http://elec3004.com

$\mathcal{Z}$-Transforms (Take 2)
Linear Time Invariant Discrete Systems

ELEC 3004: **Systems**: Signals & Controls
Dr. Surya Singh

Lecture 8

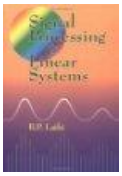
elec3004@itee.uq.edu.au
http://robotics.itee.uq.edu.au/~elec3004/

March 22, 2019

Lecture Schedule:

Week	Date	Lecture Title
1	27-Feb	Introduction
	1-Mar	Systems Overview
2	6-Mar	Systems as Maps & Signals as Vectors
	8-Mar	Systems: Linear Differential Systems
3	13-Mar	Sampling Theory & Data Acquisition
	15-Mar	Aliasing & Antialiasing
4	20-Mar	Discrete Time Analysis & Z-Transform
	22-Mar	**Second Order LTID (& Convolution Review)**
5	27-Mar	Frequency Response
	29-Mar	Filter Analysis
6	3-Apr	Digital Filters (IIR) & Filter Analysis
	5-Apr	Digital Filter (FIR)
7	10-Apr	Digital Windows
	12-Apr	FFT
8	17-Apr	Active Filters & Estimation & Holiday
	19-Apr	**Holiday**
	24-Apr	
	26-Apr	
9	1-May	Introduction to Feedback Control
	3-May	Servoregulation/PID
10	8-May	PID & State-Space
	10-May	State-Space Control
11	15-May	Digital Control Design
	17-May	Stability
12	22-May	State Space Control System Design
	24-May	Shaping the Dynamic Response
13	29-May	System Identification & Information Theory
	31-May	Summary and Course Review

Follow Along Reading:



B. P. Lathi
Signal processing and linear systems
1998
TK5102.9.L38 1998

Today

- Chapter 11 (**Discrete-Time System Analysis Using the *z*-Transform**)
  - § 11.1 The $\mathcal{Z}$-Transform
  - § 11.2 Some Properties of the *Z*-Transform

- Chapter 9 (**Time-Domain Analysis of Discrete-Time Systems**)
  - § 9.4 System Response to External Input
  - § 9.6 System Stability

Next Time

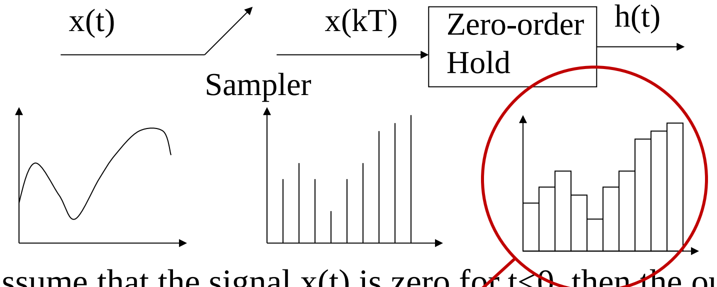
---

z Transforms

(Digital Systems Made eZ)

Extended Explanation ☺

Back to the Zero-order **Hold** (ZOH)


x(t)

x(kT)

Zero-order Hold

h(t)

Sampler

- Assume that the signal x(t) is zero for t<0, then the output h(t) is related to x(t) as follows:

$$h(t) = x(0)[1(t) - 1(t-T)] + x(T)[1(t-T) - 1(t-2T)] + \cdots$$
$$= \sum_{k=0}^{\infty} x(kT)[1(t - kT) - 1(t - (k+1)T)]$$

➔The “hold” adds a delay. The delay leads to difference equations

---

Coping with Complexity

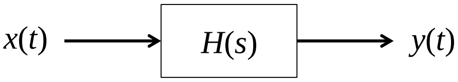
Transfer functions help control complexity

  – Recall the Laplace transform:

$$\mathcal{L}\{f(t)\} = \int_0^{\infty} f(t)e^{-st}dt = F(s)$$

where

$$\mathcal{L}\{\dot{f}(t)\} = sF(s)$$


$x(t)$ → $H(s)$ → $y(t)$

- Is there a something similar for sampled systems?

The z-Transform

- It is defined by:

$$z = re^{j\omega}$$

- Or in the Laplace domain:

$$z = e^{sT}$$

- Thus: $Y(z) = \sum_{k=-\infty}^{\infty} h[k]z^{-k}$ $y[n] \overset{\mathcal{Z}}{\longleftrightarrow} Y(z)$

- **That is → it is a discrete version of the Laplace:**

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

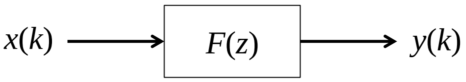
The z-transform

- The discrete equivalent is the *z*-Transform†:

$$\mathcal{Z}\{f(k)\} = \sum_{k=0}^{\infty} f(k)z^{-k} = F(z)$$

and

$$\mathcal{Z}\{f(k-1)\} = z^{-1}F(z)$$



Convenient!

†This is not an approximation, but approximations are easier to derive

The z-Transform [2]

- Thus:

$$Y(z) = \sum_{k=-\infty}^{\infty} h[k] z^{-k} \qquad\qquad y[n] \overset{\mathcal{Z}}{\longleftrightarrow} Y(z)$$

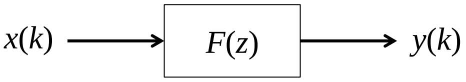
- z-Transform is analogous to other transforms:

$$\mathcal{Z}\{f(k)\} = \sum_{k=0}^{\infty} f(k) z^{-k} = F(z)$$

and

$$\boxed{\mathcal{Z}\{f(k-1)\} = z^{-1}F(z)}$$

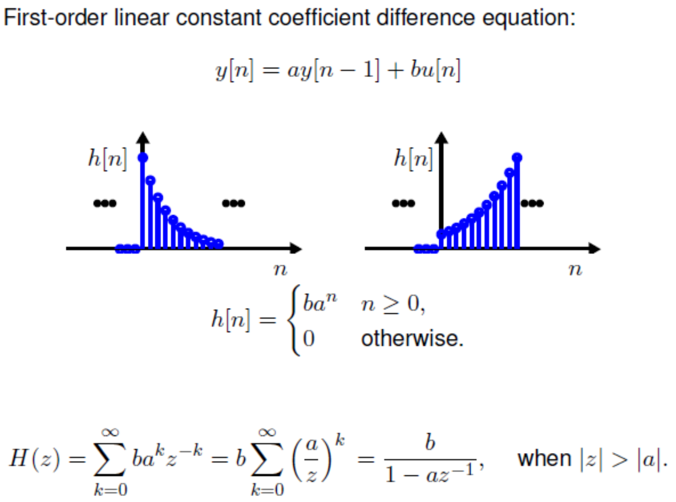
∴ Giving:



---

z-Transforms for Difference Equations

- First-order linear constant coefficient difference equation:



First-order linear constant coefficient difference equation:

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

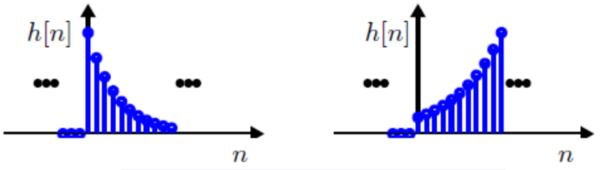
$$h[n] = \begin{cases} ba^n & n \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

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

z-Transforms for Difference Equations

First-order linear constant coefficient difference equation:

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



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

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

The z-transform

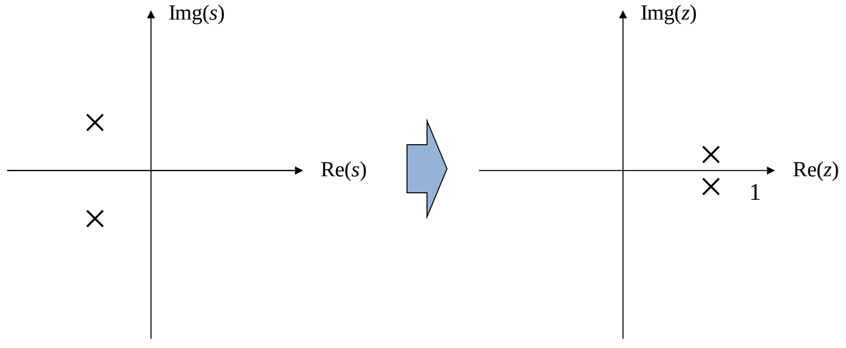
- In practice, you'll use look-up tables or computer tools (ie. Matlab) to find the *z*-transform of your functions

$\boldsymbol{F(s)}$	$\boldsymbol{F(kt)}$	$\boldsymbol{F(z)}$
$\frac{1}{s}$	1	$\frac{z}{z-1}$
$\frac{1}{s^2}$	kT	$\frac{Tz}{(z-1)^2}$
$\frac{1}{s+a}$	e^{-akT}	$\frac{z}{z-e^{-aT}}$
$\frac{1}{(s+a)^2}$	kTe^{-akT}	$\frac{zTe^{-aT}}{(z-e^{-aT})^2}$
$\frac{1}{s^2+a^2}$	$\sin(akT)$	$\frac{z\sin aT}{z^2-(2\cos aT)z+1}$

The z-Plane

z-domain poles and zeros can be plotted just like *s*-domain poles and zeros (of the $\mathcal{L}$):

- S-plane:
- $\boldsymbol{z = e^{sT}}$ Plane

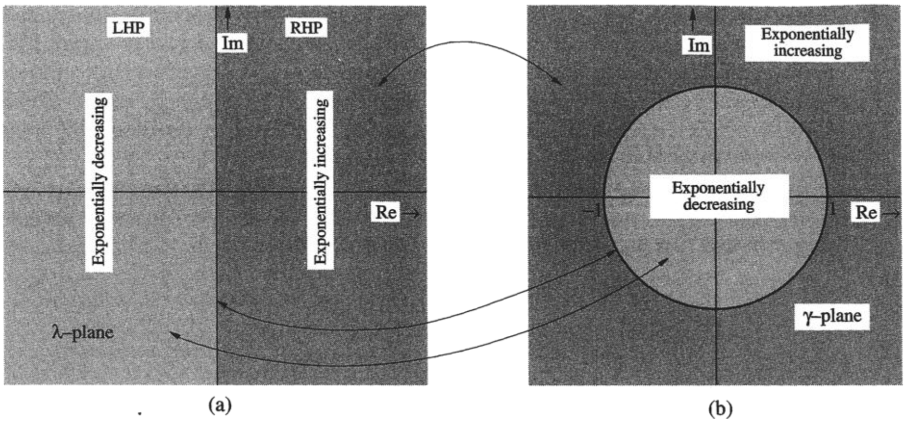


  - λ – Plane
  - γ – Plane

Discrete-Time Exponential γ^k

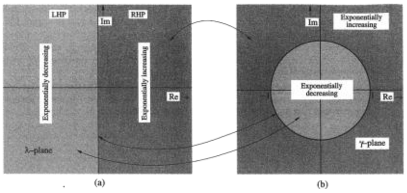
Recall we defined $z = e^{sT}$
Thus, this is the γ plane ➔

$$e^{\lambda k} = \gamma^k$$



(a) (b)

Discrete-Time Exponential γ^k

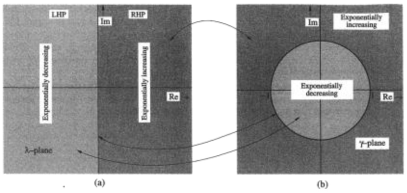


- $e^{\lambda k} = \gamma^k$
- $\gamma = e^{\lambda}$ or $\lambda = \ln \gamma$
- In discrete-time systems, unlike the continuous-time case, the form γ^k proves more convenient than the form $e^{\lambda k}$

Why?

- Consider $e^{j\Omega k}$ ($\lambda = j\Omega \therefore$ constant amplitude oscillatory)
- $e^{j\Omega k}$ ➔ γ^k, for $\gamma \equiv e^{j\Omega}$
- $\left|e^{j\Omega}\right| = 1$, hence $|\gamma| = 1$

Discrete-Time Exponential γ^k



- Consider $e^{\lambda k}$
  When λ: LHP
- Then
- $\gamma = e^{\lambda}$
- $\gamma = e^{\lambda} = e^{a+jb} = e^{a}e^{jb}$
- $|\gamma| = \left|e^{a}e^{jb}\right| = |e^{a}| \because \left|e^{jb}\right| = 1$

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!

More Z-Transform Properties

- Time Reversal

$$x[n] \leftrightarrow X(z) \qquad \mathrm{ROC} = R$$

$$x[-n] \leftrightarrow X\left(\frac{1}{z}\right) \qquad R' = \frac{1}{R}$$

- Multiplication by z^n

$$x[n] \leftrightarrow X(z) \qquad \mathrm{ROC} = R$$

$$z_0^n x[n] \leftrightarrow X\left(\frac{z}{z_0}\right) \qquad R' = |z_0|R$$

- Multiplication by n (or Differentiation in z):

$$x[n] \leftrightarrow X(z) \qquad \mathrm{ROC} = R$$

$$nx[n] \leftrightarrow -z\frac{dX(z)}{dz} \qquad R' = R$$

- Convolution

$$x_1[n] \leftrightarrow X_1(z) \qquad \mathrm{ROC} = R_1$$
$$x_2[n] \leftrightarrow X_2(z) \qquad \mathrm{ROC} = R_2$$

$$x_1[n] * x_2[n] \leftrightarrow X_1(z)X_2(z) \qquad R' \supset R_1 \cap R_2$$

Z-Transform Properties: Time Shifting

$$y[n-n_0] \xleftrightarrow{\mathcal{Z}} z^{-n_0}Y(z)$$

$$\begin{aligned} y_2[n] &= y[n-n_0] \\ Y_2(e^{jw}) &= \sum_{k=-\infty}^{\infty} y[k-n_0]z^{-k} \\ &= \sum_{l=-\infty}^{\infty} y[l]z^{-(l+n_0)} \\ &= z^{-n_0}Y(z) \end{aligned}$$

- Two Special Cases:
- z^{-1}: the *unit-delay operator:*

$$x[n-1] \leftrightarrow z^{-1}X(z) \qquad R' = R \cap \{0 < |z|\}$$

- z: *unit-advance operator:*

$$x[n+1] \leftrightarrow zX(z) \qquad R' = R \cap \{|z| < \infty\}$$

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}

BREAK

LTI(D) Systems Properties

System Stability

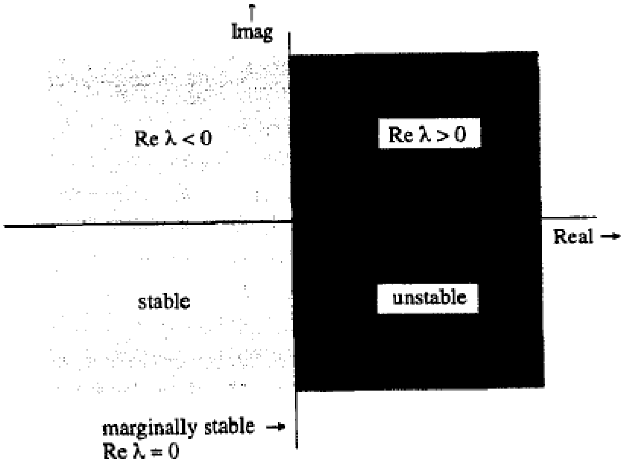
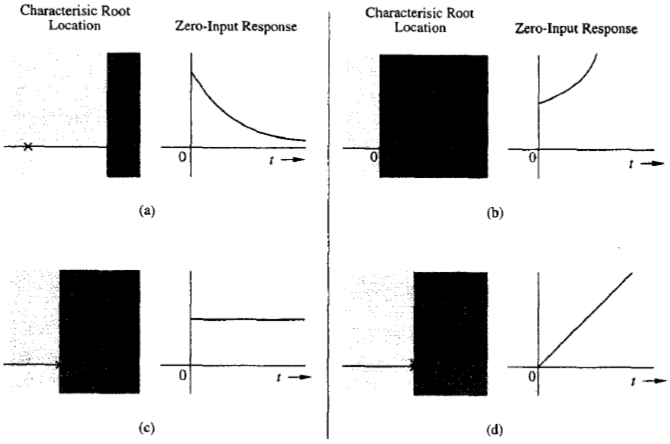


Fig. 2.15 Characteristic roots location and system stability.

Lathi, p. 149

System Stability [II]



Lathi, p. 150

System Stability [III]

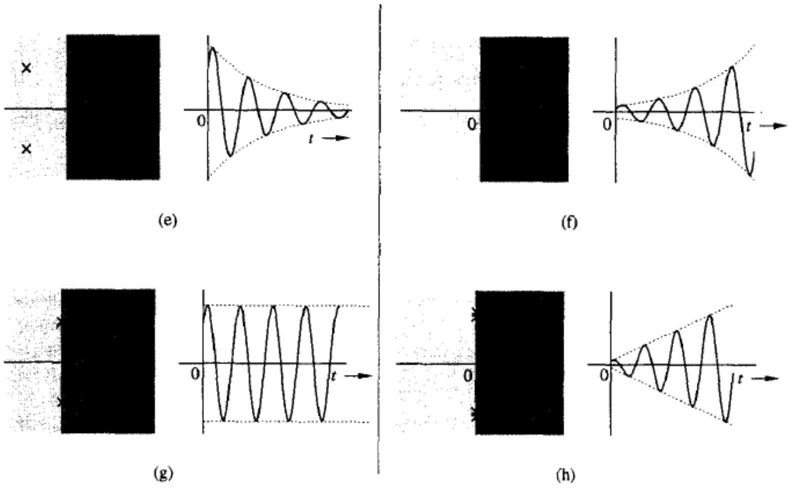
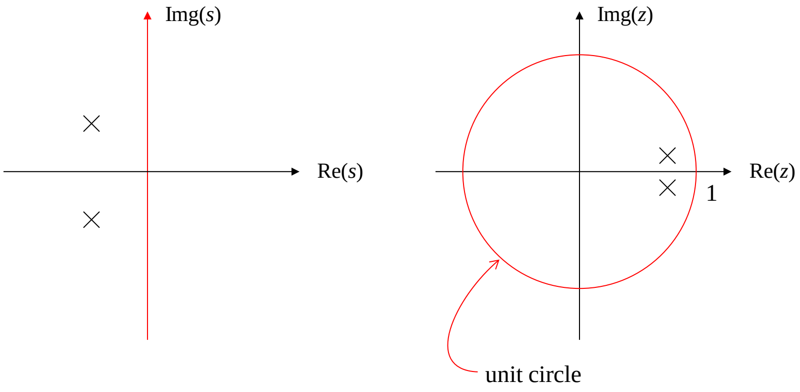


Fig. 2.16 Location of characteristic roots and the corresponding characteristic modes.

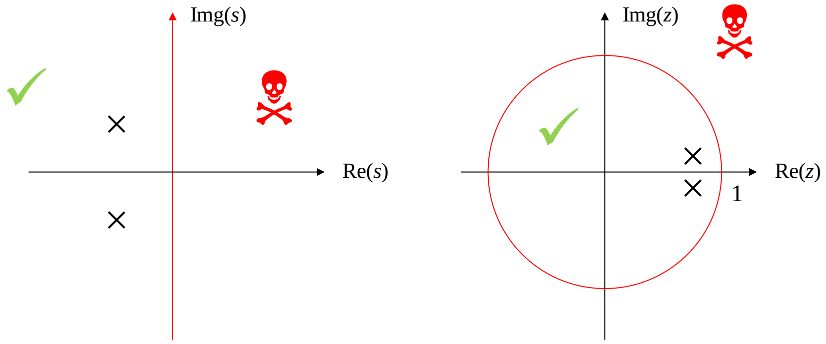
γ-plane Stability

- For a γ-Plane (e.g. the one the *z*-domain is embedded in) the unit circle is the system stability bound



γ-plane Stability

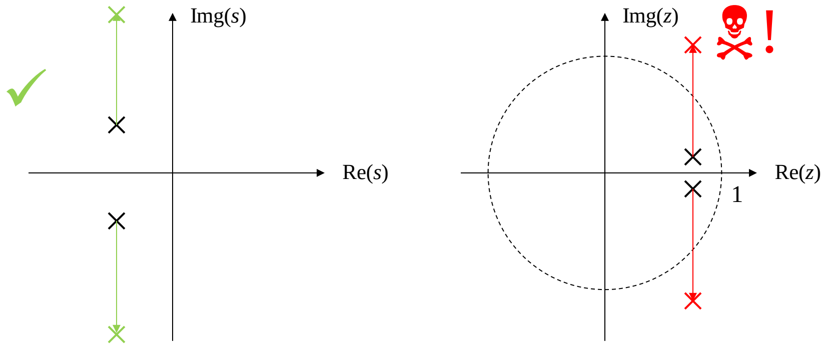
- That is, in the *z*-domain,
  the unit circle is the system stability bound



---

z-plane stability

- The z-plane root-locus in closed loop feedback behaves just like the s-plane:



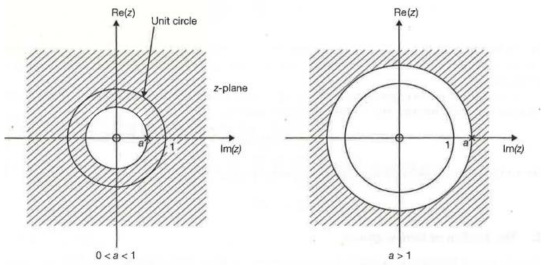
Region of Convergence

- For the convergence of X(z) we require that

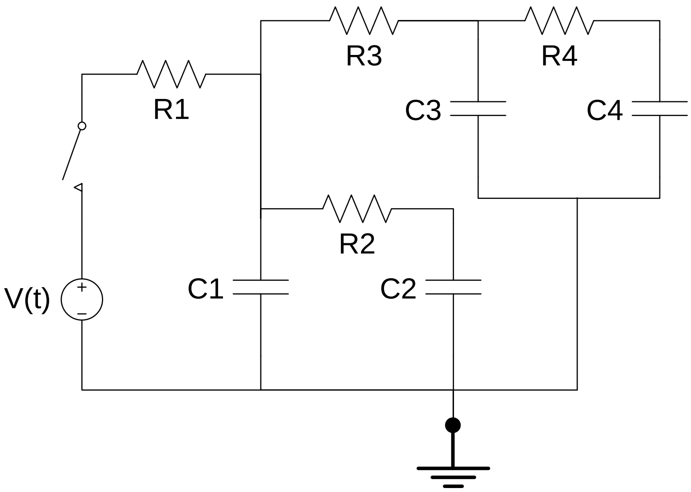
$$\sum_{n=0}^{\infty} |az^{-1}|^n < \infty$$

- Thus, the ROC is the range of values of z for which $|az^{-1}| < 1$ or, equivalently, $|z| > |a|$. Then

$$X(z) = \frac{z}{z-a} \qquad |z| > |a|$$

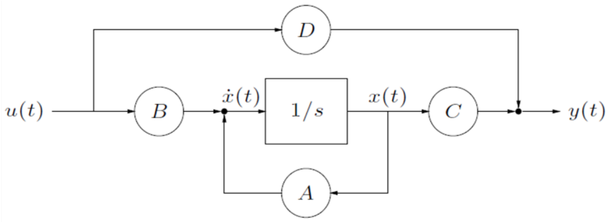


An Example Circuit…



Ref: Boyd, EE263, 13-14

Frame as a LDS ➔LTI LTS ➔LTID LTS



$$\dot{x} = \boldsymbol{A}x + \boldsymbol{B}u$$
$$y = \boldsymbol{C}x + \boldsymbol{D}u$$

- $\boldsymbol{A}x$ is the drift term (of $\dot{x}$)
- $\boldsymbol{B}u$ is the input term (of $\dot{x}$)

Ref: Boyd, EE263, 13-4

Transfer Function

Take the Laplace transform of $\dot{x} = \boldsymbol{A}x + \boldsymbol{B}u$

$$sX(s) - x(0) = \boldsymbol{A}X(s) + \boldsymbol{B}U(s)$$

$$X(s) = (sI - \boldsymbol{A})^{-\mathbf{1}}x(0) + (sI - A)\text{^} - 1\boldsymbol{B}U(s)$$

$$\Rightarrow x(t) = e^{t\boldsymbol{A}}x(0) + \int_0^t e^{(t-\tau)\boldsymbol{A}} \cdot Bu(\tau)\, d\tau$$

- $e^{t\boldsymbol{A}}x(0)$: unforced or autonomous response
- $e^{t\boldsymbol{A}}B$: input-to-state ➔ impulse response matrix
- $(sI - A)^{-1}\boldsymbol{B}$: transfer function or transfer matrix

Ref: Boyd, EE263, 13-6

Transfer Function [2]

with $y = \boldsymbol{C}x + \boldsymbol{D}u$ we have:

$$Y(s) = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}x(0) + (\boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + D)U(s)$$

$$\Rightarrow y(t) = \boldsymbol{C}e^{t\boldsymbol{A}}x(0) + \int_0^t \boldsymbol{C}e^{(t-\tau)\boldsymbol{A}} \cdot \boldsymbol{B}u(\tau)\, d\tau + \boldsymbol{D}u(t)$$

- $Ce^{tA}x(0)$:initial condition
- $H(s) = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D}$:transfer function/matrix
- $h(t) = \boldsymbol{C}e^{t\boldsymbol{A}} \cdot B + D\delta(t)$: impulse response

With zero initial conditions we have:

- $Y(s) = H(s)U(s), y = h * u$

Ref: Boyd, EE263, 13-7

Impulse Response

impulse response $h(t) = Ce^{tA}B + D\delta(t)$

with $x(0) = 0$, $y = h * u$, *i.e.*,

$$y_i(t) = \sum_{j=1}^{m} \int_0^t h_{ij}(t-\tau)u_j(\tau)\, d\tau$$

interpretations:

- $h_{ij}(t)$ is impulse response from jth input to ith output
- $h_{ij}(t)$ gives y_i when $u(t) = e_j\delta$
- $h_{ij}(\tau)$ shows how dependent output i is, on what input j was, τ seconds ago
- i indexes output; j indexes input; τ indexes time lag

Ref: Boyd, EE263, 13-9

Step Response

the *step response* or *step matrix* is given by

$$s(t) = \int_0^t h(\tau)\, d\tau$$

interpretations:

- $s_{ij}(t)$ is step response from jth input to ith output
- $s_{ij}(t)$ gives y_i when $u = e_j$ for $t \geq 0$

for invertible A, we have

$$s(t) = CA^{-1}\left(e^{tA} - I\right)B + D$$

Ref: Boyd, EE263, 13-10

Static (DC) Gain Matrix

- transfer function at $s = 0$ is $\boxed{H(0) = -CA^{-1}B + D \in \mathbf{R}^{m \times p}}$
- DC transfer function describes system under *static* conditions, *i.e.*, x, u, y constant:

  $$0 = \dot{x} = Ax + Bu, \qquad y = Cx + Du$$

  eliminate x to get $y = H(0)u$
- if system is stable,

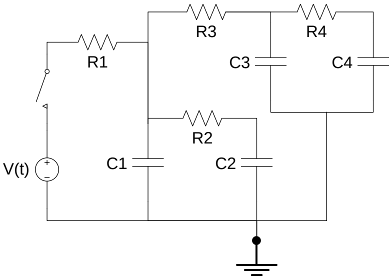
  $$H(0) = \int_0^\infty h(t)\, dt = \lim_{t \to \infty} s(t)$$

  (recall: $H(s) = \int_0^\infty e^{-st}h(t)\, dt$, $s(t) = \int_0^t h(\tau)\, d\tau$)

  if $u(t) \to u_\infty \in \mathbf{R}^m$, then $y(t) \to y_\infty \in \mathbf{R}^p$ where $y_\infty = H(0)u_\infty$

Ref: Boyd, EE263, 13-18

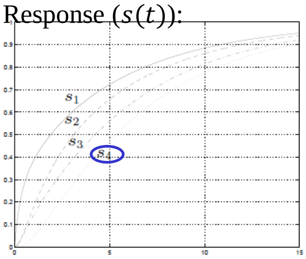
Back to the Example



$$\Rightarrow \quad \dot{x} = \begin{bmatrix} -3 & 1 & 1 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & -2 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u, \qquad y = x$$

```
> eig(A)
>   -3.9563
    -2.2091
    -0.6617
    -0.1729
```

Step Response ($s(t)$):



Ref: Boyd, EE263, 13-15

BREAK

Discretization with piecewise constant inputs

linear system $\dot{x} = Ax + Bu$, $y = Cx + Du$

suppose $u_d : \mathbf{Z}_+ \to \mathbf{R}^m$ is a sequence, and

$$u(t) = u_d(k) \qquad \text{for} \;\; kh \leq t < (k+1)h, \quad k = 0, 1, \ldots$$

define sequences

$$x_d(k) = x(kh), \qquad y_d(k) = y(kh), \qquad k = 0, 1, \ldots$$

- $h > 0$ is called the *sample interval* (for x and y) or *update interval* (for u)
- u is piecewise constant (called *zero-order-hold*)
- x_d, y_d are sampled versions of x, y

Ref: Boyd, EE263, 13-20

Discretization with piecewise constant inputs [2]

$$\begin{aligned} x_d(k+1) &= x((k+1)h) \\ &= e^{hA}x(kh) + \int_0^h e^{\tau A} Bu((k+1)h - \tau)\, d\tau \\ &= e^{hA}x_d(k) + \left(\int_0^h e^{\tau A}\, d\tau \right) B\, u_d(k) \end{aligned}$$

x_d, u_d, and y_d satisfy discrete-time LDS equations

$$x_d(k+1) = A_d x_d(k) + B_d u_d(k), \qquad y_d(k) = C_d x_d(k) + D_d u_d(k)$$

where

$$A_d = e^{hA}, \qquad B_d = \left(\int_0^h e^{\tau A}\, d\tau \right) B, \qquad C_d = C, \qquad D_d = D$$

called *discretized system*

if A is invertible, we can express integral as

$$\int_0^h e^{\tau A}\, d\tau = A^{-1}\left(e^{hA} - I\right)$$

stability: if eigenvalues of A are $\lambda_1, \ldots, \lambda_n$, then eigenvalues of A_d are $e^{h\lambda_1}, \ldots, e^{h\lambda_n}$

discretization preserves stability properties since

$$\Re\lambda_i < 0 \;\; \Leftrightarrow \;\; \left|e^{h\lambda_i}\right| < 1$$

for $h > 0$

Ref: Boyd, EE263, 13-22

Convolution

Convolution Definition

The convolution of two functions $f_1(t)$ and $f_2(t)$ is defined as:

$$f(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t-\tau) d\tau$$

$$= f_1(t) * f_2(t)$$

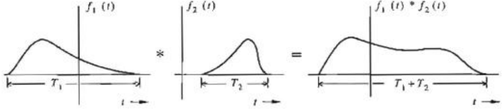
Source: URI ELE436

Convolution Properties

$$f_1(t) * f_2(t) \equiv \int_{-\infty}^{\infty} f_1(\tau) f_2(t-\tau)\, d\tau$$

Properties:

- Commutative: $f_1(t) * f_2(t) = f_2(t) * f_1(t)$
- Distributive: $f_1(t) * [f_2(t) + f_3(t)] = f_1(t) * f_2(t) + f_1(t) * f_3(t)$
- Associative: $f_1(t) * [f_2(t) * f_3(t)] = [f_1(t) * f_2(t)] * f_3(t)$
- Shift:
  if $f_1(t)*f_2(t)=c(t)$, then $f_1(t-\mathbf{T})*f_2(t)= f_1(t)*f_2(t-\mathbf{T})=c(t-\mathbf{T})$
- Identity (Convolution with an Impulse):
  $f(t) * \delta(t) = f(t)$
- Total Width:



Based on Lathi, *SPLS*, Sec 2.4-1

Convolution Properties [II]

- Convolution systems are **linear**:

$$h * (\alpha u_1 + \beta u_2) = \alpha(h * u_1) + \beta(h * u_2)$$

- Convolution systems are **causal**: the output y(t) at time t depends only on past inputs

- Convolution systems are **time-invariant**
  (if we shift the signal, the output similarly shifts)

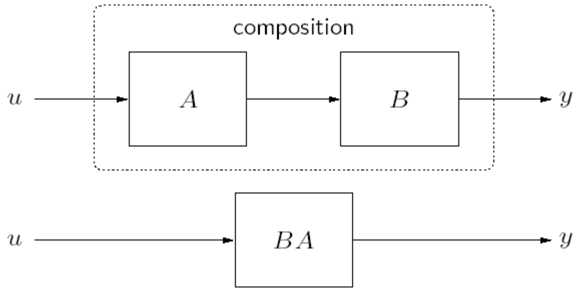
➔

$$\tilde{u}(t) = \begin{cases} 0 & t < T \\ u(t-T) & t \geq 0 \end{cases}$$

$$\tilde{y}(t) = \begin{cases} 0 & t < T \\ y(t-T) & t \geq 0 \end{cases}$$

Convolution Properties [III]

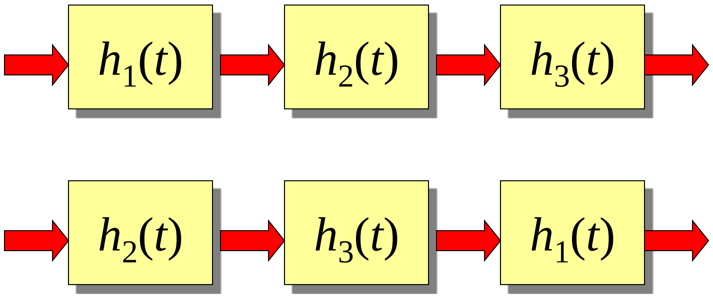
- Composition of convolution systems corresponds to:
  - multiplication of transfer functions
  - convolution of impulse responses



- Thus:
  - We can manipulate block diagrams with transfer functions as if they were simple gains
  - convolution systems commute with each other

Properties of Convolution: Distributive Property

$$[f_1(t) * f_2(t)] * f_3(t) = f_1(t) * [f_2(t) * f_3(t)]$$



- The two systems are identical!

Source: URI ELE436

Properties of Convolution: Commutative Property

$$\boxed{f_1(t) * f_2(t) = f_2(t) * f_1(t)}$$

$$f_1(t)*f_2(t) = \int_{-\infty}^{\infty} f_1(\tau)f_2(t-\tau)d\tau = \int_{\tau=-\infty}^{\tau=\infty} f_1(\tau)f_2(t-\tau)d\tau$$

$$= \int_{t-\tau=-\infty}^{t-\tau=\infty} f_1(t-\tau)f_2[t-(t-\tau)]d(t-\tau)$$

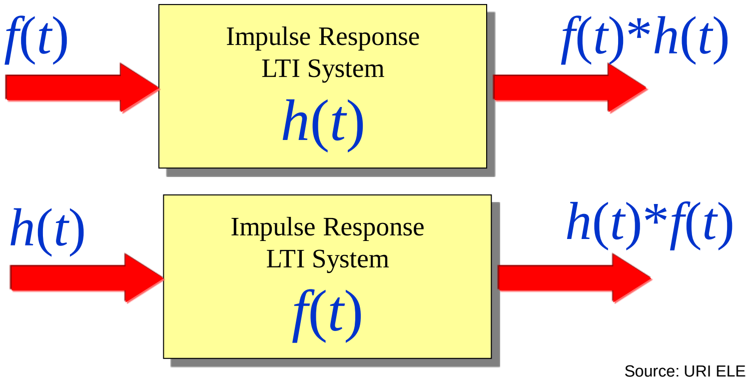
$$= -\int_{\tau=\infty}^{\tau=-\infty} f_1(t-\tau)f_2(\tau)d\tau$$

$$= \int_{-\infty}^{\infty} f_1(t-\tau)f_2(\tau)d\tau = f_2(t) * f_1(t)$$

Source: URI ELE436

Properties of Convolution: LTI System Response

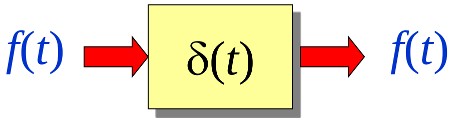
$$\boxed{f_1(t) * f_2(t) = f_2(t) * f_1(t)}$$



Source: URI ELE436

Properties of Convolution

$$\boxed{f(t) * \delta(t) = f(t)}$$

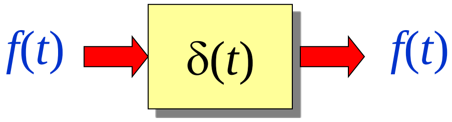


$$f(t) * \delta(t) = \int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)d\tau$$
$$= \int_{-\infty}^{\infty} f(t-\tau)\delta(\tau)d\tau$$
$$= f(t)$$

Source: URI ELE436

Properties of Convolution

$$\boxed{f(t) * \delta(t) = f(t)}$$



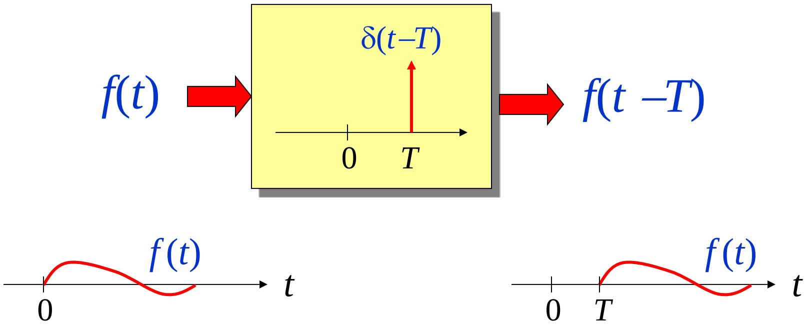
$$\boxed{f(t) * \delta(t-T) = f(t-T)}$$

$$f(t) * \delta(t-T) = \int_{-\infty}^{\infty} f(\tau)\delta(t-T-\tau)d\tau$$
$$= \int_{-\infty}^{\infty} f(t-T-\tau)\delta(\tau)d\tau$$
$$= f(t-T)$$

Source: URI ELE436

Properties of Convolution

$$\boxed{f(t) * \delta(t-T) = f(t-T)}$$



Source: URI ELE436

Properties of Convolution

$$\boxed{f_1(t) * f_2(t) \xleftrightarrow{\mathcal{F}} F_1(j\omega)F_2(j\omega)}$$

$$F[f_1(t) * f_2(t)] = \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f_1(\tau) f_2(t-\tau)d\tau\right]e^{-j\omega t}dt$$

$$= \int_{-\infty}^{\infty} f_1(\tau)\left[\int_{-\infty}^{\infty} f_2(t-\tau)e^{-j\omega t}dt\right]d\tau$$

$$= \int_{-\infty}^{\infty} f_1(\tau)F_2(j\omega)e^{-j\omega\tau}d\tau$$

$$= F_2(j\omega)\int_{-\infty}^{\infty} f_1(\tau)e^{-j\omega\tau}d\tau \quad = F_1(j\omega)F_2(j\omega)$$

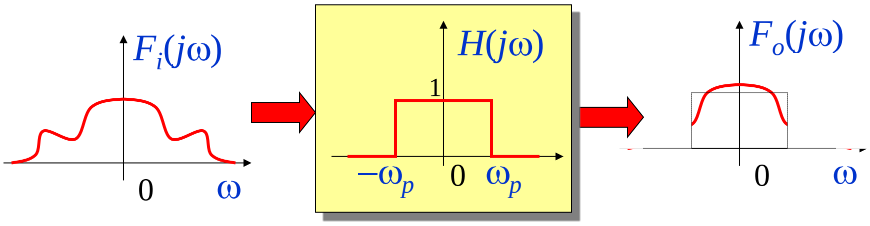
Time Domain
convolution

Frequency Domain
multiplication

Source: URI ELE436

Properties of Convolution

$$f_1(t) * f_2(t) \xleftrightarrow{\mathcal{F}} F_1(j\omega)F_2(j\omega)$$

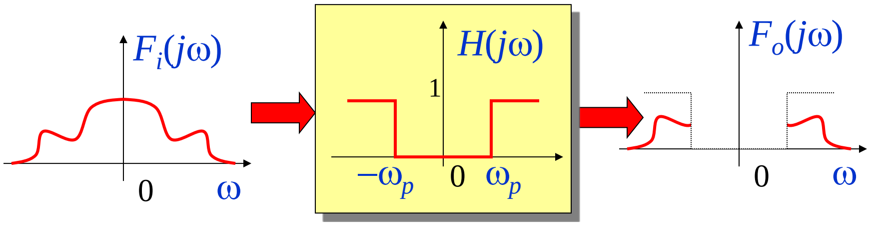


An Ideal Low-Pass Filter

Source: URI ELE436

Properties of Convolution

$$f_1(t) * f_2(t) \xleftrightarrow{\mathcal{F}} F_1(j\omega)F_2(j\omega)$$



An Ideal High-Pass Filter

Source: URI ELE436

Discrete Convolution

$$y[n] = x[n]*h[n] = \sum_{m=-\infty}^{\infty} x[m]\cdot h[n-m] = \sum_{m=-\infty}^{\infty} h[m]\cdot x[n-m]$$

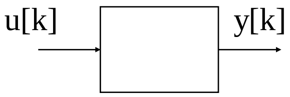
Consider this for the discrete case:

1. Rename the independent variable as **m**. You now have **x**[**m**] and **h**[**m**]. Flip **h**[**m**] over the origin. This is **h**[**-m**]
2. Shift **h**[**-m**] as far left as possible to a point "**n**", where the two signals barely touch. This is **h**[**n-m**]
3. Multiply the two signals and sum over all values of **m**. This is the convolution sum for the specific "**n**" picked above.
4. Shift / move **h**[**-m**] to the right by one sample, and obtain a new **h**[**n-m**]. Multiply and sum over all **m**.
5. 5. Repeat 2~4 until **h**[**n-m**] no longer overlaps with **x**[**m**], i.e., shifted out of the **x**[**m**] zone.

➔ The "*n*" dependency of *y*[*n*] deserves some care:
For each value of "*n*" the convolution sum must be computed ***separately*** over all values of a dummy variable "*m*".

Discrete-Time Systems & Discrete Convolution [1]

Will consider linear time-invariant (LTI) systems



Linear :

input u1[k] -> output y1[k]

input u2[k] -> output y2[k]

hence a.u1[k]+b.u2[k]-> a.y1[k]+b.y2[k]

Time-invariant (shift-invariant)

input u[k] -> output y[k]

hence input u[k-T] -> output y[k-T]

Discrete-Time Systems & Discrete Convolution [2]

Will consider causal systems

iff for all input signals with u[k]=0,k<0 -> output y[k]=0,k<0

Impulse response

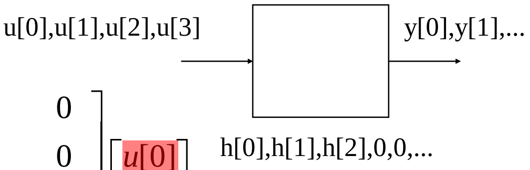
input …,0,0, $\overset{K=0}{1}$,0,0,0,...-> output …,0,0, $\overset{K=0}{\text{h[0]}}$,h[1],h[2],h[3],...

General input u[0],u[1],u[2],u[3] (cfr. linearity & shift-invariance!)

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \\ y[4] \\ y[5] \end{bmatrix} = \begin{bmatrix} h[0] & 0 & 0 & 0 \\ h[1] & h[0] & 0 & 0 \\ h[2] & h[1] & h[0] & 0 \\ 0 & h[2] & h[1] & h[0] \\ 0 & 0 & h[2] & h[1] \\ 0 & 0 & 0 & h[2] \end{bmatrix} . \begin{bmatrix} u[0] \\ u[1] \\ u[2] \\ u[3] \end{bmatrix}$$

this is called a `Toeplitz' matrix

Discrete-Time Systems & Discrete Convolution [3]



$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \\ y[4] \\ y[5] \end{bmatrix} = \begin{bmatrix} h[0] & 0 & 0 & 0 \\ h[1] & h[0] & 0 & 0 \\ h[2] & h[1] & h[0] & 0 \\ 0 & h[2] & h[1] & h[0] \\ 0 & 0 & h[2] & h[1] \\ 0 & 0 & 0 & h[2] \end{bmatrix} . \begin{bmatrix} u[0] \\ u[1] \\ u[2] \\ u[3] \end{bmatrix}$$

$$y[k] = \sum_{\bar{k}} h[k - \bar{k}].u[\bar{k}] \overset{\Delta}{=} h[k] * u[k]$$

= `convolution sum'

(=more convenient than Toeplitz matrix notation when considering (infinitely) long input and impulse response sequences

Discrete-Time Systems & Discrete Convolution [4]

Z-Transform of <u>system</u> h[k] and <u>signals</u> u[k],y[k]

Definition:

$$H(z)\overset{D}{=}\sum_k h[k].z^{-k} \quad U(z)\overset{D}{=}\sum_k u[k].z^{-k} \quad Y(z)\overset{D}{=}\sum_k y[k].z^{-k}$$

Input/output relation:

$$\underbrace{\begin{bmatrix}1 & z^{-1} & z^{-2} & z^{-3} & z^{-4} & z^{-5}\end{bmatrix}.\begin{bmatrix}y[0]\\y[1]\\y[2]\\y[3]\\y[4]\\y[5]\end{bmatrix}}_{Y(z)} = \underbrace{\begin{bmatrix}1 & z^{-1} & z^{-2} & z^{-3} & z^{-4} & z^{-5}\end{bmatrix}.\begin{bmatrix}h[0] & 0 & 0 & 0\\h[1] & h[0] & 0 & 0\\h[2] & h[1] & h[0] & 0\\0 & h[2] & h[1] & h[0]\\0 & 0 & h[2] & h[1]\\0 & 0 & 0 & h[2]\end{bmatrix}}_{H(z).\begin{bmatrix}1 & z-1 & z-2 & z-3\end{bmatrix}}.\begin{bmatrix}u[0]\\u[1]\\u[2]\\u[3]\end{bmatrix}$$

$$\Rightarrow Y(z) = H(z).U(z)$$ H(z) is `transfer function’

Matrix Formulation of Convolution

$$\mathbf{y} = \mathbf{Hx}$$

$$\begin{bmatrix}3\\8\\14\\20\\26\\14\\5\end{bmatrix} = \begin{bmatrix}1&2&3&0&0&0&0&0&0\\0&1&2&3&0&0&0&0&0\\0&0&1&2&3&0&0&0&0\\0&0&0&1&2&3&0&0&0\\0&0&0&0&1&2&3&0&0\\0&0&0&0&0&1&2&3&0\\0&0&0&0&0&0&1&2&3\end{bmatrix}.\begin{bmatrix}0\\0\\1\\2\\3\\4\\5\\0\\0\end{bmatrix}$$

Toeplitz Matrix

Graphical Understanding of Convolution

➔ For c(***τ***)= $(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,d\tau$:

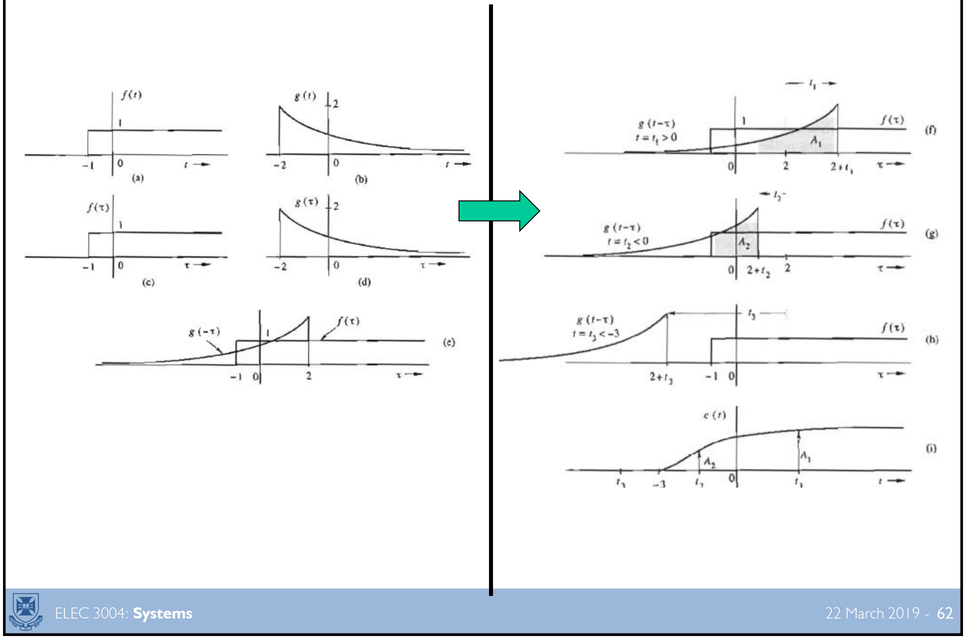
1. Keep the function ***f (τ)*** fixed
2. **Flip** (invert) the function ***g(τ)*** about the vertical axis (τ=0)
   = this is ***g(-τ)***
3. **Shift** this frame (***g(-τ)***) along τ (horizontal axis) by $\mathbf{t_0}$.
   = this is $\boldsymbol{g(t_0 - \tau)}$

➔ For c($\mathbf{t_0}$):

4. $\mathbf{c(t_0)}$ = the area under the product of ***f (τ)* and** $\boldsymbol{g(t_0 - \tau)}$

5. Repeat this procedure, shifting the frame by different values (positive and negative) to obtain **<u>c(t) for all values of t</u>**.

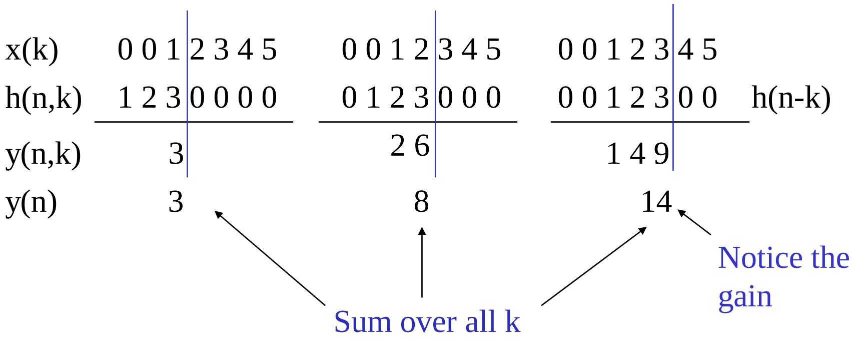
Graphical Understanding of Convolution (Ex)



Another View

e.g. convolution

x(n) = 1 2 3 4 5
h(n) = 3 2 1



x(k)	0 0 1 2 3 4 5	0 0 1 2 3 4 5	0 0 1 2 3 4 5	
h(n,k)	1 2 3 0 0 0 0	0 1 2 3 0 0 0	0 0 1 2 3 0 0	h(n-k)
y(n,k)	3	2 6	1 4 9	
y(n)	3	8	14	

Sum over all k

Notice the gain

Convolution & Systems

- Convolution system with input u (u(t) = 0, t < 0) and output y:

$$y(t) = \int_0^t h(\tau)u(t-\tau)\,d\tau = \int_0^t h(t-\tau)u(\tau)\,d\tau$$

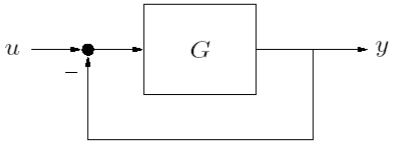
- abbreviated:

$$y = h * u$$

- in the frequency domain:

$$Y(s) = H(s)U(s)$$

Convolution & Feedback



- In the time domain:

$$y(t) = \int_0^t g(t-\tau)(u(\tau) - y(\tau))\, d\tau$$

- In the frequency domain:
  - Y=G(U-Y)

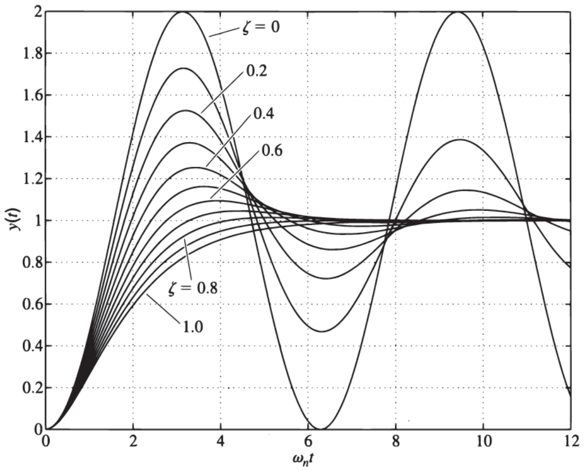
  ➔Y(s) = H(s)U(s)

$$H(s) = \frac{G(s)}{1 + G(s)}$$

2nd Order LTID

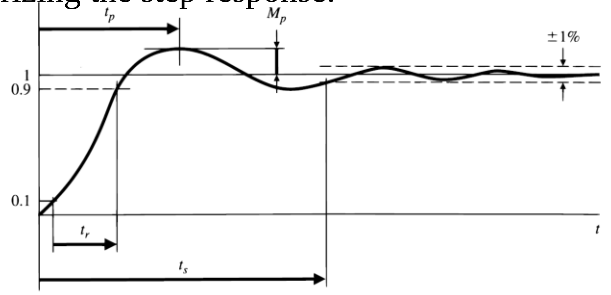
2^{nd} Order System Response

- Response of a 2^{nd} order system to increasing levels of damping:



2^{nd} Order System Specifications

Characterizing the step response:

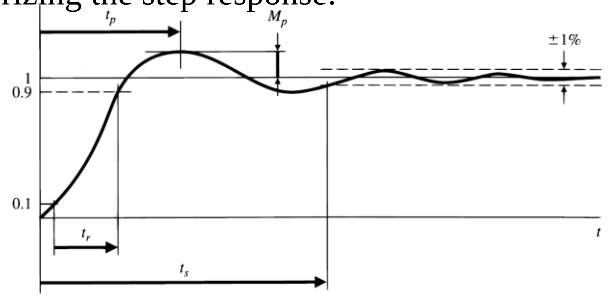


- Rise time (10% → 90%): $t_r \approx \frac{1.8}{\omega_0}$
- Overshoot: $M_p \approx \frac{e^{-\pi\zeta}}{\sqrt{1-\zeta^2}}$
- Settling time (to 1%): $t_s = \frac{4.6}{\zeta\omega_0}$
- Steady state error to unit step: e_{ss}
- Phase margin:

$$\phi_{PM} \approx 100\zeta$$

2nd Order System Specifications

Characterizing the step response:

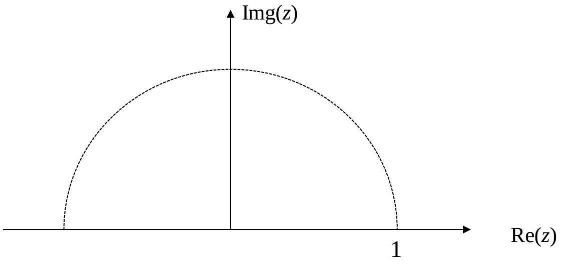


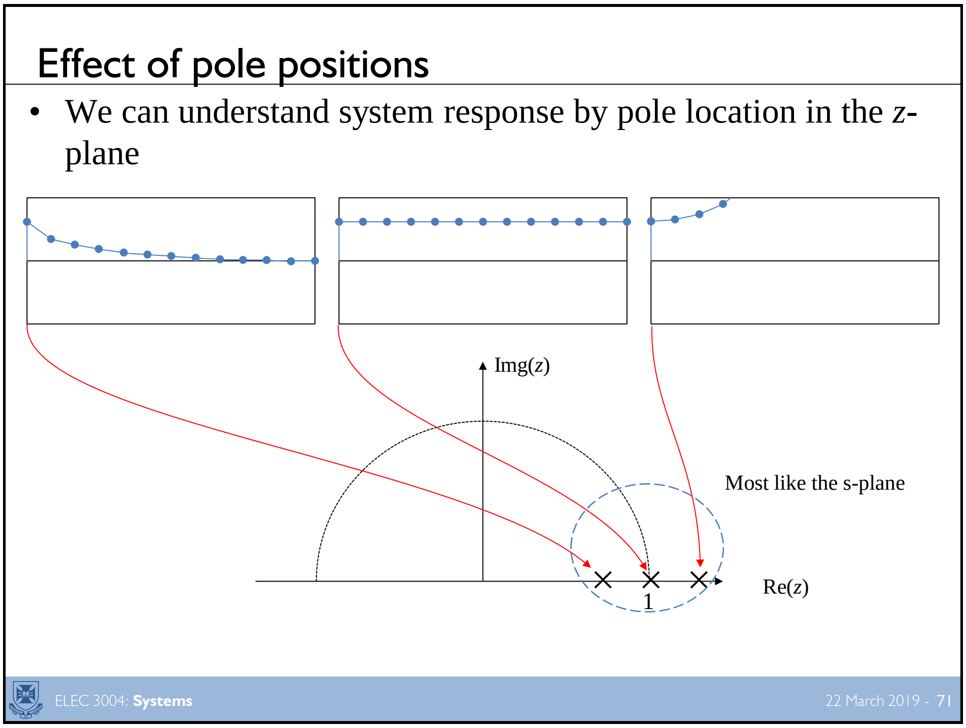
- Rise time (10% → 90%) & Overshoot:
  t_r, M_p → ζ, ω_0 : Locations of dominant poles
- Settling time (to 1%):
  t_s → radius of poles: $|z| < 0.01^{\frac{T}{t_s}}$
- Steady state error to unit step:
  e_{ss} → final value theorem $e_{ss} = \lim_{z \to 1} \{(z-1)\, F(z)\}$

The z-plane [for all pole systems]

- We can understand system response by pole location in the *z*-plane

[Adapted from Franklin, Powell and Emami-Naeini]

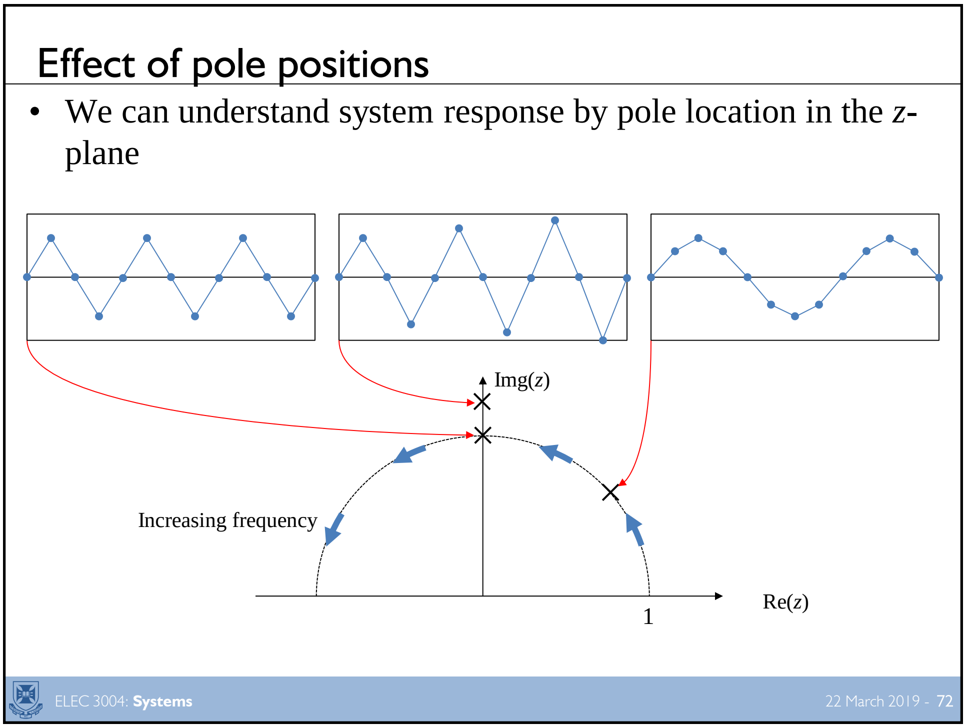




Effect of pole positions

- We can understand system response by pole location in the z-plane

---

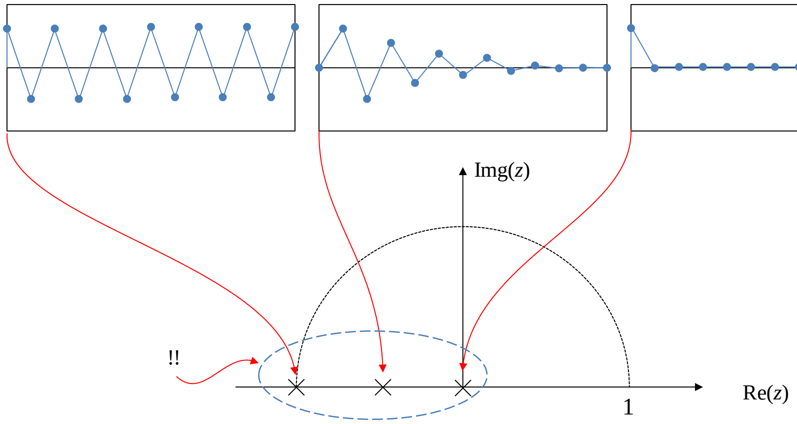


Effect of pole positions

- We can understand system response by pole location in the z-plane

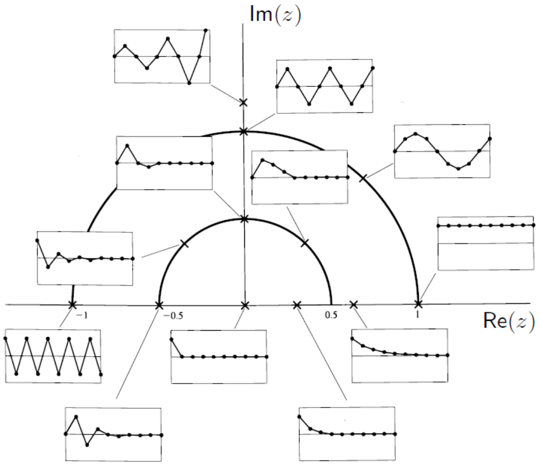
Effect of pole positions

- We can understand system response by pole location in the *z*-plane



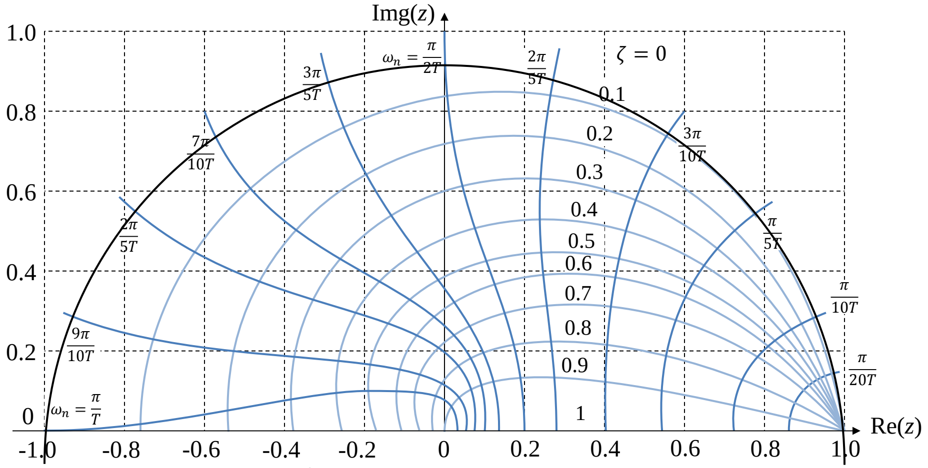
Pole positions in the z-plane

- Poles inside the unit circle are **stable**
- Poles outside the unit circle **unstable**
- Poles on the unit circle are oscillatory
- Real poles at 0 < z < 1 give exponential response
- Higher frequency of oscillation for larger
- Lower apparent damping for larer and r



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]

Second Order Digital Systems

Consider the z-transform of a decaying exponential signal:

$$y(t) = e^{-at}\cos(bt)\mathcal{U}(t) \qquad (\mathcal{U}(t) = \text{unit step})$$

- ⋆ sample: $y(kT) = r^k\cos(k\theta)\mathcal{U}(kT)$ with $r = e^{-aT}$ & $\theta = bT$
- ⋆ transform: $Y(z) = \frac{1}{2}\frac{z}{(z-re^{j\theta})} + \frac{1}{2}\frac{z}{(z-re^{-j\theta})}$

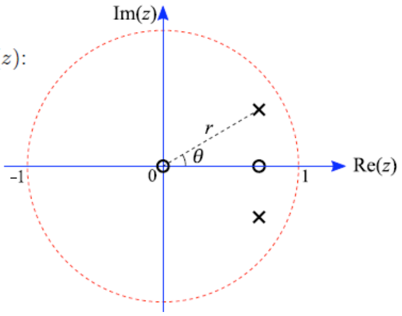
$$= \frac{z(z-r\cos\theta)}{(z-re^{j\theta})(z-re^{-j\theta})}$$

- ⋆ e.g. y_k is the pulse response of $G(z)$:

$$G(z) = \frac{z(z-r\cos\theta)}{(z-re^{j\theta})(z-re^{-j\theta})}$$

poles: $\begin{cases} z = re^{j\theta} \\ z = re^{-j\theta} \end{cases}$

zeros: $\begin{cases} z = 0 \\ z = r\cos\theta \end{cases}$



Response of 2nd order system [1/3]

Responses for varying r:

- ▷ $r < 1$

  $\Downarrow$

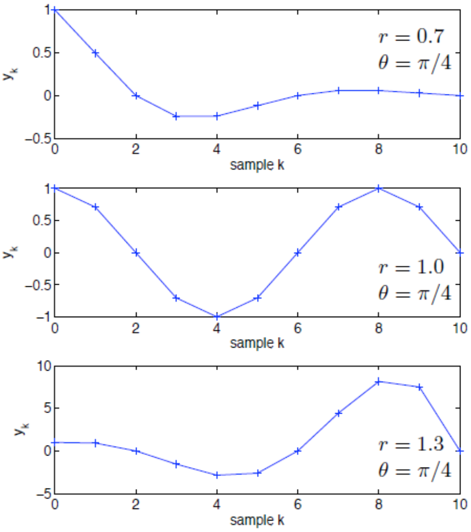
  exponentially decaying envelope
- ▷ $r = 1$

  $\Downarrow$

  sinusoidal response with $2\pi/\theta$ samples per period
- ▷ $r > 1$

  $\Downarrow$

  exponentially increasing envelope



Response of 2nd order system [2/3]

Responses for varying θ:

- ▷ $\theta = 0$

  $\Downarrow$

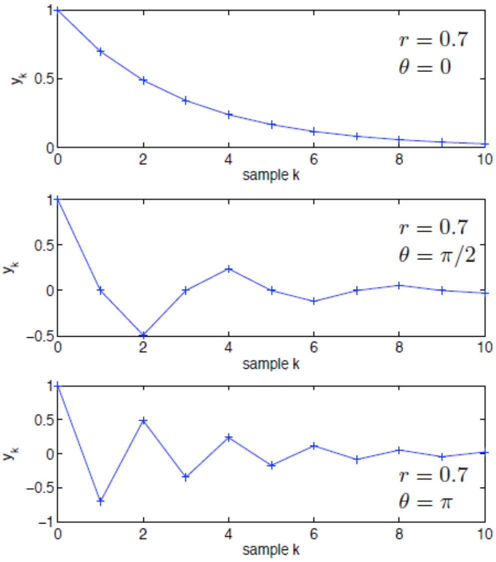
  decaying exponential
- ▷ $\theta = \pi/2$

  $\Downarrow$

  $2\pi/\theta = 4$ samples per period
- ▷ $\theta = \pi$

  $\Downarrow$

  2 samples per period



Response of 2nd order system [3/3]

Some special cases:

- for $\theta = 0$, $Y(z)$ simplifies to:

$$Y(z) = \frac{z}{z - r}$$

$\Longrightarrow$ exponentially decaying response

- when $\theta = 0$ and $r = 1$:

$$Y(z) = \frac{z}{z - 1}$$

$\Longrightarrow$ unit step

- when $r = 0$:

$$Y(z) = 1$$

$\Longrightarrow$ unit pulse

- when $\theta = 0$ and $-1 < r < 0$:

samples of alternating signs

Discrete-time transfer function

take $\mathcal{Z}$-transform of system equations

$$x(t+1) = Ax(t) + Bu(t), \qquad y(t) = Cx(t) + Du(t)$$

yields

$$zX(z) - zx(0) = AX(z) + BU(z), \qquad Y(z) = CX(z) + DU(z)$$

solve for $X(z)$ to get

$$X(z) = (zI - A)^{-1}zx(0) + (zI - A)^{-1}BU(z)$$

(note extra z in first term!)

hence

$$Y(z) = H(z)U(z) + C(zI - A)^{-1}zx(0)$$

where $H(z) = C(zI - A)^{-1}B + D$ is the *discrete-time transfer function*

note power series expansion of resolvent:

$$(zI - A)^{-1} = z^{-1}I + z^{-2}A + z^{-3}A^2 + \cdots$$

Source: Boyd, Lecture Notes for EE263, 13-39

Ex: System Specifications → Control Design [1/4]

Design a controller for a system with:

- A continuous transfer function: $G(s) = \dfrac{0.1}{s(s+0.1)}$
- A discrete ZOH sampler
- Sampling time (T_s): T_s= 1s
- Controller:

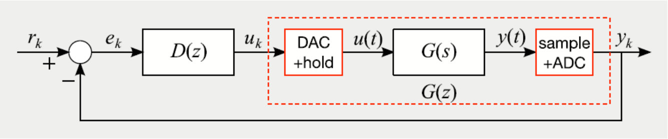
$$u_k = -0.5u_{k-1} + 13\left(e_k - 0.88e_{k-1}\right)$$

The closed loop system is required to have:

- $M_p < 16\%$
- $t_s < 10$ s
- $e_{ss} < 1$

Ex: System Specifications → Control Design [2/4]

1. (a) Find the pulse transfer function of $G(s)$ plus the ZOH



$$G(z) = (1 - z^{-1})\mathcal{Z}\left\{\frac{G(s)}{s}\right\} = \frac{(z-1)}{z}\mathcal{Z}\left\{\frac{0.1}{s^2(s+0.1)}\right\}$$

e.g. look up $\mathcal{Z}\{a/s^2(s+a)\}$ in tables:

$$G(z) = \frac{(z-1)}{z}\,\frac{z\Big((0.1-1+e^{-0.1})z + (1-e^{-0.1}-0.1e^{-0.1})\Big)}{0.1(z-1)^2(z-e^{-0.1})}$$
$$= \frac{0.0484(z+0.9672)}{(z-1)(z-0.9048)}$$

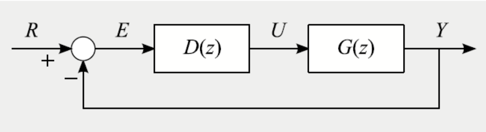
(b) Find the controller transfer function (using $z =$ shift operator):

$$\frac{U(z)}{E(z)} = D(z) = 13\frac{(1-0.88z^{-1})}{(1+0.5z^{-1})} = 13\frac{(z-0.88)}{(z+0.5)}$$

Ex: System Specifications → Control Design [3/4]

2. Check the steady state error e_{ss} when $r_k =$ unit ramp

$$e_{ss} = \lim_{k\to\infty} e_k = \lim_{z\to 1} (z-1)E(z)$$



$$\frac{E(z)}{R(z)} = \frac{1}{1+D(z)G(z)}$$

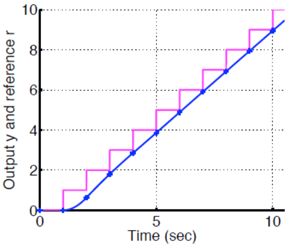
$$R(z) = \frac{Tz}{(z-1)^2}$$

so $$e_{ss} = \lim_{z\to 1}\left\{(z-1)\frac{Tz}{(z-1)^2}\frac{1}{1+D(z)G(z)}\right\} = \lim_{z\to 1}\frac{T}{(z-1)D(z)G(z)}$$

$$= \lim_{z\to 1}\frac{T}{(z-1)\dfrac{0.0484(z+0.9672)}{(z-1)(z-0.9048)}D(1)}$$

$$= \frac{1-0.9048}{0.0484(1+0.9672)D(1)} = 0.96$$

$\implies e_{ss} < 1$ (as required)



Ex: System Specifications → Control Design [4/4]

3. Step response: overshoot $M_p < 16\% \implies \zeta > 0.5$

settling time $t_s < 10 \implies |z| < 0.01^{1/10} = 0.63$

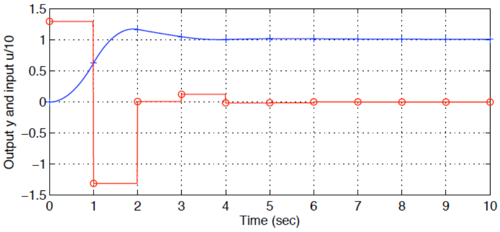
The closed loop poles are the roots of $1 + D(z)G(z) = 0$, i.e.

$$1 + 13\frac{(z-0.88)}{(z+0.5)}\frac{0.0484(z+0.9672)}{(z-1)(z-0.9048)} = 0$$

$$\implies z = 0.88,\ -0.050 \pm j0.304$$

But the pole at $z = 0.88$ is cancelled by controller zero at $z = 0.88$, and

$$z = -0.050 \pm j0.304 = re^{\pm j\theta} \implies \begin{cases} r = 0.31,\ \theta = 1.73 \\ \zeta = 0.56 \end{cases}$$



all specs satisfied!

Next Time…

- **Frequency Response**

- Review:
  - Chapter 10 of Lathi

- Nothing like a filter to smooth a rough signals ☺