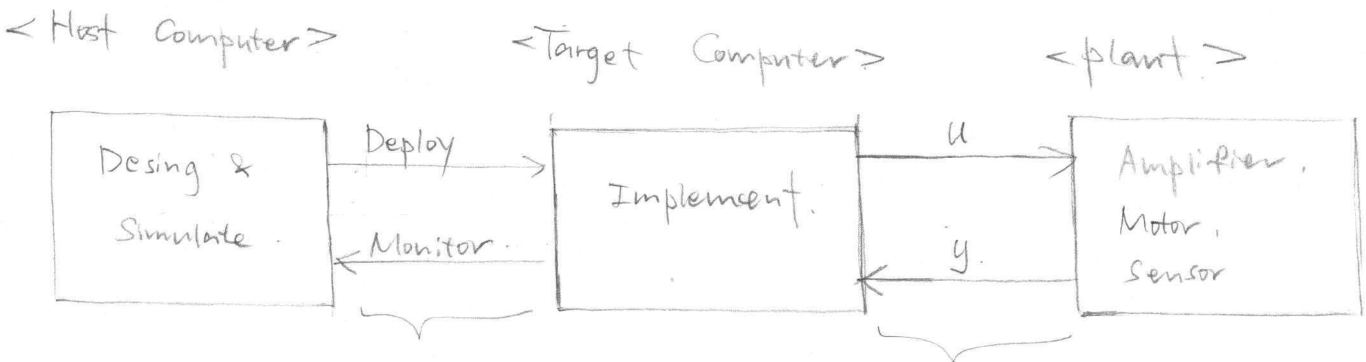
< Digital Control System >

- Objective
  - Understand the architecture of Digital Control Systems.
  - Discrete-time controller design via approximate discretization.

- Analog vs. Digital Control
  - Controller $C(s)$ can be implemented with analog circuits, or
  - We could design $C(z)$ for digital implementation.

- Digital Control Hardware



Non-deterministic Communication
e.g.) Ethernet, USB, WiFi

Deterministic Communication
e.g. Analog (ADC / DAC)
Digital (PWM / Serial).

· Non real-time OS
e.g.) Linux / Windows

· Real-time OS (e.g. RT Linux)
or, without OS

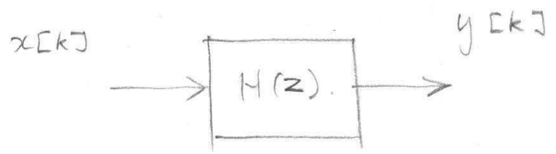
· Design & Simulate Control algorithms.

· Implement control algorithms.

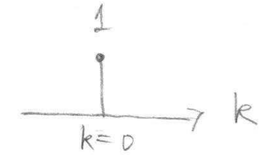
· Other names { real-time controller.
real-time target controller
⋮

· Example { Industrial PC (IPC)
Programable Automation Controller (PAC)
Programable Logic Controller (PLC)
Microcontroller / DSP / FPGA

- Discrete-time Systems.



- Impulse response



when $x[k] = \delta[k]$ "Kronecker delta"

$y[k] = h[k]$ is the impulse response.

- DT Transfer function.

$$H(z) = \mathcal{Z}\{h[k]\}$$

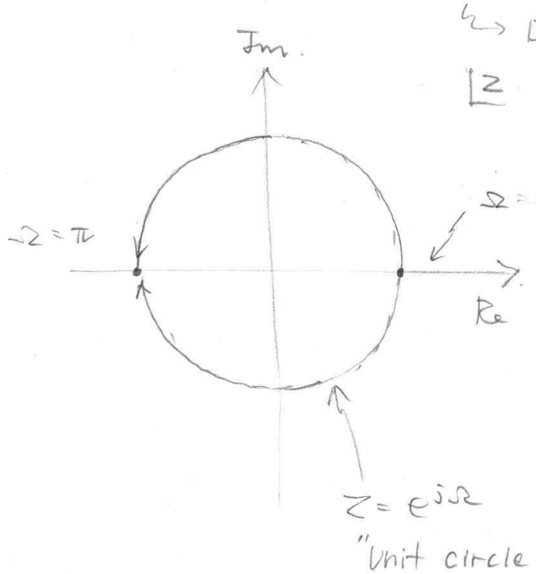
$$= \sum_{k=-\infty}^{\infty} h[k] z^{-k}$$

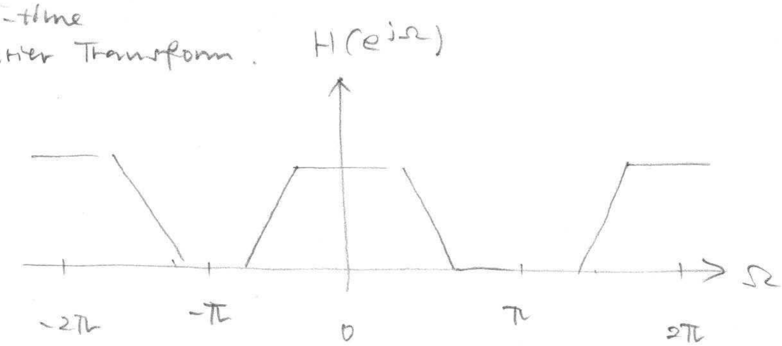
$$= \sum_{k=0}^{\infty} h[k] z^{-k} \quad \text{if} \quad h[k] = 0 \text{ for } k < 0 \text{ "causal"}$$

- DT Frequency response

$$H(e^{j\Omega}) = \mathcal{F}\{h[k]\} = \sum_{k=-\infty}^{\infty} h[k] e^{-j\Omega k} = H(z)\Big|_{z=e^{j\Omega}}$$

↳ Discrete-time Fourier Transform.



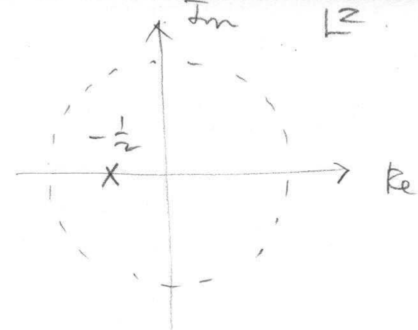


DTFT is <u>periodic!</u>.

- Note that DT signals & systems can exist for its own sake.

No need to consider underlying CT signals & system.

Example :

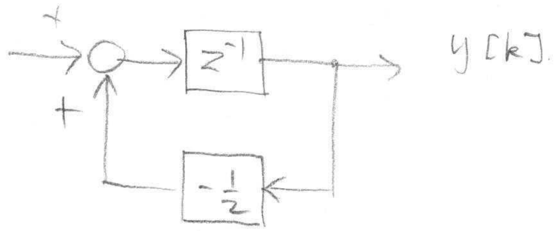


$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{z+\frac{1}{2}} = \frac{z^{-1}}{1+\frac{1}{2}z^{-1}}$$

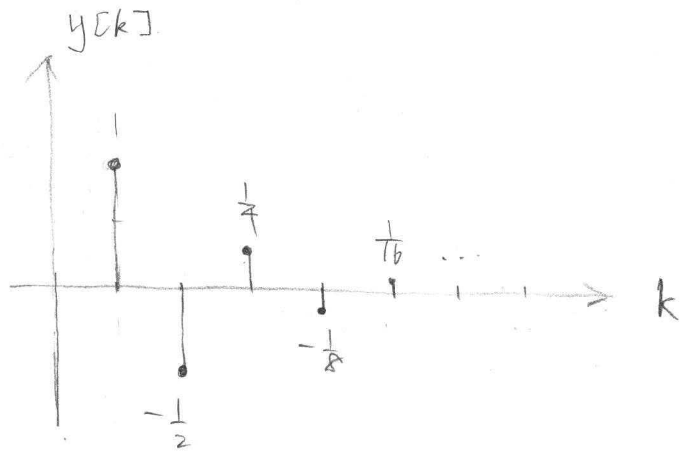
$$\rightarrow \quad Y(z)\left(1+\tfrac{1}{2}z^{-1}\right) = X(z)\,z^{-1}$$

$$\xrightarrow{\mathcal{Z}^{-1}} \quad y[k] = x[k-1] - \tfrac{1}{2}\,y[k-1]$$

Block Diagram



Impulse Response ($x[k] = \delta[k]$)

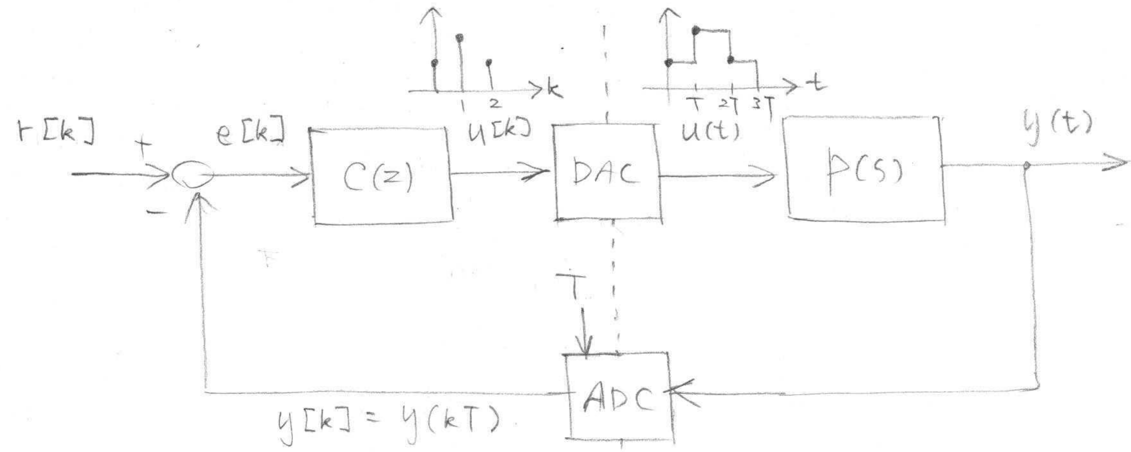


$$H(e^{j\Omega}) = \frac{1}{e^{j\Omega}+\frac{1}{2}} \qquad |H(e^{j\Omega})| = \frac{1}{\sqrt{(e^{j\Omega}+\frac{1}{2})(e^{-j\Omega}+\frac{1}{2})}} = \frac{1}{\sqrt{\frac{1}{4}+1+\frac{e^{j\Omega}+e^{-j\Omega}}{2}}}$$

$$= \frac{1}{\sqrt{\frac{5}{4}+\cos\Omega}}$$

- Note that $H(e^{j\Omega})$ is periodic with period 2π.

○ Sampled-data Systems (DT control of CT System)



· Discrete-Time signals
· Difference Equations

· Continuous-Time signals
· Differential Equations

○ Analog to Digital Converter (ADC)

- Actual ADC takes finite time for conversion (latency).

  Modeled here as an instantaneous sampler : $y[k] = y(kT)$

- Actual ADC has finite resolution (e.g. 10-16 bits)

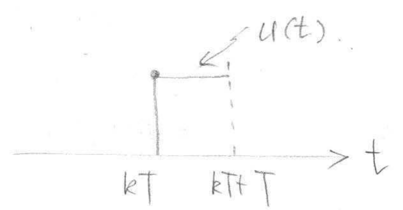
  DT signals "quantized" in amplitude $\longrightarrow$ Digital signals

  For now, we assume ADCs with infinite resolution.

○ Digital to Analog Converter (DAC)

- Extrapolates a CT signal from the past DT samples.

- Most DACs for control implement "zero-order hold" extrapolation.



$u(t) = u[k] \quad (kT \le t < kT + T)$

○ Sampling frequency : $f_s = \frac{1}{T}$ $\left(\omega_s = \frac{2\pi}{T} \right)$

· Select f_s such that $f_s > 10 f_c$ at least. ($f_c = 2\pi \omega_c$)

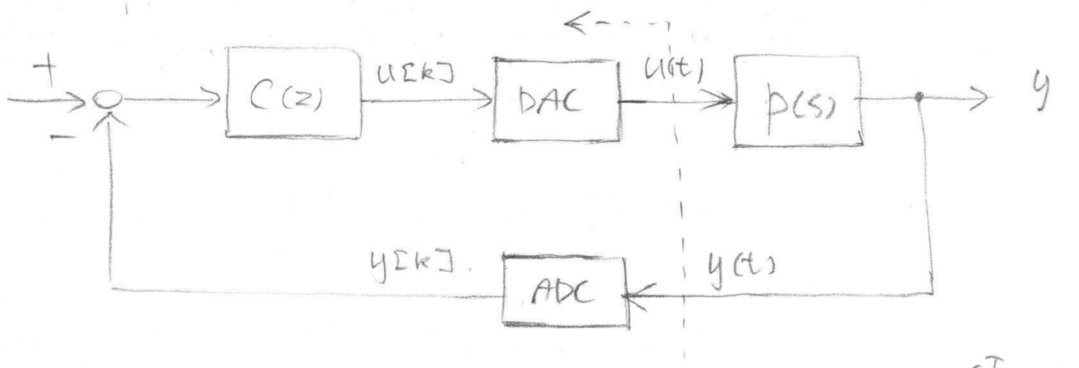
crossover freq.

$\left\{ \begin{array}{l} \cdot\ f_s > 20 f_c \\ \cdot\ f_s < 20 f_c \end{array} \right.$

$f_s > 20 f_c$: Less worries about DT effect

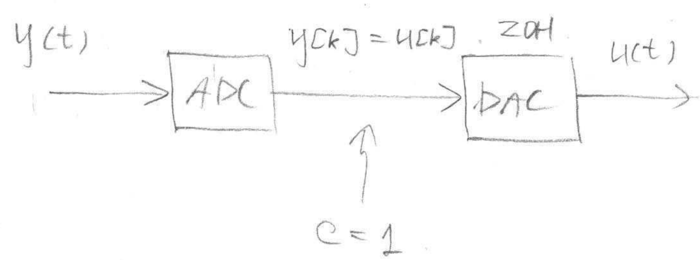
$f_s < 20 f_c$: Need to account for DT effect

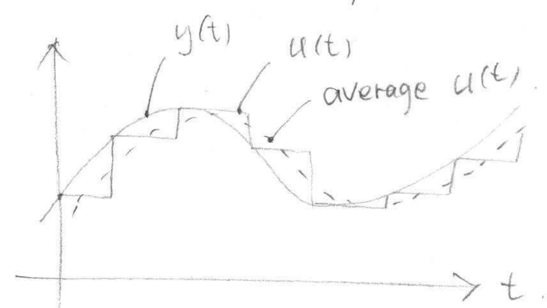
- Indirect design via DT approximations. (a.k.a. Emulation)

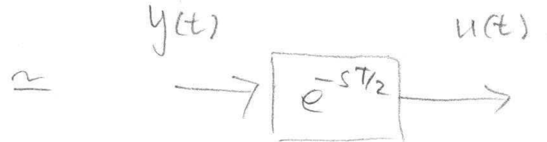


$\simeq C(s) \cdot e^{-s\frac{T}{2}}$

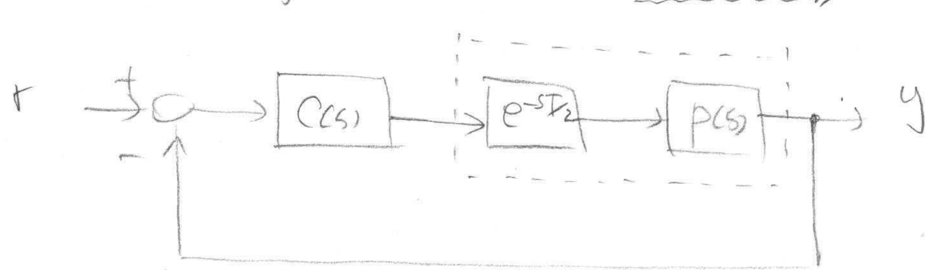
- We design a CT controller $C(s)$ and find a DT controller $C(z)$ that approximately implements $C(s)$.
- It gives satisfactory results especially when the delay is accounted for.
- The ZOH can be modeled as a half-sample delay.







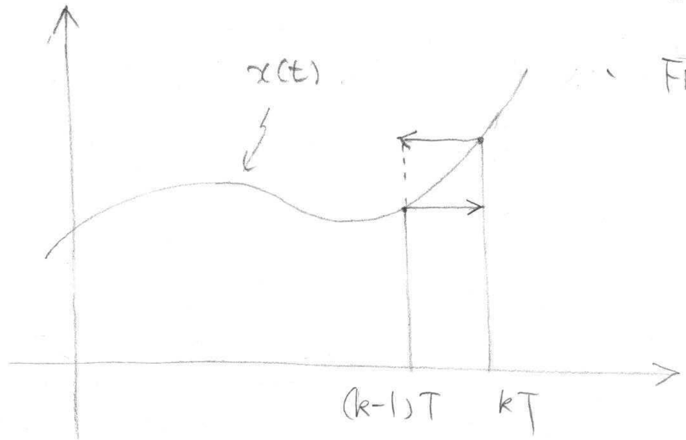
- The delay can be absorbed to the plant and we design $C(s)$ for $\underline{P(s) \cdot e^{-sT/2}}$"



- Discrete-time Approximation Methods. (a.k.a Discrete Equivalents)

① Numerical Integration.

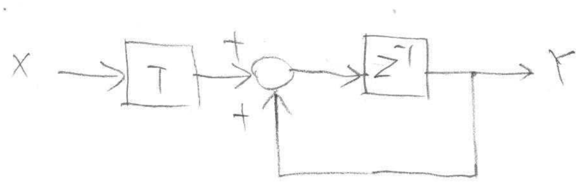
- $y(t) = \int_{-\infty}^{t} x(t)\, dt$
- Find $y[k]$ that approximates $y(t)$



- Forward rectangular method (Euler method)

$$y[k] = y[k-1] + x[k-1] \cdot T$$

$$Y(z)(1-z^{-1}) = X(z)\, z^{-1}\, T \qquad \therefore \frac{Y}{X} = T\left(\frac{z^{-1}}{1-z^{-1}}\right)$$

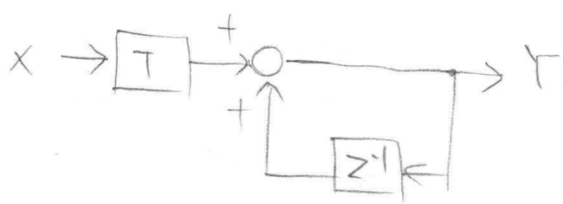


$$\boxed{s \leftarrow \frac{z-1}{T}}$$

- Backward rectangular method.

$$y[k] = y[k-1] + x[k]\, T$$

$$Y(z)(1-z^{-1}) = X(z)\, T \qquad \therefore \frac{Y}{X} = T\left(\frac{1}{1-z^{-1}}\right)$$

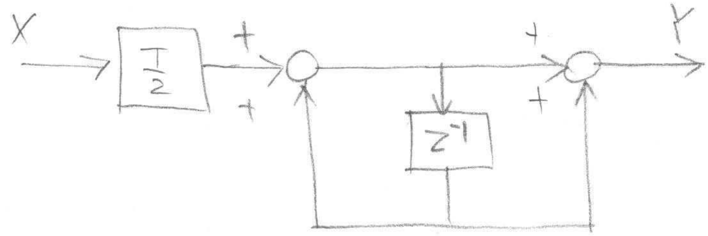


$$\boxed{s \leftarrow \frac{1}{T}\frac{z-1}{z}}$$

- Bilinear / Trapezoidal / Tustin method

- $y[k] = y[k-1] + \frac{T}{2}(x[k] + x[k-1])$

- $Y(z)(1-z^{-1}) = \frac{T}{2} X(z)(1+z^{-1}) \quad \therefore \frac{Y}{X} = \frac{T}{2} \frac{1+z^{-1}}{1-z^{-1}}$



$$\boxed{s \leftarrow \frac{2}{T} \cdot \frac{z-1}{z+1}}$$

- MATLAB command : c2d with 'tustin' option.

o Simulink / dSPACE uses the Euler metod by default, which can map a stable CT system to an unstable DT system.

< Forward Rect. >

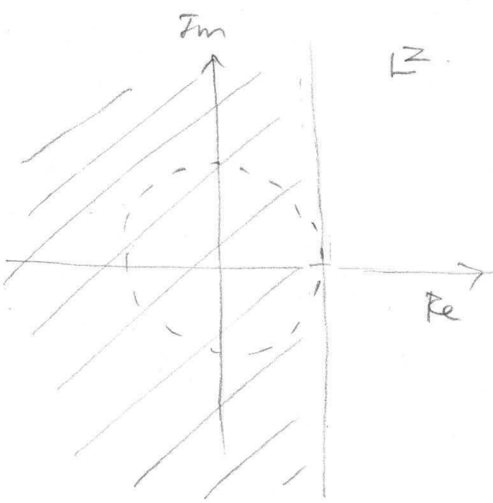
$z = 1 + Ts$

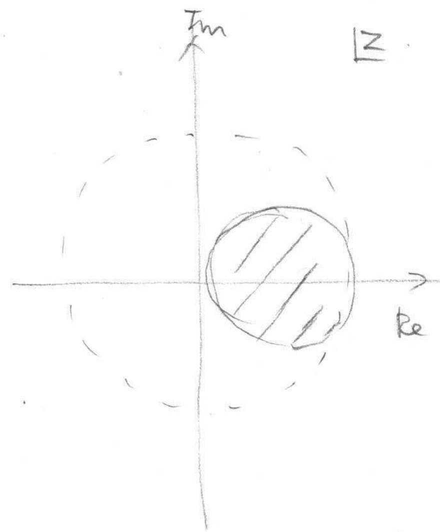
< Backward Rect. >

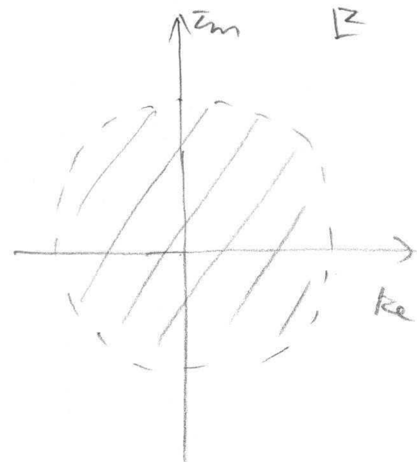
$z = \frac{1}{1-Ts}$

< Tustin. >

$z = \frac{1+Ts/2}{1-Ts/2}$







- LHP in s-domain maps to the shaded region of each z-domain.

② zero-pole matching method (FPW. Ch 6.2)

- poles & zeros of $H(s)$ $\xrightarrow{z = e^{sT}}$ poles & zeros of $H(z)$.

- MATLAB command : c2d with 'matched' option

∘ Once $C(z)$ is obtained, we can implement it into a target computer

i) Directly in forms of DT transfer function using, for example, Simulink or LabVIEW

ii) Convert it into the corresponding difference equation and implement it using text-based programming lang.