Convolution Integral

Impulse Response *h*(*t*): output of linear system when input is a delta function

Linear Time-Invariant Circuit

Input

Output

$$\delta(t) \rightarrow \boxed{\text{LS}} \rightarrow y(t) = h(t)$$

$$\delta(t-\lambda) \rightarrow \boxed{\text{LS}} \rightarrow y(t) = h(t-\lambda)$$

$$x(\lambda)\,\delta(t-\lambda) \rightarrow \boxed{\text{LS}} \rightarrow y(t) = x(\lambda)\,h(t-\lambda)$$

$$\int_{-\infty}^{\infty} x(\lambda)\,\delta(t-\lambda)\,d\lambda \rightarrow \boxed{\text{LS}} \rightarrow y(t) = \int_{-\infty}^{\infty} x(\lambda)\,h(t-\lambda)\,d\lambda$$

$$x(t) \rightarrow \boxed{\text{LS}} \rightarrow y(t) = \int_{0}^{t} x(\lambda)\,h(t-\lambda)\,d\lambda$$

y(*t*) cannot depend on excitations occurring after time *t*

Assumes $x(t) = 0$ for $t < 0$

$$y(t) = x(t) * h(t) = \int_{0}^{t} x(\lambda)\,h(t-\lambda)\,d\lambda.$$

Definition of convolution

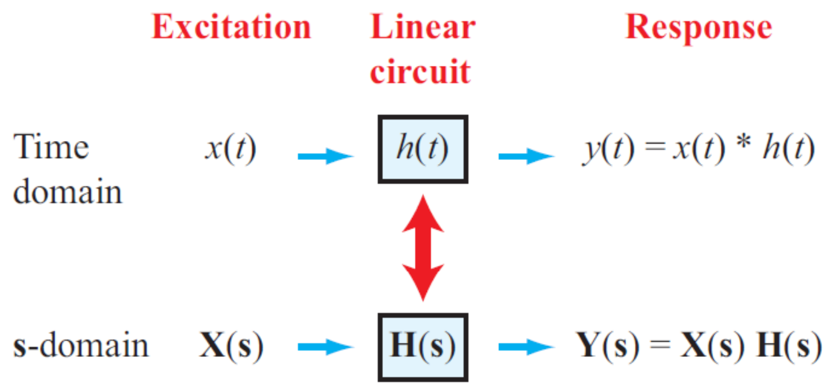
Transfer Function

In the **s-domain**, the circuit is characterized by a **transfer function** $\mathbf{H}(\mathbf{s})$, defined as the ratio of the **output** $\mathbf{Y}(\mathbf{s})$ to the **input** $\mathbf{X}(\mathbf{s})$, assuming that all initial conditions relating to currents and voltages in the circuit are zero at $t = 0^-$.

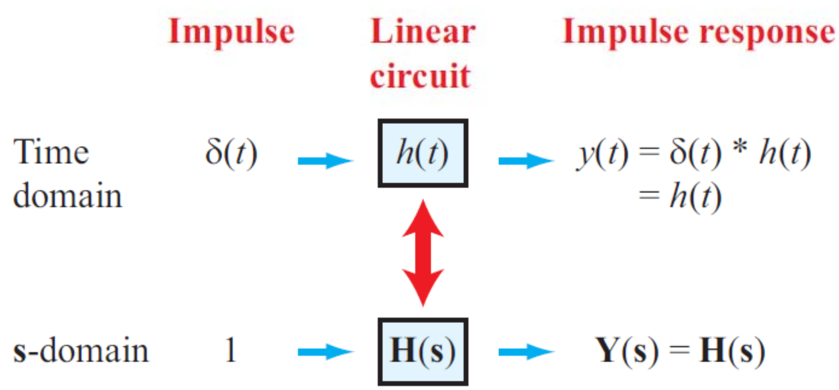
$$\mathbf{H}(\mathbf{s}) = \frac{\mathbf{Y}(\mathbf{s})}{\mathbf{X}(\mathbf{s})}$$

$$\mathbf{H}(\mathbf{s}) = \mathbf{Y}(\mathbf{s}), \qquad (\text{when } x(t) = \delta(t)).$$

Transfer Function (cont.)



(a) any $x(t)$



(b) $x(t) = \delta(t)$

$$x(t) * h(t) \leftrightarrow \mathbf{X}(\mathbf{s})\,\mathbf{H}(\mathbf{s}),$$

Convolution in time domain

Multiplication in s-domain

$$h(t) \leftrightarrow \mathbf{H}(\mathbf{s}).$$

(impulse response) (transfer function)

Convolution Integral

- Can be used to determine output response entirely in the time domain
- Can be useful when input is a sequence of experimental data or not a function with a definable Laplace transform
- Convolution can be performed by shifting *h*(*t*) or *x*(*t*):

$$y(t) = x(t) * h(t) = \int_0^t x(\lambda)\, \underbrace{h(t-\lambda)}_{h(t)\text{ shifted}}\, d\lambda.$$

$$y(t) = x(t) * h(t) = \int_0^t \underbrace{x(t-\lambda)}_{x(t)\text{ shifted}}\, h(\lambda)\, d\lambda.$$

↓ s domain

$$\frac{V_o(s)}{V_i(s)} = H(s)$$ [ignore initial conditions]

Transfer function in s-domain
System function

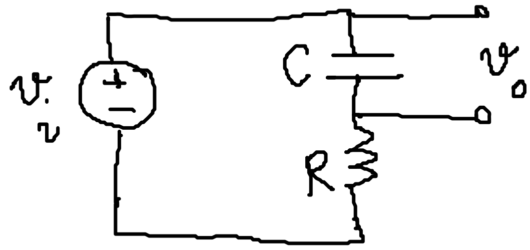
↓ Frequency domain

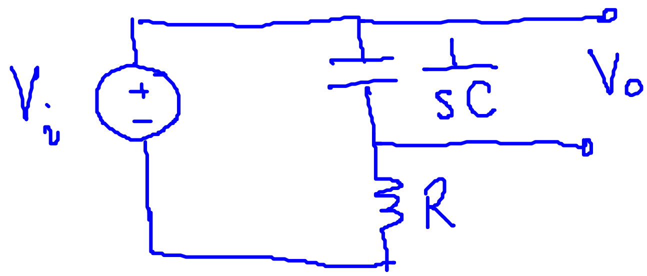
$$H(j\omega) = H(s)\Big|_{s=j\omega}$$

Magnitude Response $|H(j\omega)|$

Phase Response $\angle H(j\omega)$

RC filter



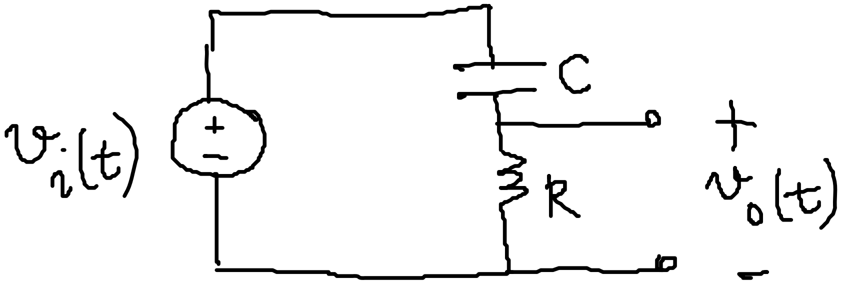


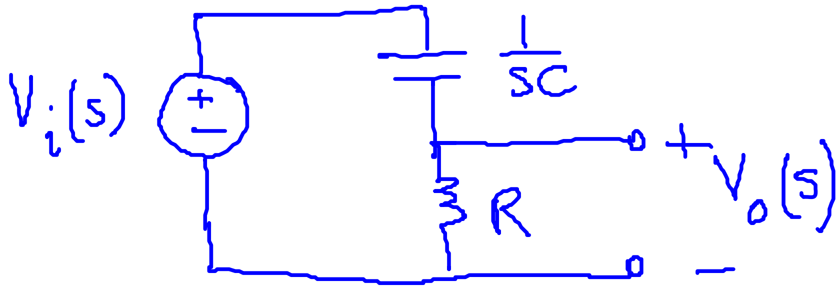
$$V_o = \frac{\frac{1}{sC}}{R + \frac{1}{sC}} V_i \qquad \frac{V_o}{V_i} = \frac{1}{1 + sCR}$$

$$H(s) = \frac{1}{1 + sCR}$$ Transfer function

Freq. response

$$H(j\omega) = \frac{1}{1 + j\omega CR}$$



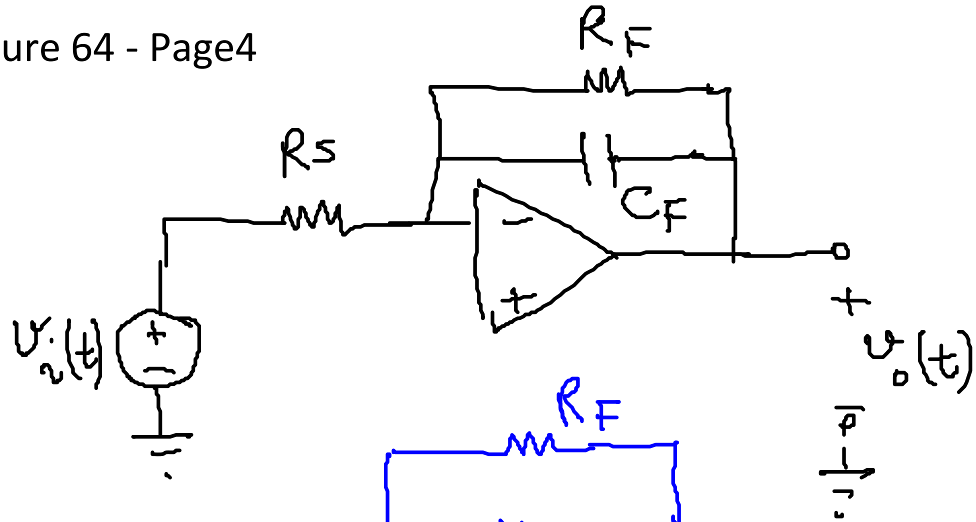


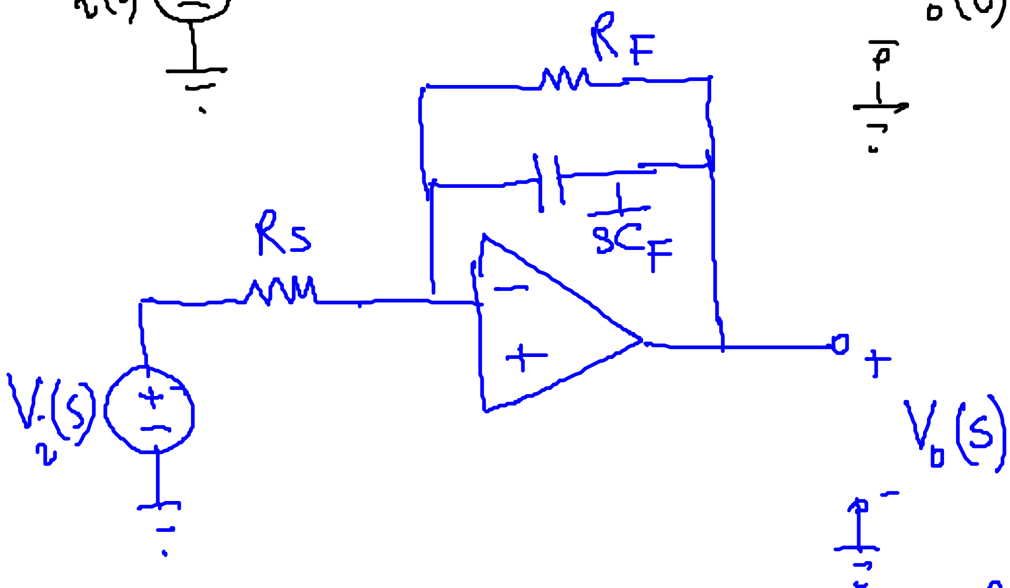
$$V_o = \frac{R}{1 + \frac{1}{sC}} V_i = \frac{sCR}{1+sCR} V_i$$

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

Frequency Response

$$H(j\omega) = \frac{j\omega CR}{1+j\omega CR}$$





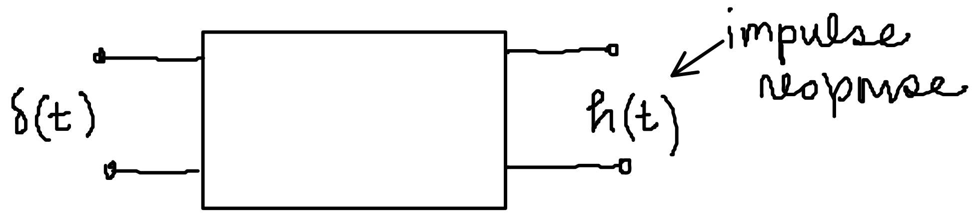
$$Z_F = R_F \,\|\, \frac{1}{sC_F} = \frac{R_F \frac{1}{sC_F}}{R_F + \frac{1}{sC_F}}$$

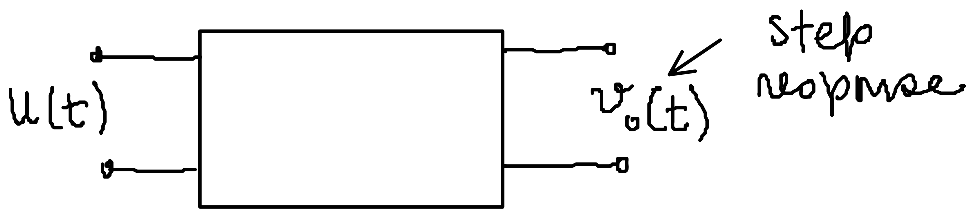
$$Z_F = \frac{R_F}{1 + sC_F R_F}$$

$$\frac{V_o}{V_i} = -\frac{Z_F}{R_S} = -\frac{1}{R_S}\,\frac{R_F}{1 + sC_F R_F}$$

$$H(s) = -\frac{R_F}{R_S}\,\frac{1}{1 + sC_F R_F}$$

$$H(j\omega) = -\frac{R_F}{R_S}\,\frac{1}{1 + j\omega C_F R_F}$$


$\delta(t)$
$h(t)$
impulse response


$u(t)$
$v_o(t)$
step response

MATLAB

```
>> n = [2 5 1];
>> d = [1 2 3];
>> tf(n,d)
```

transfer function

```
Transfer function:
2 s^2 + 5 s + 1
---------------
 s^2 + 2 s + 3

>> tf([2 5 1], [1 2 3])

Transfer function:
2 s^2 + 5 s + 1
---------------
 s^2 + 2 s + 3

>>
>> pole(tf(n,d))
```

poles

```
ans =

  -1.0000 + 1.4142i
  -1.0000 - 1.4142i

>> zero(tf(n,d))
```

zeros

```
ans =

   -2.2808
   -0.2192
```

Lecture 64 - Page 7

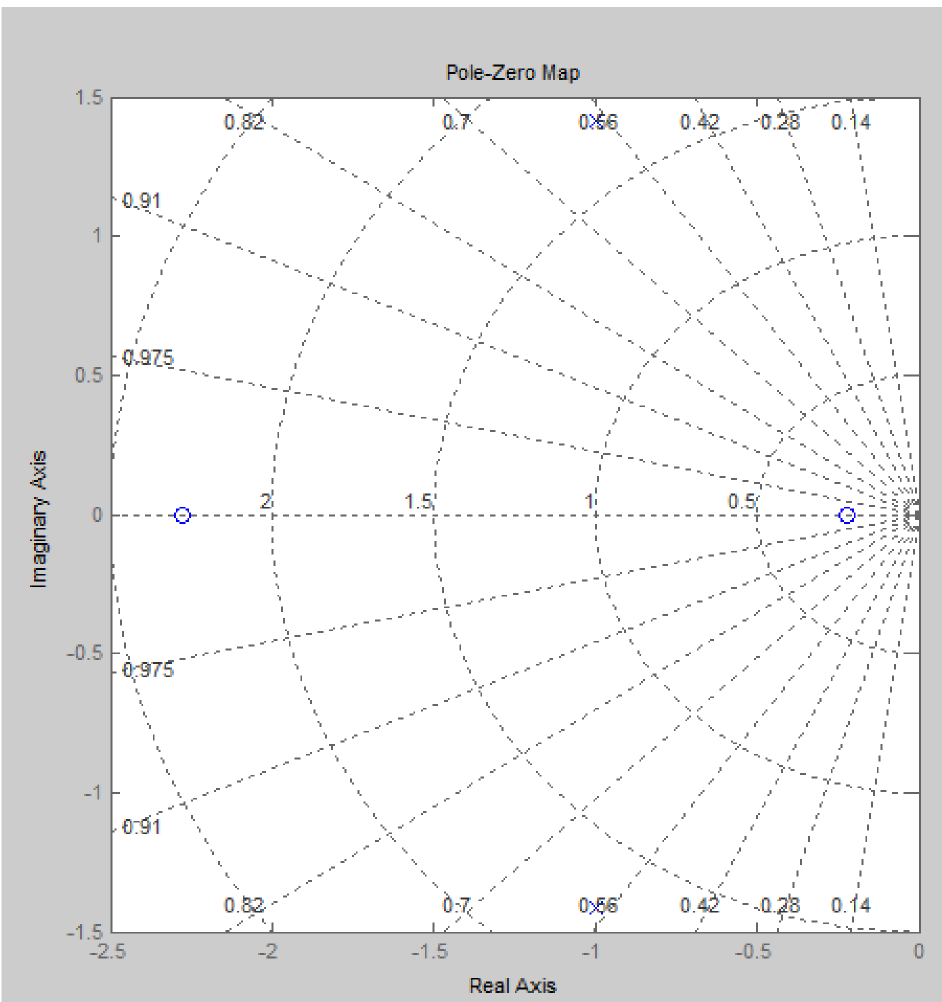
```
>> t = tf(n,d)
```
transfer function

```
Transfer function:
2 s^2 + 5 s + 1
---------------
 s^2 + 2 s + 3

>> pole(t)
```
poles

```
ans =
  -1.0000 + 1.4142i
  -1.0000 - 1.4142i

>> zero(t)
```
zeros

```
ans =

   -2.2808
   -0.2192
>>
>> pzmap(t)
```
pole zero map

```
>> grid
```

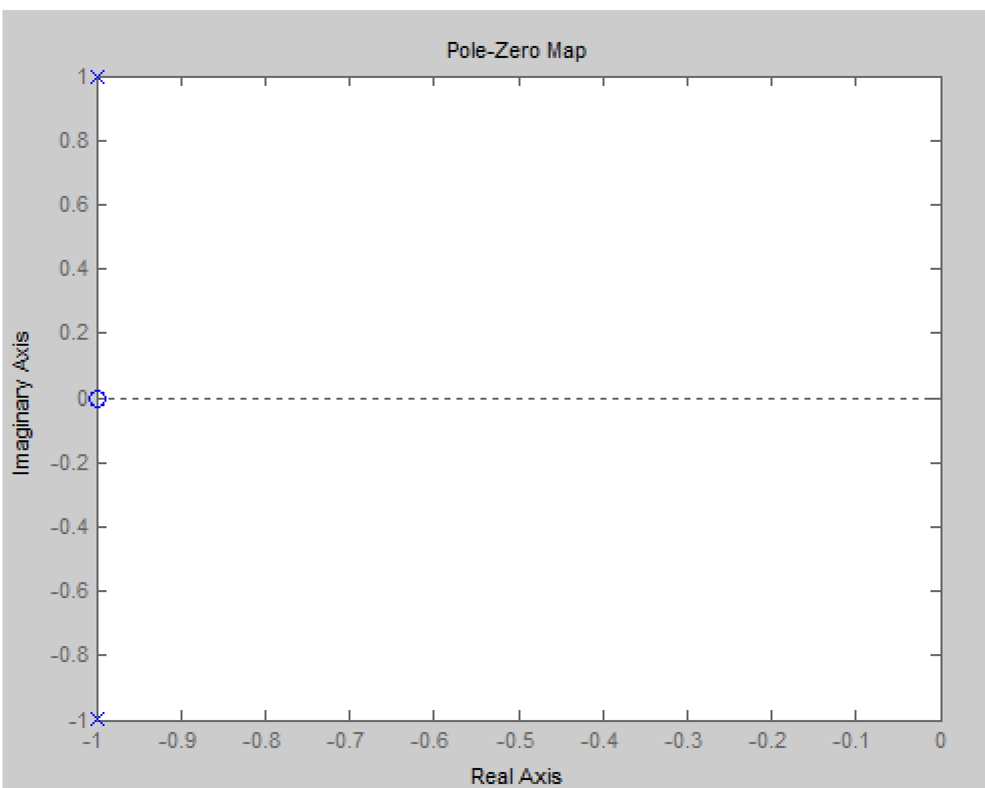


```
>> num = [1 1]

num =

    1    1

>> den = [1 2 2]

den =

    1    2    2

>> H = tf(num, den)
```

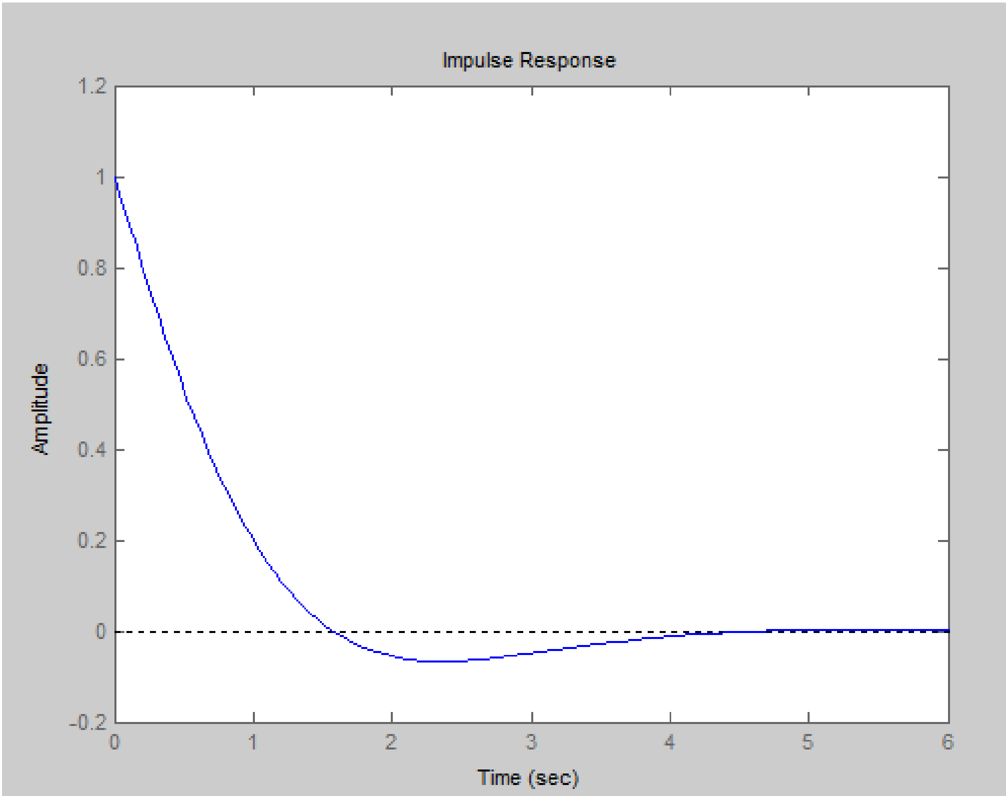
transfer function

```
Transfer function:
   s + 1
-------------
s^2 + 2 s + 2

>> pzmap(H)
```

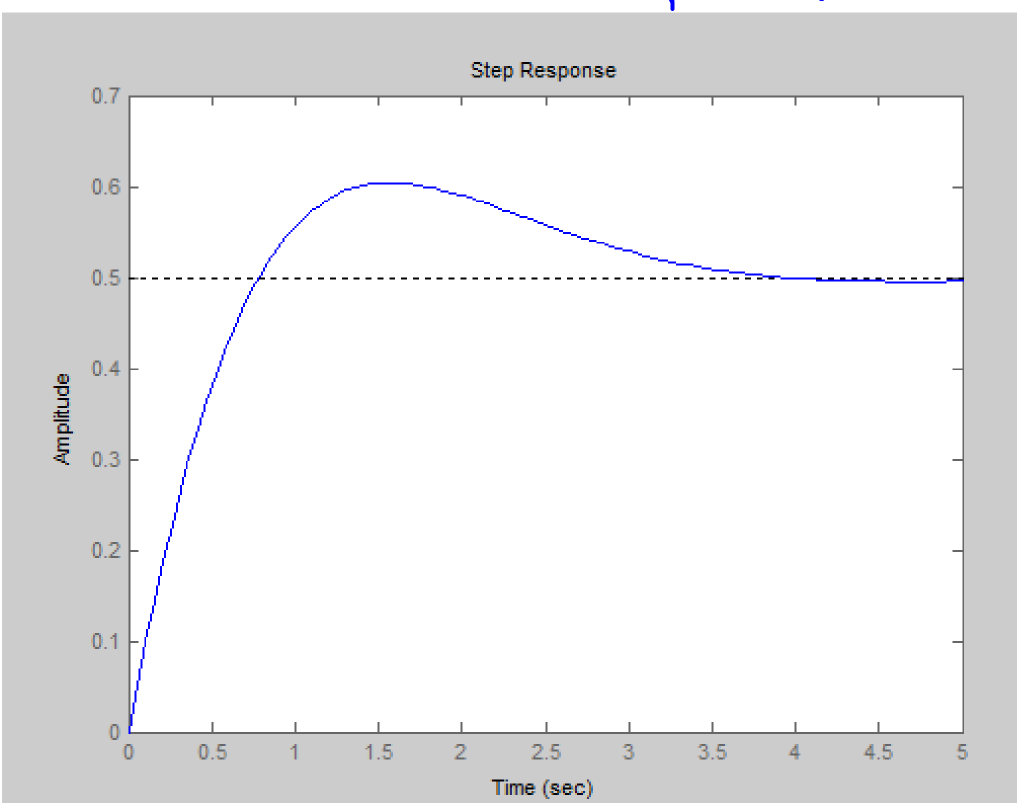
pole zero map



impulse(H) ← Impulse Response



step(H) ← step response



Bode(H) ← Bode Plot