Teoría de Circuitos II

# Trabajo Semanal 3

*Tobias Bavasso Piizzi*

La resolución detallada se puede encontrar aquí

# Ejercicio 1

## Función Transferencia

La Función transferencia buscada responde a la forma

$$T_{(s)} = k \cdot \frac{s-z}{s+p} \tag{1}$$

$z = p = \sigma$

De esta forma lograremos tener un pasa todo y en base a la frecuencia que se utilice se rotará la fase. En otras palabras estamos buscando un ***Filtro Pasa Todo***

Entonces la función transferencia tendrá la forma

$$T_{(s)} = \frac{s-\sigma}{s+\sigma} \tag{2}$$

## Implementación Pasiva

Para lograr una implementación pasiva de la misma se puede utilizar el circuito ***lattice***

$$T_{(s)} = \frac{1}{2}\frac{s-\sigma}{s+\sigma} \tag{3}$$

$$\sigma = \frac{1}{R \cdot C} \tag{1}$$

Si ahora quisieramos diseñar para tener una cierta rotación de fase respecto al comienzo solo habría que realizar el estudio de su comportamiento.

$$\phi(\omega) = \arctan(\frac{-\omega}{\sigma}) - \arctan(\frac{\omega}{\sigma}) \tag{4}$$

$$\phi(\omega = 1) = \frac{\pi}{12} = \arctan(\frac{-1}{\sigma}) - \arctan(\frac{1}{\sigma}) \tag{5}$$

$$\frac{1}{\sigma} = 0.1316 \tag{2}$$

$$R \cdot C = 0.1316 \tag{3}$$

Adopto los valores de la tabla y no considero ninguna normalización en frecuencia ya que en un filtro pasatodo carece de sentido

| Compo | Valor | $\Omega z$ |
|---|---|---|
| R | 0.5 | 50k$\Omega$ |
| C | 0.26 | 2.632 pf |

$$T_{(s)} = \frac{1}{2}\frac{s-7.59}{s+7.59} \tag{6}$$

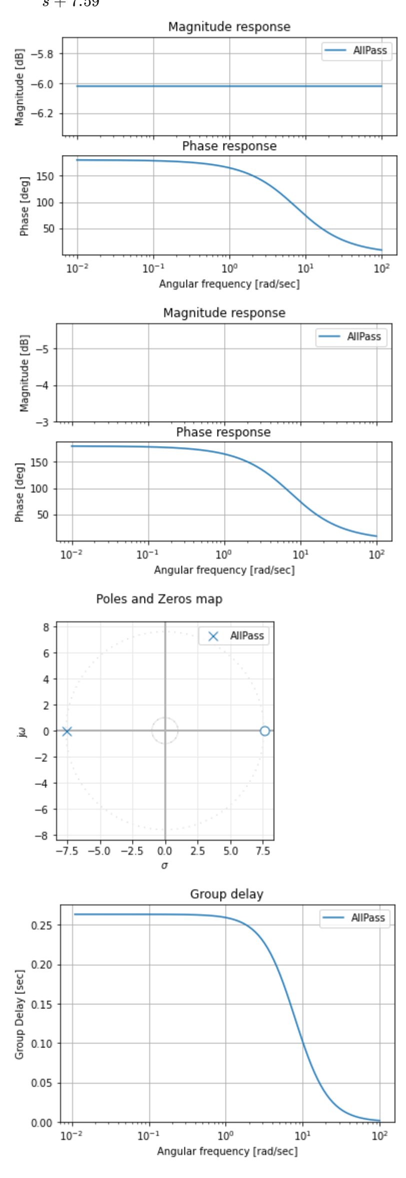
## Simulación Circuital

Con la ayuda de los cursores vemos que para la pulsación angular pedida la fase disminuye 15° como se pretendía

## Simulación matemática

In [1]:
```python
import scipy.signal as sig
import matplotlib as mpl
from splane import analyze_sys, pretty_print_lti
import numpy as np
#%matplotlib qt5

z = [7.59]
p = [-7.59]
k = 0.5

this_lti = sig.ZerosPolesGain(z, p, k).to_tf()

pretty_print_lti(this_lti)

analyze_sys( [this_lti], ["AllPass" ])
```

$$\frac{s\ 0.5 + -3.795}{s + 7.59}$$




## Implementación Activa

Este circuito nos presenta una transferencia que también nos permite realizar un ***Filtro Pasa Todo***

$$T_{(s)} = -k \cdot \frac{s - \frac{1}{C \cdot R2 \cdot k}}{s - \frac{1}{C \cdot R2}} \tag{7}$$

$$k = \frac{RF}{R1} = 1 \tag{4}$$

Vemos que los ***valores que anteriormente utilizamos siguen siendo válidos ya que forzamos al grado de libertad k = 1***.

| Compo | Valor | $\Omega z$ |
|---|---|---|
| R | 0.5 | 50k$\Omega$ |
| C | 0.26 | 2.632 pf |
| Rk | 0.1 | 10k$\Omega$ |

### Simulación Circuital

Nuevamente se observa como la fase se comporta de la misma forma, pero ahora la ***ganancia es unitaria***.

# Ejercicio 2

$$T_{(s)} = k \cdot \frac{s^2 + s \cdot \frac{\omega_n}{Q_n} + \omega_n^2}{s^2 + s \cdot \frac{\omega_p}{Q_p} + \omega_p^2} \tag{8}$$

Sabiendo que el donominador responde a un ***Butter de 2° Orden*** ya puedo conocer el polinomio de la transferecnia

$$q(s) = s^2 + s \cdot \sqrt{2} + 1 \tag{5}$$

Para lograr las respuestas solicitadas es necesario ***colocar correctamente los Z***.

## Notch Normalizado Pasabajo

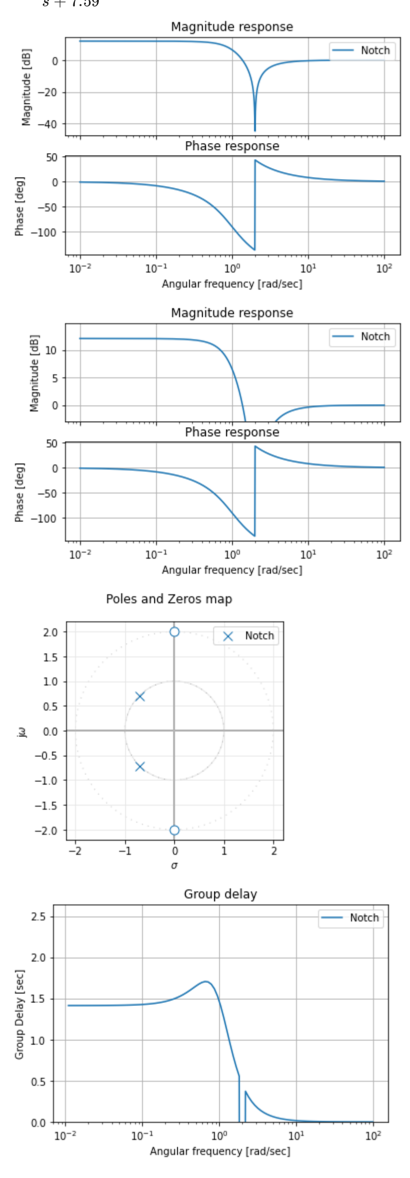
Planteando la siguiente transferecnia podemos obtener la respuesta buscada. Tendremos que colocar un par de ***Z complejos conjugados por encima de la circunferencia unitaria***

$$T_{(s)} = k \cdot \frac{s^2 + 2^2}{s^2 + s \cdot \sqrt{2} + 1} \tag{9}$$

### Simulación Matemática

In [2]:
```python
#n1 = [ 1 , np.sqrt(2)/2 , 1 ]
n2 = [ 1 , 0 , 4]

p = [ 1 , np.sqrt(2) , 1]
k = 1

#tf_be   = sig.TransferFunction(n1,p)
tf_n    = sig.TransferFunction(n2,p)
pretty_print_lti(this_lti)

#analyze_sys( [tf_be], ["BE" ])

analyze_sys( [tf_n], ["Notch" ])
```

$$\frac{s\ 0.5 + -3.795}{s + 7.59}$$




### Implementación

El circuito a realizar debe ser activo si o si porque para bajas frecuencias tenemos ganancia superior a 10 dB.

Este Ackerberg-Mossberg tiene la particularidad que varios componentes están levantados lo cual nos permite lograr una transferecia como la siguiente

$$T_{(s)} = \frac{-a \cdot s^2 + d \cdot \omega_o^2}{s^2 + s \cdot \frac{\omega_o}{Q} + \omega_o^2} \tag{10}$$

Adoptamos las siguientes condiciones

a = 1 d = 4

$$T_{(s)} = -\frac{s^2 + 4}{s^2 + s \cdot \sqrt{2} + 1} \tag{11}$$

$\Omega_w = 2 \cdot \pi \cdot 1KHz \; \Omega_z = 1k\omega$

| Compo | Valor nor | Valor |
|---|---|---|
| R | 1 | 1k$\Omega$ |
| C | 1 | 79.57 nf |

### Simulación Circuital

## Elimina Banda

En este caso se debe analizar la respuesta en módulo del filtro según las especificaciones solicitadas. Obviamente ***todas las singularidades estarán en la circunferencia unitaria***

$$|T|_{\omega=1} = \frac{Q}{Q_z} = 0.5 \tag{12}$$

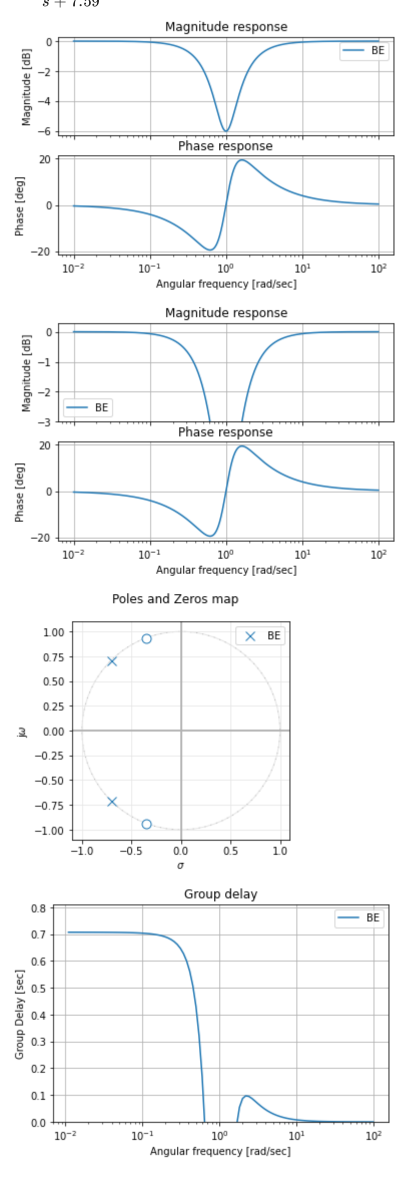
$$|Q_z = \sqrt{2} \tag{6}$$

$$T_{(s)} = -\frac{s^2 + s \cdot \frac{\sqrt{2}}{2} + 1}{s^2 + s \cdot \sqrt{2} + 1} \tag{13}$$

### Simulación Matemática

In [3]:
```python
n1 = [ 1 , np.sqrt(2)/2 , 1 ]
n2 = [ 1 , 0 , 4]

p = [ 1 , np.sqrt(2) , 1]
k = 1

tf_be   = sig.TransferFunction(n1,p)
tf_n    = sig.TransferFunction(n2,p)

pretty_print_lti(this_lti)

analyze_sys( [tf_be], ["BE" ])

#analyze_sys( [tf_n], ["Notch" ])
```

$$\frac{s\ 0.5 + -3.795}{s + 7.59}$$




### Implementaación

Utilizaremos un circuito con ***levantamiento de masa*** para lograr los 0 dB en la banda de paso.

El levantamiento de masa nos permite quitar parte de un componente para lograr una modificación en el numerador de la transferencia y así ajustar los "Z".

$$T_{(s)} = \frac{s^2 + s \cdot \frac{G \cdot a}{C} + \frac{1}{L \cdot C}}{s^2 + s \cdot \frac{G}{C} + \frac{1}{L \cdot C}} \tag{14}$$

Vemos que el parámetro a nos da el grado de libertad que buscamos. en este caso necesitamos a = 2 para sintetizar la transferencia

$\Omega_w = 2 \cdot \pi \cdot 1KHz \; \Omega_z = 2k\omega$

| Compo | Valor nor | Valor |
|---|---|---|
| R | $\frac{1}{\sqrt{2}}$ | 1.414k$\Omega$ |
| C | 1 | 79.57 nf |
| L | 1 | 318.3 mH |

### Simulación Circuital

# Ejercicio 3

\begin{equation}
\label{eqn:n14}\tag{14}
\Theta_{(\omega)} = \frac{\pi}{2} - \arctan(\frac{6\omega}{-\omega^{2} + 4})
\end{equation}

Por simple inspección vemos que la fase responde a un ***Filtro Pasa Banda ya que arranca desde +90°***

Vamos a trabajarla como un Filtro Pasa Bajos y luego le impondremos la singularidad en el origen.

$$\Theta 2_{(\omega)} = -\arctan(\frac{6\omega}{-\omega^2 + 4}) \tag{15}$$

$$\frac{F_{j\omega}}{F_{-j\omega}} = \frac{1 + j\tan(\Theta 2_{\omega})}{1 + j\tan(\Theta 2_{\omega})} \tag{16}$$

$$\frac{F_{j\omega}}{F_{-j\omega}} = \frac{1 - j\frac{6\omega}{-\omega^2+4}}{1 - j\frac{6\omega}{-\omega^2+4}} \tag{17}$$

$$\frac{F_s}{F_{-s}} = \frac{s^2 - s \cdot 6 + 4}{s^2 + s \cdot 6 + 4} \tag{18}$$

Ahora nos toca elegir las singularidades que representen un ***sistema estable***, es decir que sus polos estén en el semiplano izquierdo.

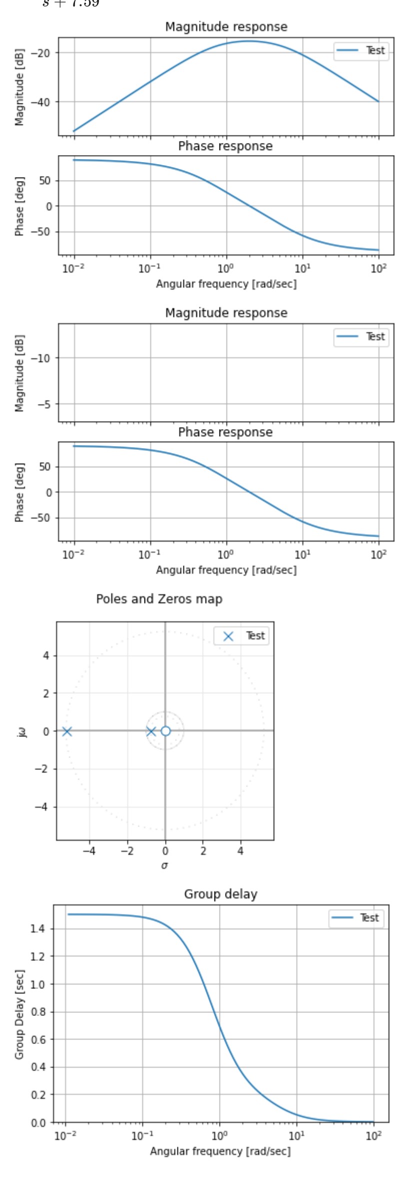
$$Flp_s = \frac{1}{s^2 + s \cdot 6 + 4} \tag{19}$$

$$F_s = \frac{s}{s^2 + s \cdot 6 + 4} \tag{20}$$

## Simulación Matemática

In [4]:
```python
num = [ 1 , 0 ]
den = [ 1 , 6 , 4]


tf   = sig.TransferFunction(num,den)

pretty_print_lti(this_lti)

analyze_sys( [tf], ["Test" ])
```

$$\frac{s\ 0.5 + -3.795}{s + 7.59}$$




## Implementación

$$F_s = \frac{s}{s^2 + s \cdot 6 + 4} \tag{21}$$

Vamos a implementarlo con el siguiente filtro cuya transferencia normalizada es

$$F_s = \frac{G}{C} \cdot \frac{s \cdot \frac{G}{C}}{s^2 + s \cdot \frac{G}{C} + 1} \tag{22}$$

$\Omega_w = 2 \cdot \pi \cdot 1KHz \; \Omega_z = 1k\omega$

| Compo | Valor nor | Valor |
|---|---|---|
| R | $\frac{1}{3}$ | 333$\Omega$ |
| C | 1 | 159.15 nf |
| L | 1 | 159.15 mH |

## SImulación Circuital