E1. Sampling of signals, aliasing, z-transform, matrix exponential

1.2, 1.3, 1.4
1.5, 1.1
V22, 3D

① key points from Lecture 1, P21-31, VT13

- locate three types of signals
  (P21, L1, VT13)
  on computer controlled system blockdiagram
  (P20, L1, VT13)

* → $y(t) \rightarrow \boxed{\text{sample}} \rightarrow y_k$: continuous-timed, continuous-valued → discrete-timed, continuous-valued

$u(t) \leftarrow \boxed{\text{hold}} \leftarrow u_k$ ←

* → $y_k(t) \rightarrow \boxed{AD} \rightarrow \bar{y}_k$: discrete-timed, continuous-valued → discrete-time, discrete-valued

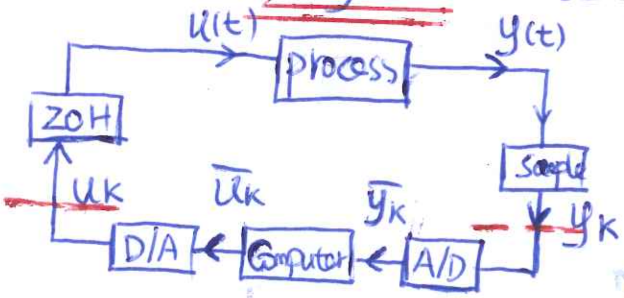
$u_k \leftarrow \boxed{DA} \leftarrow \bar{u}_k$ ←

- today's focus: sample, hold

② extra notes:

aliasing /'eiliəsiŋ/ or frequency folding

* need tools to analyze discrete system: $u_k \rightarrow y_k$



this and next session

discrete-time system

* then for design

→ intro
→ review exercise
basic concepts
→ focus on analyzing DTS

Exercise 1.1

① • why we want to solve this problem?

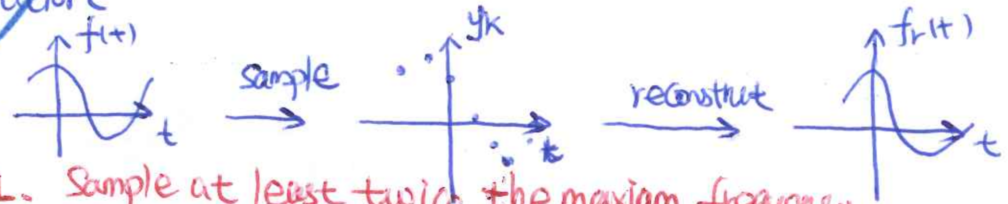
to feed $y(t)$ into a digital computer, it has to be sampled + AD, (or into a discrete-time system)

• how fast should the sampling period be? $y_k \to u_k$

not to lose information. (shannon's theorem)

② extra notes

• illustration



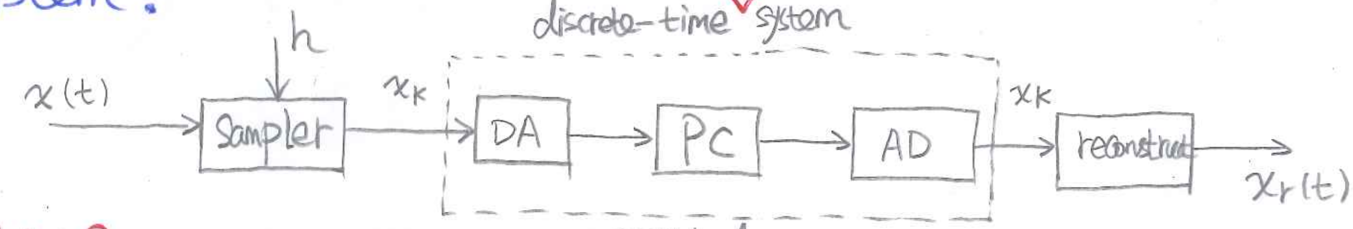
two 1. Sample at least twice the maxiam frequency
2. how to reconstruct.

• why not simply increase f_s? more control effort, more commnication, more energy

Can aliasing be removed afterwards? no, very little to do.

sol.③ later input signal: $x(t) = \cos(w_0 t)$

first system:



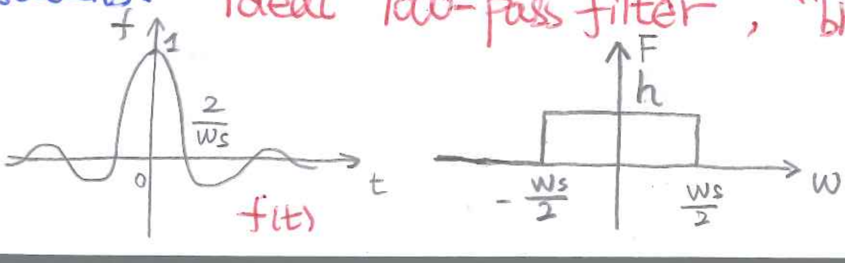
later? reconstruction filter:

$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$

time domain: $f(t) = \text{sinc}\left(\frac{w_s}{2} t\right)$, $w_s = \frac{2\pi}{h}$ (sampling frequency)

freq domain: $F(jw) = \text{Fourier-transform}(f(t)) = \begin{cases} h, & -w_s/2 < w < \frac{w_s}{2} \\ 0, & |w| > |\frac{w_s}{2}| \end{cases}$

characteristics: "ideal" low-pass filter, "brick-wall" filter



perfectly cutoff high-freq.
pass low-freq.

application:

Shannon's sampling Theorem (how to chose sampling frequency)

- give a continuous-time signal $f(t)$, which is band-limited:

$\boxed{1}$ $$\text{Fourier\_transform}(f(t)) = F(\omega) = 0, \ \forall |\omega| > |\omega_0|$$

- then sampled by a periodic $\boxed{2}$ sampler, $\omega_s = \frac{2\pi}{h}$

$$P(t) = \sum_{k=-\infty}^{\infty} \delta(t-kh)$$

- sampled signal $\boxed{3}$

$$f_k = f_s(t) = \sum_{k=-\infty}^{\infty} f(t)\cdot\delta(t-kh) = \sum_{k=-\infty}^{\infty} f(kh)\cdot\delta(t-kh)$$

$$\text{Fourier\_transform}(x_s(t)) = \frac{1}{h}\sum_{k=-\infty}^{\infty} \underbrace{F(\omega+k\omega_s)}_{\text{why?}} = F_s(\omega)$$

- reconstruction $\boxed{4}$

idea low-pass filter move to here

- ① If $\omega_s > 2\omega_0$, no aliasing

$f(t)$ can be uniquely constructed from f_k

$$f(t) = \sum_{k=-\infty}^{\infty} f_k \operatorname{sinc}\left(\frac{\omega_s(t-kh)}{2}\right)$$

- sinc function spans over infinite time domain, not applicable
- often approximated by a "windowed" sinc function
- ★ remarked-able: // only some samples are enough

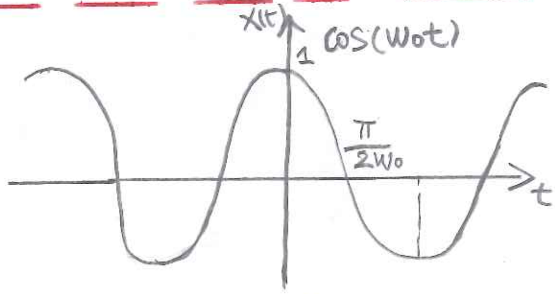
"the information content" of a bandwidth limited function is infinitely smaller than that of a general continuous function.

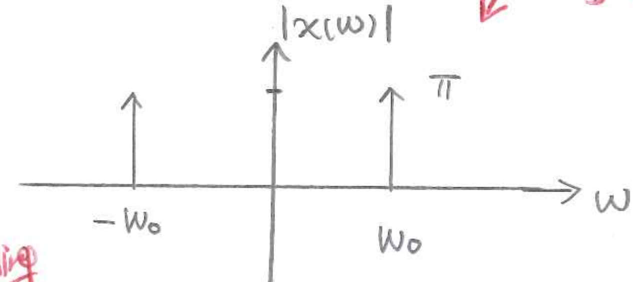
- ★ the entire information of $f(t)$ can be recorded by sampling it at twice the maximum frequency

② If $w_c < 2w_0$, or $f(t)$ is not bandwidth limited, aliasing

- high frequency part is interpreted as low frequencies
- frequency folding
- information loss

why?



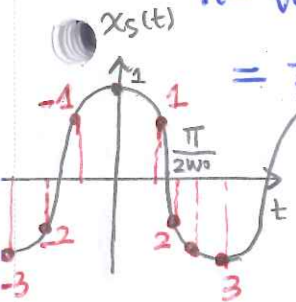


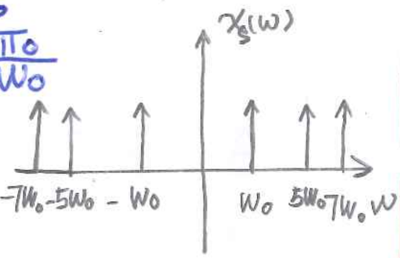
Sampling

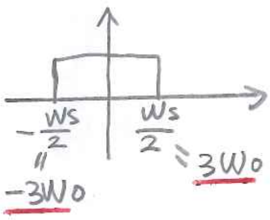
Sampling interval

$h = \frac{2\pi}{w_s} = \frac{\pi}{3w_0}$

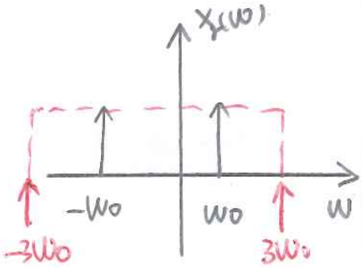
$w_s = 6w_0$







low-pass filter

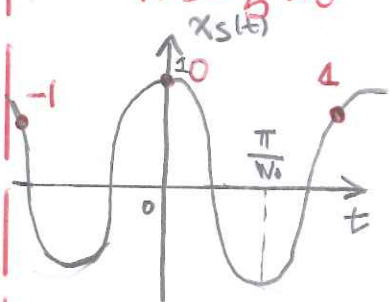


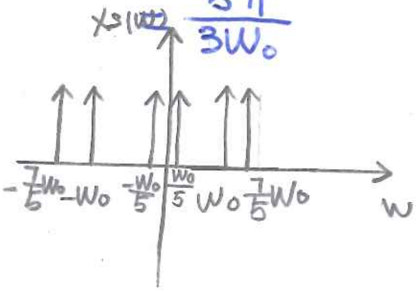
$x_r(t) = x(t) = \cos w_0 t$

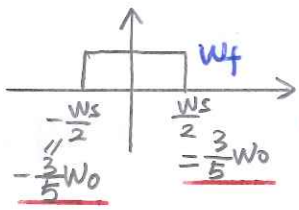
$w_s = \frac{6}{5}w_0$

Sampling interval

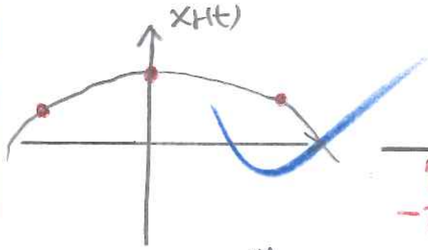
$h = \frac{2\pi}{w_s} = \frac{5\pi}{3w_0}$

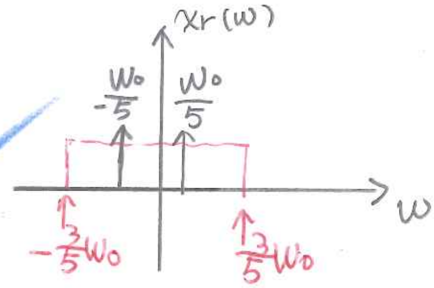






low-pass filter





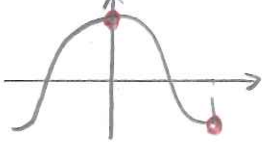
$x_r(t) = \cos(\frac{w_0}{5}t)$

- Sample at least two points per cycle.
- $x(t) = \cos w_0 t$

$f_{min} = 2 \cdot \frac{w_0}{2\pi} = \frac{w_0}{\pi} = 2f_0$

$h_{max} = \frac{1}{2f_0} = \frac{1}{2}T_0$

$w_{min} = 2w_0$



- why not increase w_f s.t. w_0 is included in $x_r(w)$?
  still can't get rid of $\pm\frac{w_0}{5}$
- to know the bandwidth of $x(t)$
  or enforce a bandwidth

Exercise 1.2

① • why we want to solve this problem?

* 'in accurate model based design of a discrete Controller for a process originally in the form of a Continuous-time model. The latter should be discretized to get a discrete-time process model (sampled) before the design is started.'
* Implementation of continuous-time control/filter-functions in computer.

• application

LTS

SS $\dot{x} = Ax + Bu$

$y = Cx + Du$

(if Periodic Sampling)

$\xrightarrow[\text{hold on input}]{\text{zero-order}}$

note $x(t) = e^{A(t-kh)}$, $t \in [kh, (k+1)h)$

$x((k+1)h) = e^{Ah} x(kh) + \left(\int_0^h e^{A\tau} d\tau\right) B u(kh)$

$y((k+1)h) = Cx(kh) + Du(kh)$

(briefly)

(aperiodic sampling)

$x(t_{k+1}) = e^{A(t_{k+1}-t_k)} x(t_k) + \left(\int_0^{t_{k+1}-t_k} e^{A\tau} d\tau\right) B u(t_k)$

This teaches how to compute e^{At} from A

② extra notes

help c2d ← check

very important to E2

sysd = c2d(sys, Ts)

scipy.signal

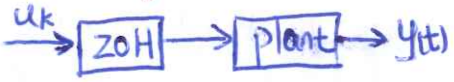
cont2discrete()

Cayley-hamilton, characteristic polynomial. $P(\lambda) = \det(\lambda I - A)$

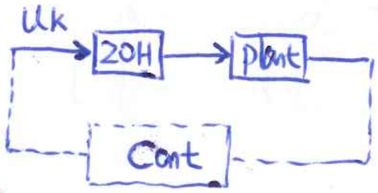
$\Rightarrow P(A) = 0$

some useful tips:

~~Usage~~ • analysis



• design



mathematic handbook allowed in exam

③ Method one: (exact calculation)

why? → $e^{Ah} = \mathcal{L}^{-1}\{(sI-A)^{-1}\}\Big|_{t=h}$

$\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}\right)$

$sI - A = \begin{bmatrix} s & -1 \\ 1 & s \end{bmatrix} \Rightarrow (sI-A)^{-1} = \frac{1}{s^2+1}\begin{bmatrix} s & 1 \\ -1 & s \end{bmatrix}$

Laplace-transform table

$\sin at \overset{\mathcal{L}}{\sim} \frac{a}{s^2+a^2}$

$\cos at \overset{\mathcal{L}}{\sim} \frac{s}{s^2+a^2}$

$\Rightarrow \mathcal{L}^{-1}\{(sI-A)^{-1}\}\Big|_{t=h} = \begin{bmatrix} \cos h & \sin h \\ -\sin h & \cos h \end{bmatrix}$

ethod three:

By Cayley–Hamilton theorem, for a matrix of order n

Any function of M can be expressed as a polynomial of degree less than n

$$f(M) = a_{n-1}M^{n-1} + a_{n-2}M^{n-2} + \cdots + a_1 M + a_0 I_n = P(M)$$

n our case,

$$e^{Ah} = a_1 Ah + a_0 I$$

then $a_0 \cdots a_{n-1}$ can be solved by

$f(\lambda_i) = P(\lambda_i)$, $i = 1 \ldots n$ (*)

where λ_i are eigenvalues of M

eigenvalue with multiplicity m.

m-th derivative of (*) should be taken

eigenvalues of Ah: $M = Ah$

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad Ah = \begin{bmatrix} 0 & h \\ -h & 0 \end{bmatrix}$$

determinant

$\lambda I - Ah = \begin{bmatrix} \lambda & -h \\ h & \lambda \end{bmatrix}$, $\det(\lambda I - Ah) = 0$ characteristic polynomial

$$\Rightarrow \lambda^2 + h^2 = 0 \Rightarrow \lambda = \pm h i$$

Note: Ah, instead of A

wo equations

$$e^{hi} = a_1 \cdot (hi) + a_0$$
$$e^{-hi} = a_1 \cdot (-hi) + a_0$$

$$\Rightarrow \cosh + \sinh \cdot i = a_1 h \cdot i + a_0$$
$$\cosh - \sinh \cdot i = -a_1 h \cdot i + a_0$$

Compare real and imaginary parts

$$\Rightarrow a_1 = \frac{\sinh}{h}, \quad a_0 = \cosh$$

Thus $e^{Ah} = \sinh \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} + \cosh \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$$= \begin{bmatrix} \cosh & \sinh \\ -\sinh & \cosh \end{bmatrix}$$

(Same as method one)

how matlab does it, good at LE

r diagonal matrix, $A = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \Rightarrow e^{Ah} = \begin{bmatrix} e^{\lambda_1 h} & & \\ & e^{\lambda_2 h} & \\ & & \ddots & \\ & & & e^{\lambda_n h} \end{bmatrix}$

final remark

Method two: approximate (numerical calculation)

$$e^{At} = I + At + \frac{A^2}{2!}t^2 + \frac{A^3}{3!}t^3 + \cdots + \frac{A^n}{n!}t^n + \cdots$$

the larger n, the more accurate it is. factorial() Taylor expension

good for high-D matrix

Exercise 1.3, 1.4

① why we want to compute Z transform? why?

- closedy related to Exercise 1.5
- much easier to compute response of discrete-time system in frequency domain
- for the purpose of analysis and design

(normally controll-inputs to transform)

---

② extra notes

- region of convergence: set of points in complex-plane, where Z-transform converges.
  ROC
  - Z-transform does not exist outside ROC
- Z-transform over discrete-time signal $\overset{\text{plays the}}{\underset{\text{same role}}{\longleftrightarrow}}$ Laplace-transform over continuous-time signal

---

③ definition of Z-transform: $y(k): \mathbb{Z}, N \to \mathbb{R}$

one-sided transform

time starts from 0

$$Y(z) = Z\{y(k)\} = \sum_{k=0}^{\infty} y(k) z^{-k}$$

geometric series $1+x+x^2+\cdots+x^n = \frac{1-x^{n-1}}{1-x}$

$(y-x^n)x = y-1$:)

$\boxed{\text{for } 1.3}$

$$X(z) = \sum_{k=0}^{\infty} e^{-kh/T} z^{-k} = \sum_{k=0}^{\infty} (e^{-\frac{h}{T}} z^{-1})^k$$

$$= \frac{1}{1-e^{-\frac{h}{T}}z^{-1}} \quad (\text{when } |e^{-\frac{h}{T}}z^{-1}| < 1) \text{ converges}$$

$\boxed{\text{for } 1.4}$

$$\sin(\omega kh) = \frac{1}{2j}(e^{j\omega kh} - e^{-j\omega kh}) \leftarrow \begin{aligned} e^{j\omega h} + e^{-j\omega h} &= 2\cos\omega h \\ e^{j\omega h} - e^{-j\omega h} &= 2j\sin\omega h \end{aligned}$$

$$\Rightarrow X(z) = \frac{1}{2j}\left[\frac{1}{1-e^{j\omega h}z^{-1}} - \frac{1}{1-e^{-j\omega h}z^{-1}}\right]$$

$$= \frac{\sin\omega h \cdot z}{z^2 - 2z\cos\omega h + 1} \quad (|e^{\pm j\omega h}z^{-1}| < 1)$$

Exercise 1.5

① why do we want to analyze the response of discrete-time system?

- to analyze the the response of continuous-time system to sampled + ZOH inputs? to ignore $x(t),\ t\in(t_k, t_{k+1})$?
- invese. for the controller design later

Time-delay property
$\mathcal{Z}\{y(k-1)\} = z^{-1}Y(z) + \cdots$
$\mathcal{Z}\{y(k-2)\} = z^{-2}Y(z) + \cdots$
see E2.9

② extra notes

- Static response. (quick in time-domain?)
- inverse transform (find a proper combination of $\mathcal{Z}$-transform pairs + $\mathcal{Z}$-transform properties)

③ unitary step: $u_0(k) = \begin{cases} 0, & k < 0 \\ 1, & k \geq 0 \end{cases}$

* the first thing comes into mind is to compute $y(k)$ iteratively in time-domain.

$y(0) = 0.5$. $y(1) = 1.25$

$k=0$, $y(2) = 1.5y(1) - 0.5y(0) + u(1) = 1.4$

$k=1$, $y(3) = 1.5y(2) - 0.5y(1) + u(2) = 2.475$

$\vdots$

quick for few points, hard to derive closed-form expression

* rely on $\mathcal{Z}$-transform

$\boxed{1.}$ Time-~~delay~~ advancing property of $\mathcal{Z}$-transform: $Y(z) = \mathcal{Z}\{y(k)\}$

$$\mathcal{Z}\{y(k+1)\} = z\cdot Y(z) - z\cdot y(0)$$ also Linear Property

$$\mathcal{Z}\{y(k+2)\} = z^2Y(z) - z^2 y(0) - z\,y(1)$$

$\boxed{2.}$ $\mathcal{Z}$-transform at both side of difference equation

$$[z^2Y(z) - z^2\underset{0.5}{y(0)} - z\underset{1.25}{y(1)}] - [1.5zY(z) - 1.5z\underset{0.5}{y(0)}] + 0.5Y(z) = zU(z) - z\underset{1}{u(0)}$$

$$\Rightarrow Y(z) = \frac{0.5z^2 - 0.5z}{z^2 - 1.5z + 0.5} + \frac{z}{z^2 - 1.5z + 0.5}U(z) = \frac{0.5z}{z-0.5} + \frac{z\cdot U(z)}{(z-1)(z-0.5)}$$

$\boxed{3.}$ unitary step $u_0(k)$, $U(z) = \frac{z}{z-1}$

$$\Rightarrow Y(z) = \frac{0.5z}{z-0.5} + \frac{z^2}{(z-1)^2(z-0.5)}$$

$$= 0.5 \cdot \underbrace{\frac{1}{1-0.5z^{-1}}}_{1} + \left[\underbrace{\frac{2z^{-1}}{(1-z^{-1})^2}}_{2} + \underbrace{\frac{1}{1-0.5z^{-1}}}_{1}\right] \underbrace{z^{-1}}_{3}$$

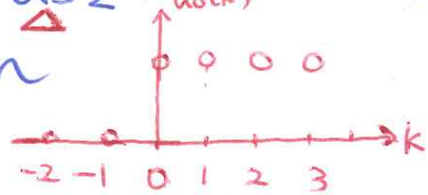
$\boxed{4.}$ table of z-transform pairs (inverse-transform to go back to time domain)

- $k u_0(k) \overset{z}{\rightleftharpoons} \frac{z^{-1}}{(1-z^{-1})^2}$ $(|z|>1)$ 1
- $a^k u_0(k) \overset{z}{\rightleftharpoons} \frac{1}{1-az^{-1}}$ $(|z|>a)$ 2

$$\Rightarrow \frac{2z^{-1}}{(1-z^{-1})^2} \overset{z^{-1}}{\rightleftharpoons} 2k\, u_0(k) \quad \Big| \quad \frac{1}{1-0.5z^{-1}} \underset{u_0(k)}{\overset{z^{-1}}{\rightleftharpoons}} 0.5^k u_0(k)$$

- time-delay property 3 of z-transform

$\triangle$ $Y(z)z^{-1} \overset{z^{-1}}{\rightleftharpoons} y(k-1)$



$$\Rightarrow y(k) = 0.5 \cdot 0.5^k u_0(k) + [(2k-2) + 0.5^{k-1}]\, u_0(k-1)$$

$\boxed{5.}$ confirmation $k=0$, $y(0) = 0.5 + 0 = 0.5$ ✓

$k=1$, $y(1) = 0.25 + 1 = 1.25$ ✓

* 'z-transform provides an systematic way to compute closed-form expression of a discrete-time system'

- Extensions:
  - transfer function from u to y $\quad u = \delta[n] = \begin{cases} 1, & n=0 \\ 0, & n\neq 0 \end{cases}$

    ‖

    z-transformed impulse response $\quad U(z) = 1$
  - z-transfer function can be written both with positive exponent or negative exponent of z.

    Control theory =)

    Signal processing