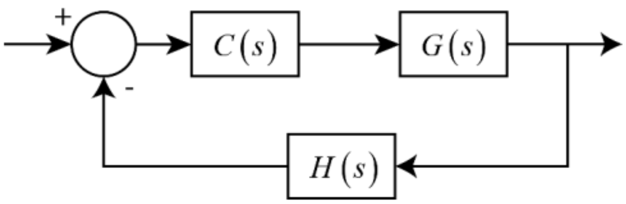
Práctica 4 - Ejercicio 2

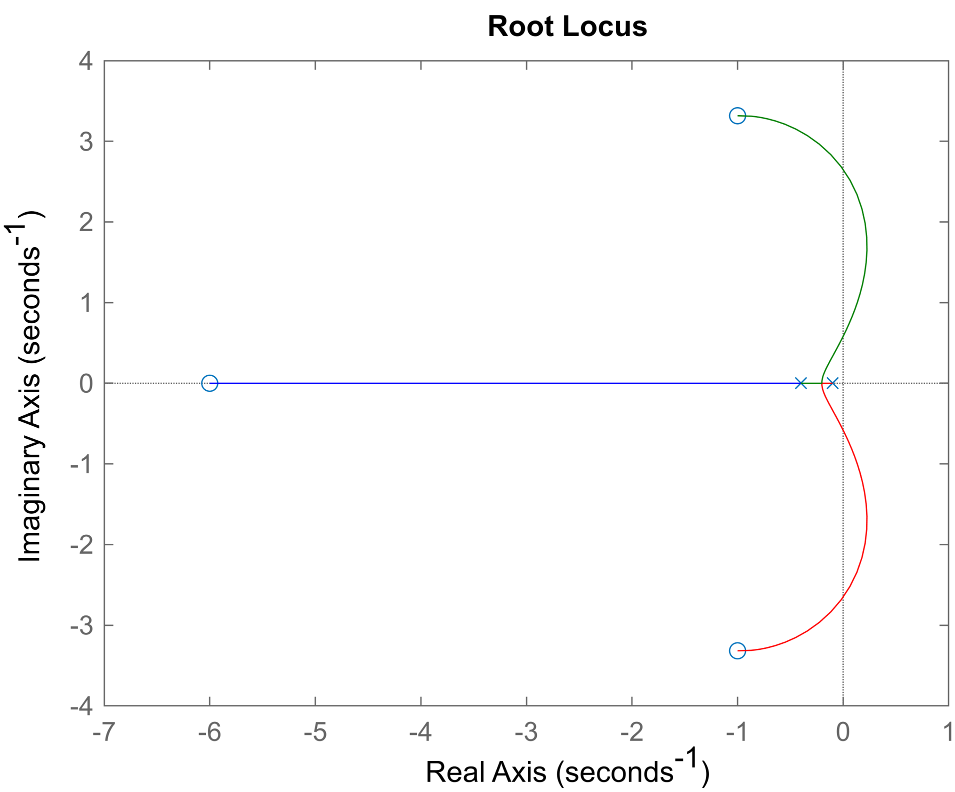
Dado el sistema representado por el diagrama de bloques de la figura, en el que el controlador es un control P:



Donde: $C(s) = K$, $G(s) = \frac{s^2+2s+12}{(s+0.4)^2}$, $H(s) = \frac{s+6}{s+0.1}$

- **Represente el lugar de las raíces del sistema (0.5 ptos)**

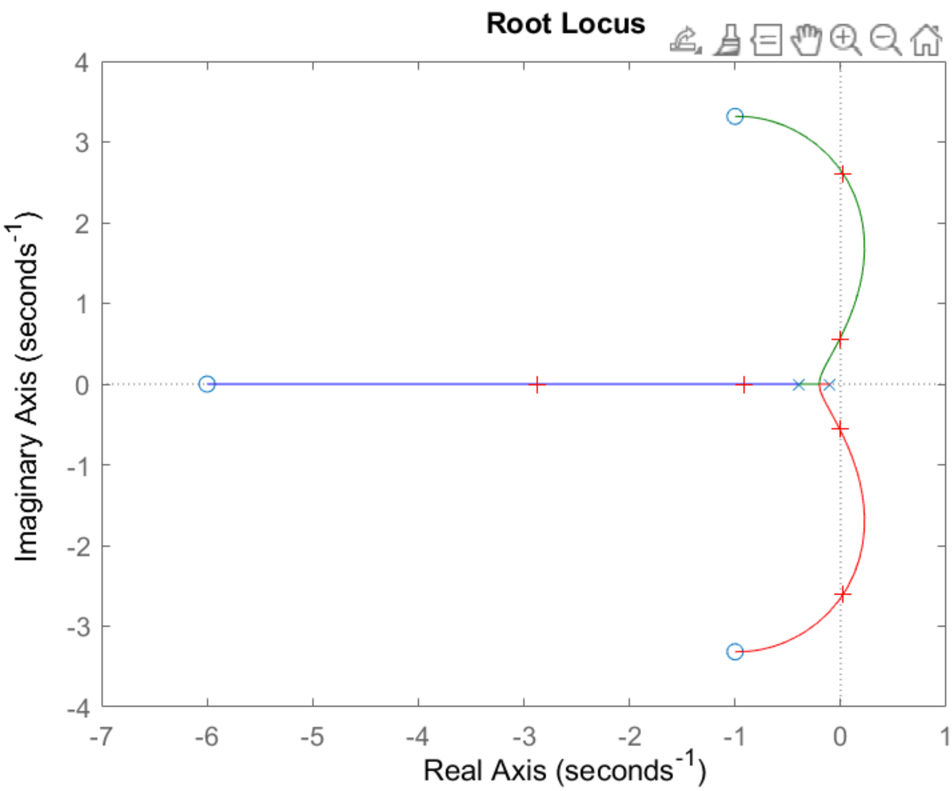
```
numG=[1 2 12];
denG=conv([1 0.4], [1 0.4]);
G=tf(numG, denG);
H=tf([1 6], [1 0.1]);

rlocus(G*H)
```



- **¿Existe algún rango de valores para el controlador que hacen que el sistema se vuelva inestable? (0.75 ptos)**

```
[K_inestable1]=rlocfind(G*H)
```

```
Select a point in the graphics window
```




```
selected_point = 0.0079 + 2.6033i
K_inestable1 = 0.3726
```

```
[K_inestable2]=rlocfind(G*H)
```

```
Select a point in the graphics window
selected_point = 0.0079 + 0.5537i
K_inestable2 = 0.0038
```

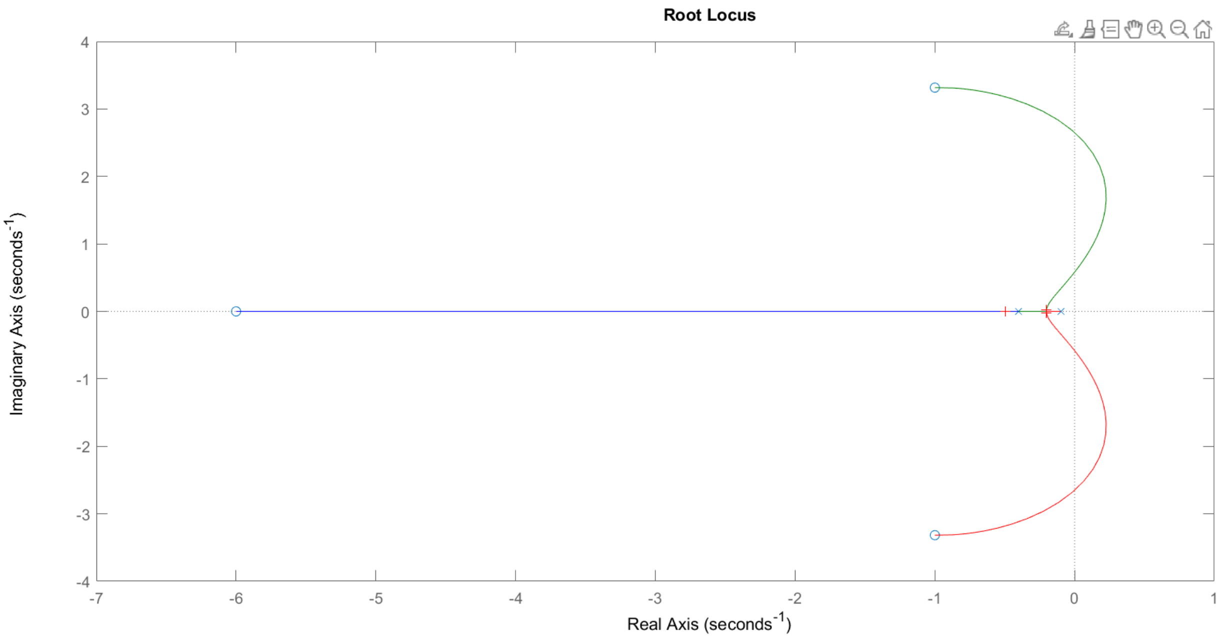
El sistema se vuelve inestable para valores de K comprendidos entre 0.0035 y 0.3934

- **¿Cuál es el controlador que consigue la respuesta más rápida posible sin que llegue a oscilar? ¿cuál es el tiempo de subida, de asentamiento, el máximo sobreimpulso y el error en estado estacionario de dicho sistema? (1.25 ptos)**

El sistema más rápido posible que no oscila sería el criticamente amortiguado, aquel que cuenta con dos polos reales e iguales.

```
[K_critAmortiguado]=rlocfind(G*H)
```

```
Select a point in the graphics window
```




```
selected_point = -0.1970 - 0.0229i
K_critAmortiguado = 6.1552e-05
```

```
G_LC=feedback(K_critAmortiguado*G, H);
stepinfo(G_LC)
```

```
ans = struct with fields:
        RiseTime: 5.2239
   TransientTime: 29.1500
    SettlingTime: 29.0405
     SettlingMin: 0.0033
     SettlingMax: 0.0041
       Overshoot: 12.7565
      Undershoot: 0
            Peak: 0.0041
        PeakTime: 12.5797
```

Sin embargo, debido a la presencia de los ceros no despreciables, el comportamiento del sistema no se corresponde con uno de segundo orden puro. Por este motivo, el sistema presenta oscilaciones (12% de sobrepico) aun cuando sus polos son reales.

```
[Y]=step(G_LC);
error_estacionario=(1-Y(end))*100 %Error en porcentaje
```

```
error_estacionario = 99.6383
```

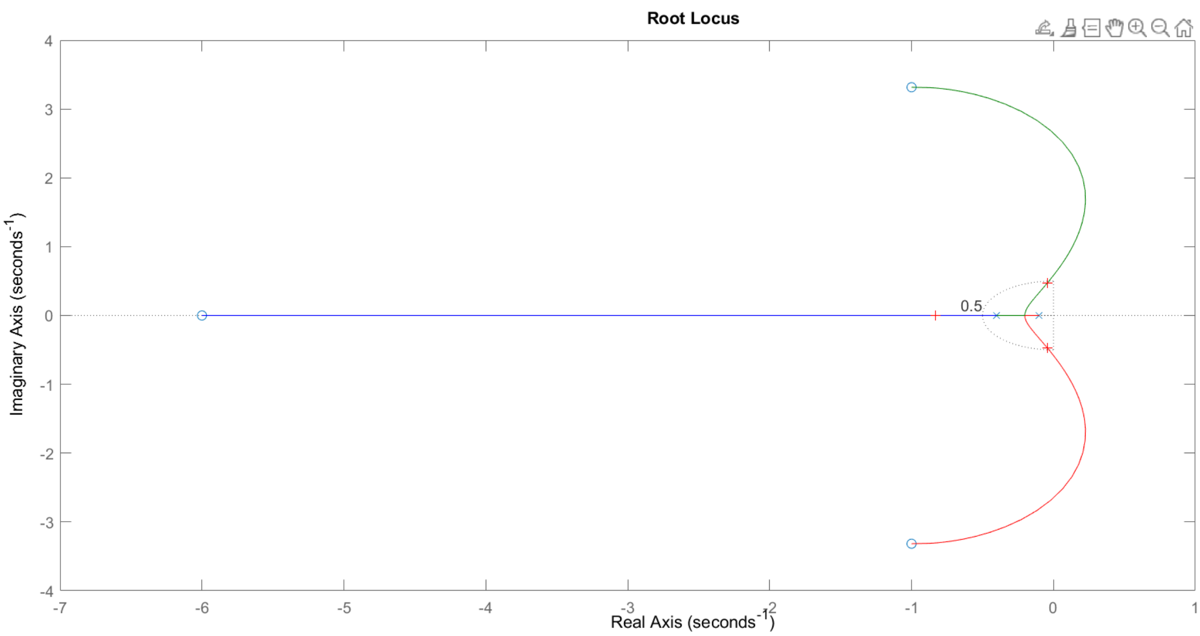
- **Se quiere que el sistema se comporte como un sistema subamortiguado con una frecuencia natural de wn = 0.5 rad/s . ¿Qué controlador consigue dicho comportamiento? ¿Es posible**

aproximar el sistema por uno de orden menor? Justifique su respuesta y, de ser posible, compruebe el resultado (1.5 ptos)

En primer lugar se dibuja la posición de todos los puntos del plano imaginario que cumple con el requisito de wn=0.5rad/s.

```
rlocus(G*H)
sgrid([], 0.5)
[k_wn]=rlocfind(G*H)
```

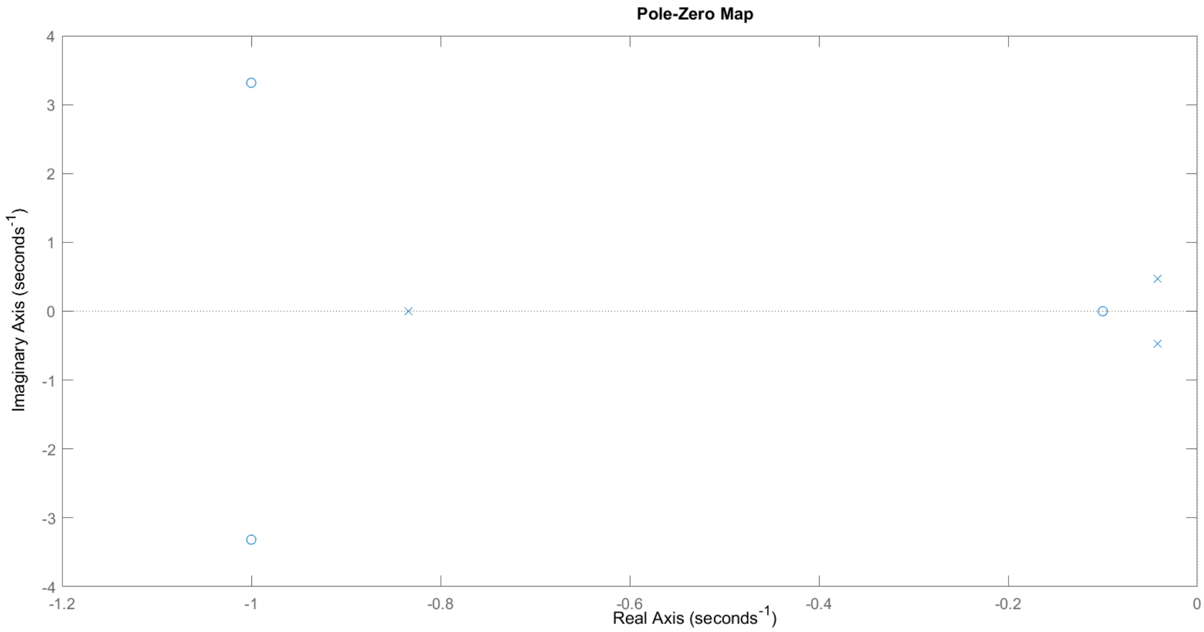
```
Select a point in the graphics window
```




```
selected_point = -0.0313 + 0.4712i
k_wn = 0.0024
```

Una vez conocido este valor de K, se va a analizar los y polos y ceros en lazo cerrado del sistema, para comprobar si es posible obtener un sistema equivalente de orden menor

```
G_LC_wn=feedback(k_wn*G,H);
pzmap(G_LC_wn)
```




```
polos_LC_wn=pole(G_LC_wn)
```

```
polos_LC_wn = 3×1 complex
  -0.8338 + 0.0000i
  -0.0417 + 0.4756i
  -0.0417 - 0.4756i
```

```
zeros_LC_wn=zero(G_LC_wn)
```

```
zeros_LC_wn = 3×1 complex
  -1.0000 + 3.3166i
  -1.0000 - 3.3166i
  -0.1000 + 0.0000i
```

Teniendo en cuenta únicamente los polos y ceros dominantes, es posible simplificar el sistema por otro de dos polos y un cero:

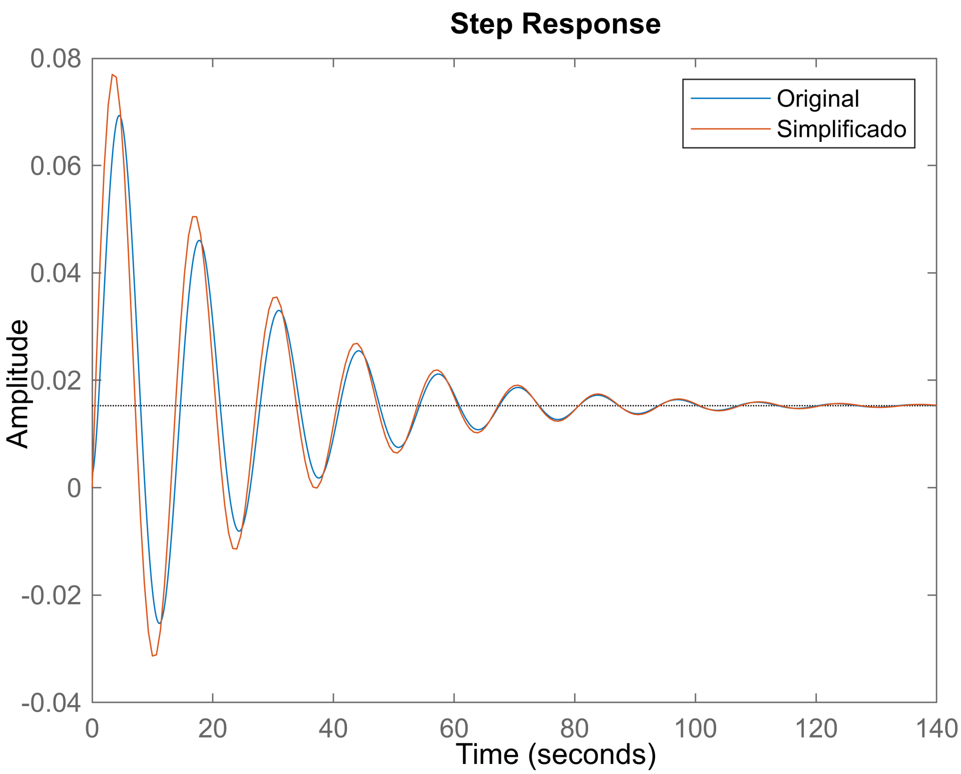
```
G_LC_wn_simplificado=zpk(-0.1, [-0.0417+0.47i, -0.0417-0.47i], 1)
```

```
G_LC_wn_simplificado =

         (s+0.1)
  ------------------------
  (s^2 + 0.0834s + 0.2226)

Continuous-time zero/pole/gain model.
```

Ajusto la ganancia de los sistemas para que ambos tengan la misma ganancia estática.

```
correccionGanancia=evalfr(G_LC_wn, 0)/evalfr(G_LC_wn_simplificado, 0);
G_LC_wn_simplificado=G_LC_wn_simplificado*correccionGanancia;
step(G_LC_wn, G_LC_wn_simplificado)
```

```
legend('Original', 'Simplificado')
```



Como se puede ver en la gráfica, el comportamiento del sistema original y el simplificado es equivalente