Discrete Systems Analysis
(From Digital Signals to Systems)

ELEC 3004: **Digital Linear Systems:** Signals & Controls

Dr. Surya Singh

Lecture 8

elec3004@itee.uq.edu.au
http://robotics.itee.uq.edu.au/~elec3004/

April 27, 2015

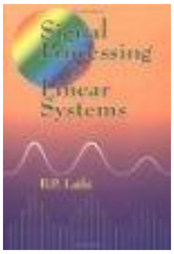
http://elec3004.org

Schedule

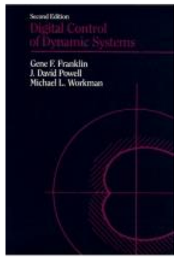
1	2-Mar	Introduction
	3-Mar	Systems Overview
2	9-Mar	Signals as Vectors & Systems as Maps
	10-Mar	[Signals]
3	16-Mar	Sampling & Data Acquisition & Antialiasing Filters
	17-Mar	[Sampling]
4	23-Mar	System Analysis & Convolution
	24-Mar	[Convolution & FT]
5	30-Mar	Discrete Systems & Z-Transforms
	31-Mar	[Z-Transforms]
6	13-Apr	Frequency Response & Filter Analysis
	14-Apr	[Filters]
7	20-Apr	Digital Filters
	21-Apr	[Digital Filters]
8	27-Apr	Discrete Systems Analysis
	28-Apr	[Feedback]
9	4-May	Introduction to (Digital) Control
	5-May	[Digitial Control]
10	11-May	Digital Control Design & State-Space
	12-May	Controllability & Observability
11	18-May	Stability of Digital Systems
	19-May	[Stability]
12	25-May	Applications in Industry
	26-May	Digitial Control System Hardware
13	1-Jun	System Identification & Information Theory + Communications
	2-Jun	Summary and Course Review

Outline:
(1) FIR< Special case of DTFT < Special case of Z-Transform
(2) Signals Recap
(3) Discrete Systems Analysis ($s \leftrightarrow \mathbf{z}$)

Additional Reading

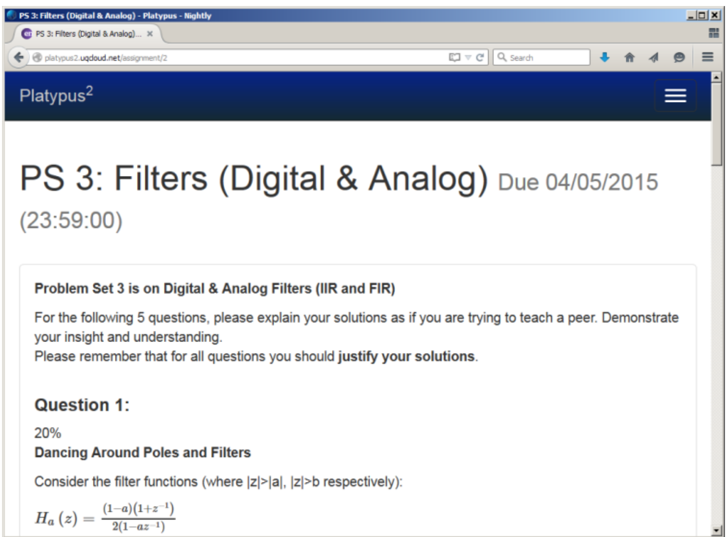


- Z-Transform – Chapter 11
- Digital Filters – Chapter 12 (pp. 723-726)
- p. 543: **s** → **z** : (λ → γ planes)
- Feedback & Control Equations (§ 6.7, p. 426)



- Chapters: 2-6, 8
- §2.5 Signal Analysis and Dynamic Response
- §5.2 Control System Specifications

http://p2.elec3004.org



2D DFT

$$\mathcal{F}(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux+vy)/N}$$

$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \mathcal{F}(u,v)e^{j2\pi(ux+vy)/N}$$

2D DFT

- Each DFT coefficient is a complex value
  - There is a single DFT coefficient for each spatial sample
  - A complex value is expressed by two real values in either Cartesian or polar coordinate space.
    - Cartesian: R(u,v) is the *real* and I(u, v) the *imaginary* component
    - Polar: |F(u,v)| is the *magnitude* and phi(u,v) the *phase*

$$\mathcal{F}(u,v) = R(u,v) + jI(u,v)$$

$$\mathcal{F}(u,v) = |F(u,v)|e^{j\phi(u,v)}$$

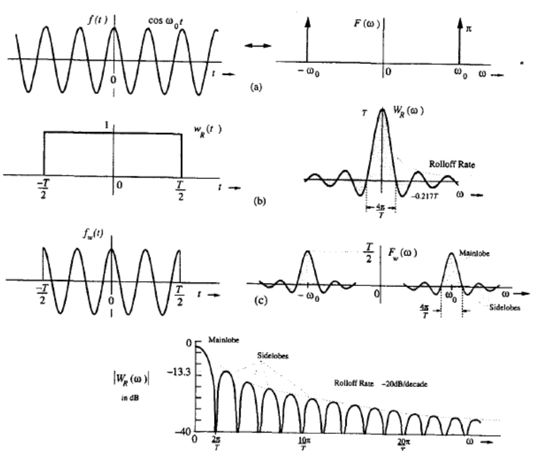
2D DFT

- Representing the DFT coefficients as magnitude and phase is a more useful for processing and reasoning.
  - The magnitude is a measure of strength or length
  - The phase is a direction and lies in [-pi, +pi]
- The magnitude and phase are easily obtained from the real and imaginary values

$$|\mathcal{F}(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$$
$$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right].$$

Windowing for the DFT

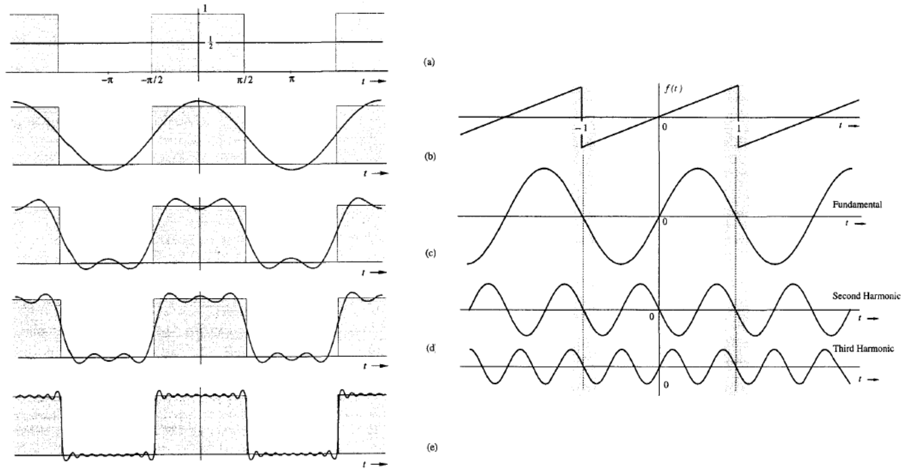
$$f_w(t) = f(t)w(t) \quad \text{and} \quad F_w(\omega) = \frac{1}{2\pi}F(\omega) * W(\omega)$$



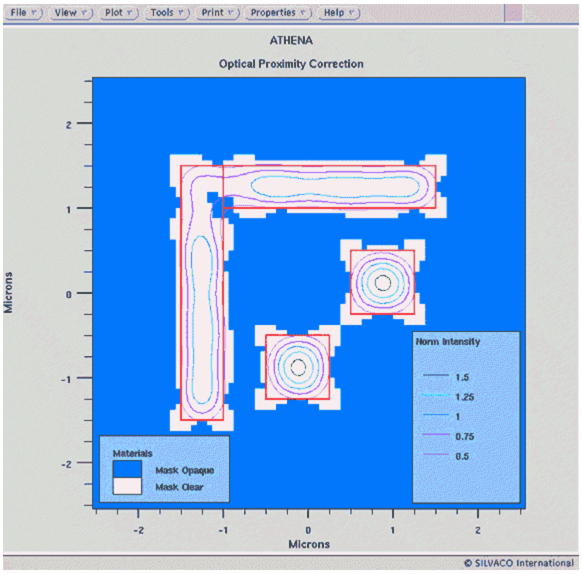
Source: Lathi, p.303

Harmonics

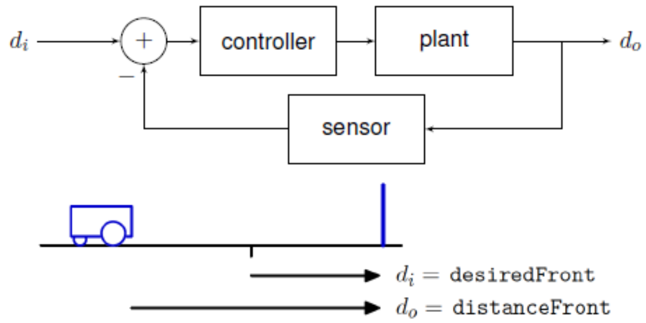
- Synthesis of a square pulse: periodic signal by successive addition of its harmonics (Lathi, p. 202-3)



Optical Proximity Correction



Simple Controller Goes Digital

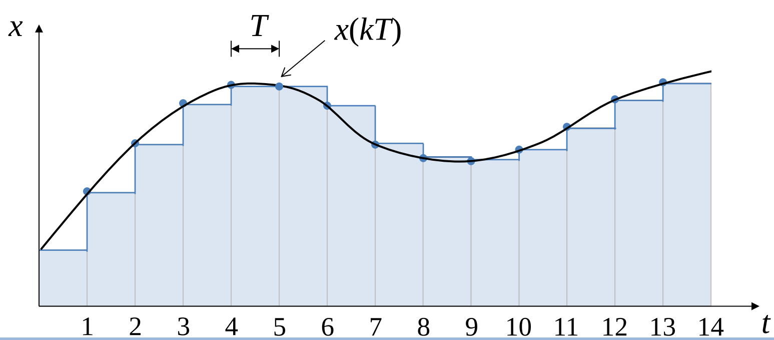


plant: $y[n] = y[n-1] - Tu[n-1]$
sensor: $y[n] = u[n-1]$
controller: $y[n] = Ku[n]$

Complex system behaviors, depending on K

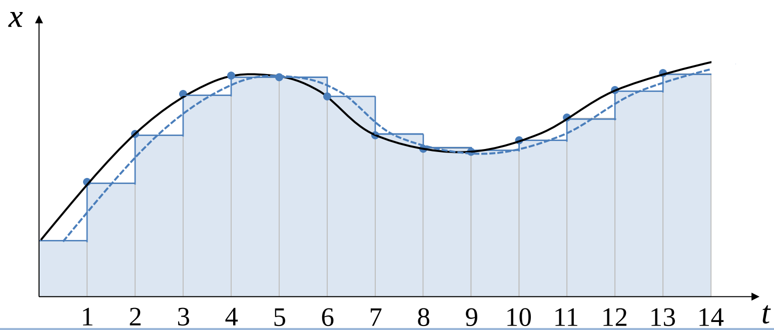
Return to the discrete domain

- Recall that continuous signals can be represented by a series of samples with period T



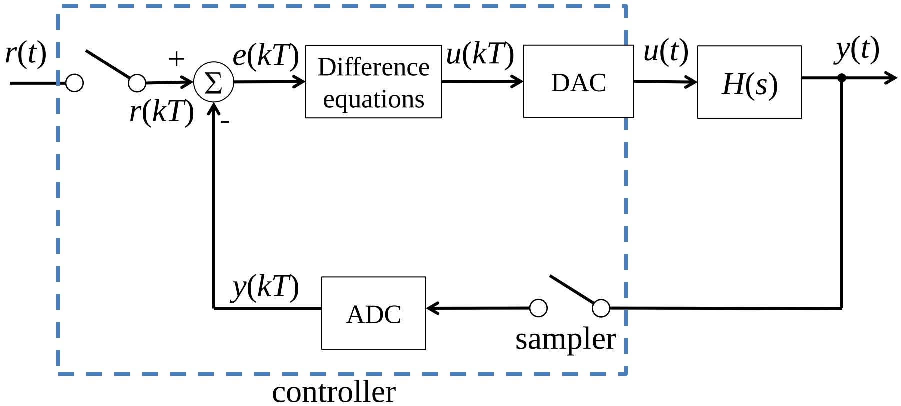
Zero Order Hold

- An output value of a synthesised signal is held constant until the next value is ready
  - This introduces an effective delay of $T/2$



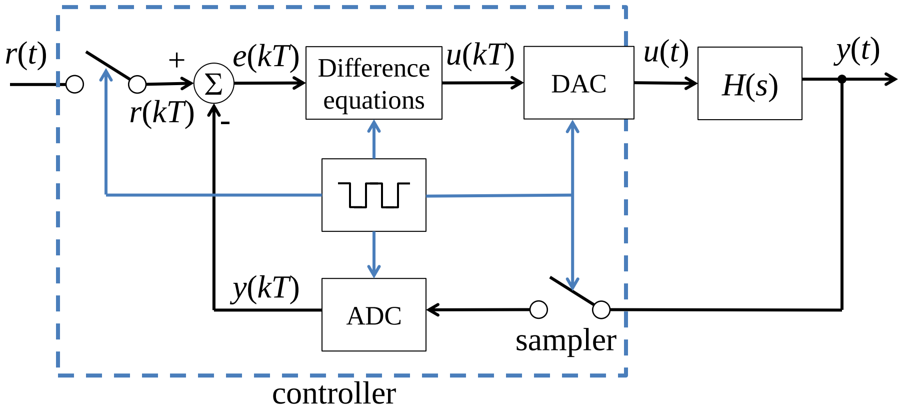
Digitisation

- Continuous signals sampled with period T
- kth control value computed at $t_k = kT$

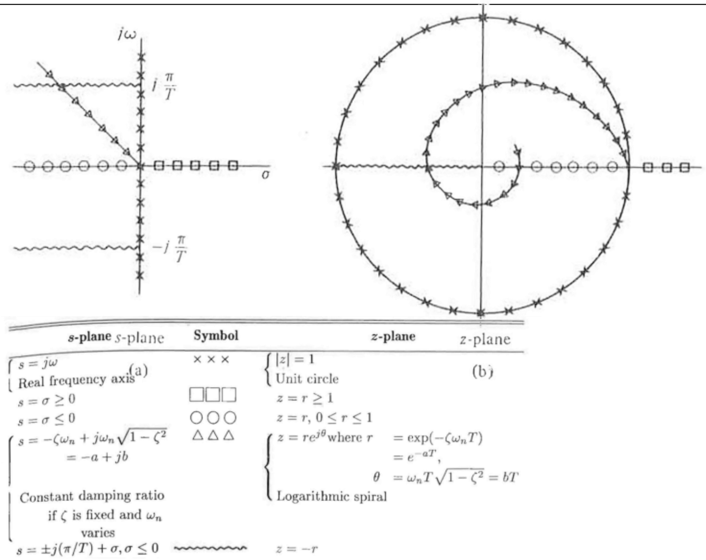


Digitisation

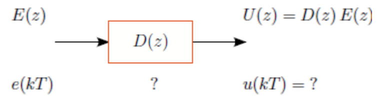
- Continuous signals sampled with period T
- kth control value computed at $t_k = kT$



$s \leftrightarrow z$: $z=e^{sT}$

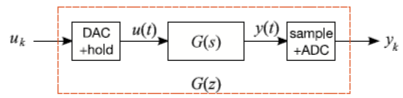


$s \leftrightarrow z$: Pulse Transfer Function Models



- Pulse in Discrete is equivalent to Dirac-δ

$$e_k = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k > 0 \end{cases}$$



➔

$$G(z) = (1 - z^{-1})\mathcal{Z}\left\{\mathcal{L}^{-1}\left\{\frac{G(s)}{s}\right\}_{t=kT}\right\} = \boldsymbol{(1 - z^{-1})\mathcal{Z}\left\{\frac{G(s)}{s}\right\}}$$

Source: Oxford 2A2 Discrete Systems, Tutorial Notes p. 26

Difference equations

- How to represent differential equations in a computer? Difference equations!

- The output of a difference equation system is a function of current and previous values of the input and output:

$$y(t_k) = D(x(t_k), x(t_{k-1}), \dots, x(t_{k-n}), y(t_{k-1}), \dots, y(t_{k-n}))$$

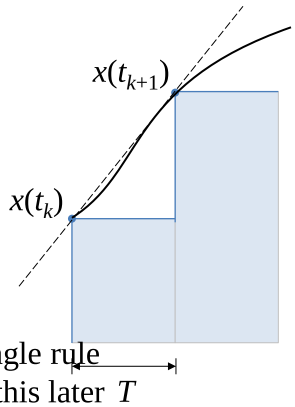
- We can think of x and y as parameterised in k
  - Useful shorthand: $x(t_{k+i}) \equiv x(k+i)$

Euler's method*

- Dynamic systems can be approximated† by recognising that:

$$\dot{x} \cong \frac{x(k+1) - x(k)}{T}$$

- As $T \to 0$, approximation error approaches 0



*Also known as the forward rectangle rule
†Just an approximation – more on this later

An example!

Convert the system $\frac{Y(s)}{X(s)} = \frac{s+2}{s+1}$ into a difference equation with period T, using Euler's method.

1. Rewrite the function as a dynamic system:

$$sY(s) + Y(s) = sX(s) + 2X(s)$$

Apply inverse Laplace transform:

$$\dot{y}(t) + y(t) = \dot{x}(t) + 2x(t)$$

2. Replace continuous signals with their sampled domain equivalents, and differentials with the approximating function

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

An example!

Simplify:

$$y(k+1) - y(k) + Ty(k) = x(k+1) - x(k) + 2Tx(k)$$
$$y(k+1) + (T-1)y(k) = x(k+1) + (2T-1)x(k)$$

$$y(k+1) = x(k+1) + (2T-1)x(k) - (T-1)y(k)$$

We can implement this in a computer.

Cool, let's try it!

Back to the future

A quick note on causality:

- Calculating the "(k+1)th" value of a signal using

$$y(k+1) = \underbrace{x(k+1)}_{\text{future value}} + \underbrace{Ax(k) - By(k)}_{\text{current values}}$$

relies on also knowing the next (future) value of $x(t)$.

(this requires very advanced technology!)

- Real systems always run with a delay:

$$y(k) = x(k) + Ax(k-1) - By(k-1)$$

Back to the example!

```
T = 0.02;  //period of 50 Hz, a number pulled from thin air
A = 2*T-1; //pre-calculated control constants
B = T-1;

...

while(1)
   {
      if(interrupt_flag)          //this triggers every 20 ms
      {
         x0 = x;                  //save previous values
         y0 = y;
         x = update_input();      //get latest x value
         y = x + A*x0 - B*y0;     //do the difference equations
         update_output(y);        //write out current value
      }
   }
```

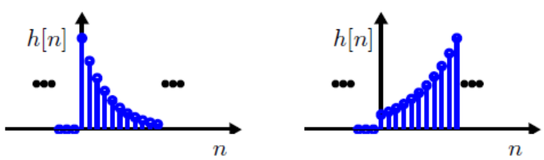
(The actual maths bit)

z-Transforms for Difference Equations

- First-order linear constant coefficient difference equation:

First-order linear constant coefficient difference equation:

$$y[n] = ay[n-1] + bu[n]$$



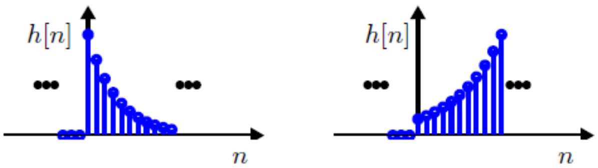
$$h[n] = \begin{cases} ba^n & n \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

$$H(z) = \sum_{k=0}^{\infty} ba^k z^{-k} = b \sum_{k=0}^{\infty} \left(\frac{a}{z}\right)^k = \frac{b}{1 - az^{-1}}, \quad \text{when } |z| > |a|.$$

z-Transforms for Difference Equations

First-order linear constant coefficient difference equation:

$$y[n] = ay[n-1] + bu[n]$$

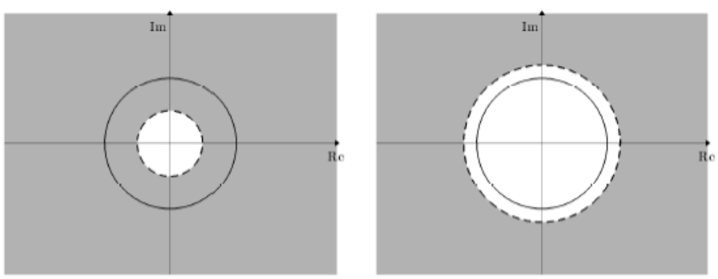


$$y[n] - ay[n-1] = bu[n]$$
$$\updownarrow$$
$$Y(z) - az^{-1}Y(z) = bU(z)$$

$H(z) = \dfrac{Y(z)}{U(z)} = \dfrac{b}{1 - az^{-1}}$, when does it converge?

Region of Convergence (ROC) Plots

$$H(z) = \frac{Y(z)}{U(z)} = \frac{b}{1 - az^{-1}}, \quad |z| > |a|$$



$a = .5$ $a = 1.2$

Properties of the ROC

➔The ROC is always defined by circles centered around the origin.

$h[k]r^{-k}$ is absolutely summable, where $r = |z|$.

➔Right-sided signals have “outsided” ROCs.

if $\exists n_0$ such that $h[n] = 0 \; \forall n < n_0$, then if $r_0 \in$ ROC, then $\forall r$ with $r_0 < r < \infty$ are also in the ROC.

➔Left-sided signals have “insided” ROCs. (with $\forall$r within $0<r<r_0$)

Properties of the the z-transform

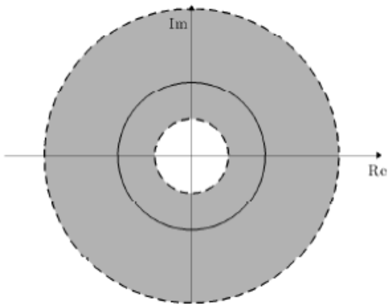
- Some useful properties
  - **Delay by n samples**: $\mathcal{Z}\{f(k-n)\} = z^{-n}F(z)$
  - **Linear**: $\mathcal{Z}\{af(k) + bg(k)\} = \mathrm{a}F(z) + bG(z)$
  - **Convolution**: $\mathcal{Z}\{f(k) * g(k)\} = F(z)G(z)$

So, all those block diagram manipulation tools you know and love will work just the same!

Combinations of Signals

$$y_1[n] = \begin{cases} ba^n & n \geq 0 \\ 0 & n < 0 \end{cases} \qquad y_2[n] = \begin{cases} 0 & n \geq 0 \\ -ba^n & n < 0 \end{cases}$$

$a = .5$ $\qquad$ $a = 2$



ROC for $\alpha_1 y_1[n] + \alpha_2 y_2[n]$

Higher-order difference equations

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + a_3 y[n-3] + b_0 u[n] + b_1 u[n-1] + \dots$$

Easy to take the Z-transform

$$Y(z) = a_1 z^{-1} Y(z) + a_2 z^{-2} Y(z) + a_3 z^{-3} Y(z) + b_0 U(z) + \dots$$

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-1} + \dots}{1 - a_1 z^{-1} - a_2 z^{-2} - a_3 z^{-3} + \dots}$$

Final value theorem

- An important question: what is the steady-state output a stable system at $t = \infty$?
  - For continuous systems, this is found by:

$$\lim_{t\to\infty} x(t) = \lim_{s\to 0} sX(s)$$

  - The discrete equivalent is:

$$\lim_{k\to\infty} x(k) = \lim_{z\to 1} (1 - z^{-1})X(z)$$

(Provided the system is stable)

An example!

- Back to our difference equation:

$$y(k) = x(k) + Ax(k-1) - By(k-1)$$

becomes

$$Y(z) = X(z) + Az^{-1}X(z) - Bz^{-1}Y(z)$$
$$(z + B)Y(z) = (z + A)X(z)$$

which yields the transfer function:

$$\frac{Y(z)}{X(z)} = \frac{z + A}{z + B}$$

Note: It is also not uncommon to see systems expressed as polynomials in z^{-n}

This looks familiar…

- Compare:

$$\frac{Y(s)}{X(s)} = \frac{s+2}{s+1} \text{ vs } \frac{Y(z)}{X(z)} = \frac{z+A}{z+B}$$

How are the Laplace and *z* domain representations related?

Consider the simplest system

- Take a first-order response:

$$f(t) = e^{-at} \Rightarrow \mathcal{L}\{f(t)\} = \frac{1}{s+a}$$

- The discrete version is:

$$f(kT) = e^{-akT} \Rightarrow \mathcal{Z}\{f(k)\} = \frac{z}{z - e^{-aT}}$$

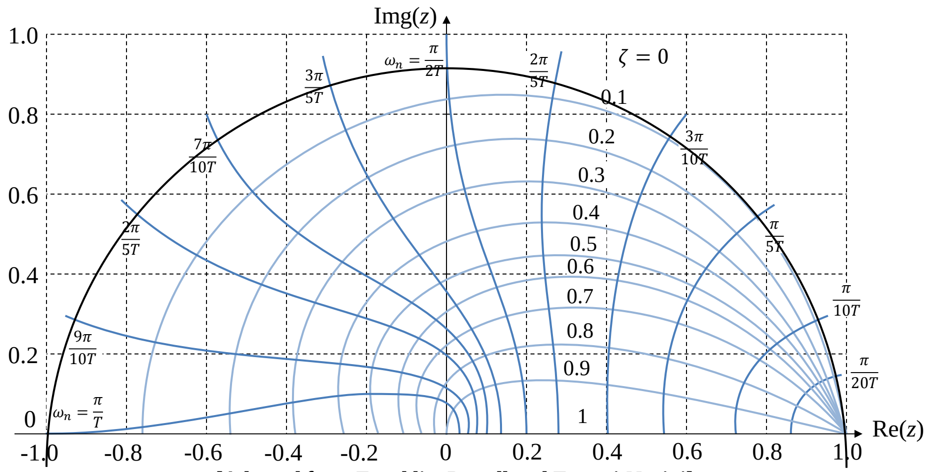
The equivalent system poles are related by

$$z = e^{sT}$$

That sounds somewhat profound… but what does it mean?

Damping and natural frequency

$$z = e^{sT} \text{ where } s = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$$



[Adapted from Franklin, Powell and Emami-Naeini]