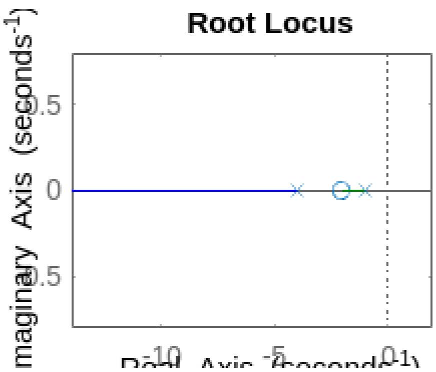
Guión de la Práctica 4 Fundamentos de la Automática:

1. Gráfica del lugar de las raíces (LDR) con MATLAB

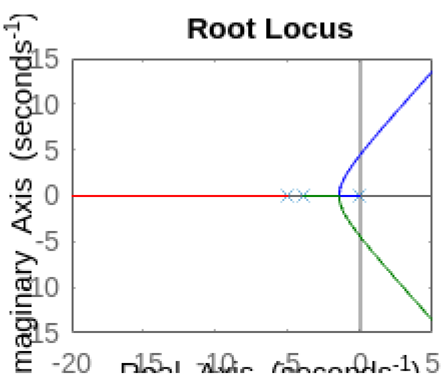
Ejercicio práctico 1: Dibujo básico del lugar de las raíces con MATLAB. Extracción de K

1. Crea un script de MATLAB y dibuja el lugar de las raíces de al menos 6 sistemas de control extraídos de los ejercicios 4.2 al 4.13 de la relación de ejercicios del Tema 4.

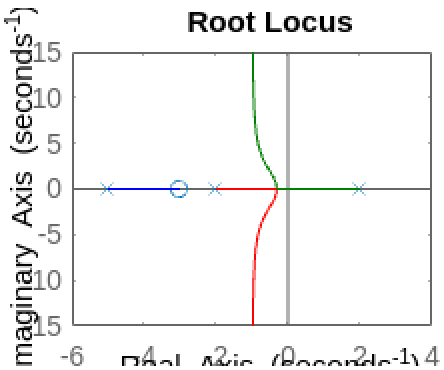
```
% Ejercicio práctico 4.2
G1 = tf([1 2],[1 4]);
H1 = tf(1,[1 1]);
rlocus(G1*H1)
```



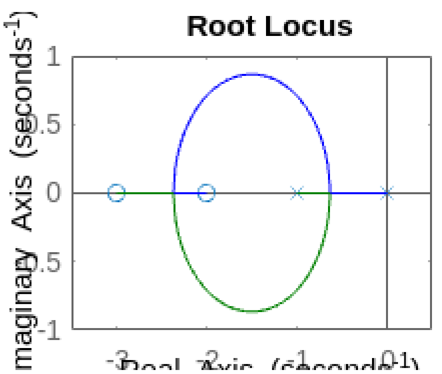

```
% Ejercicio práctico 4.3
G2 = tf(1,[1 4 0]);
H2 = tf(1,[1 5]);
rlocus(G2*H2)
```



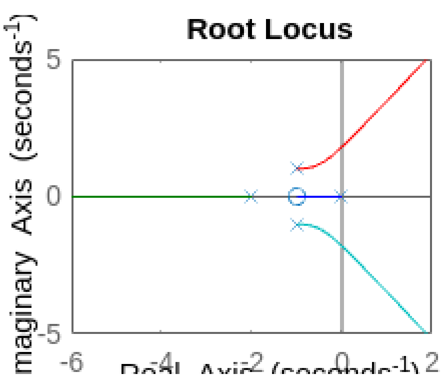

```
% Ejercicio práctico 4.4
G3 = tf(1,[1 0 -4]);
H3 = tf([1 3],[1 5]);
rlocus(G3*H3)
```



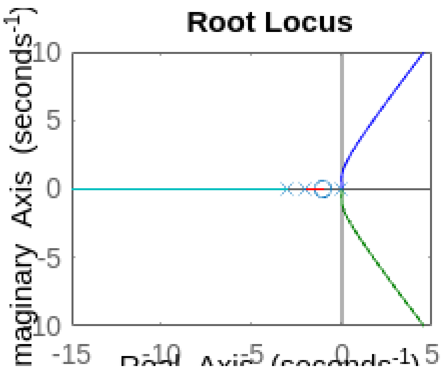

```
% Ejercicio práctico 4.5
G4 = tf([1 2],[1 0]);
H4 = tf([1 3],[1 1]);
rlocus(G4*H4)
```



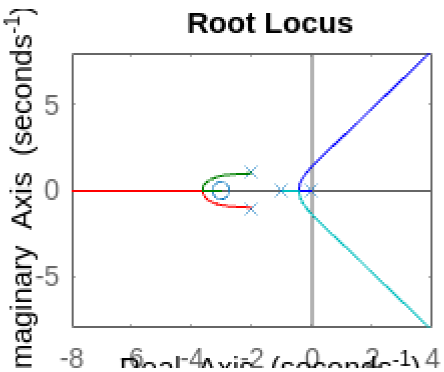

```
% Ejercicio práctico 4.6
G5 = tf([1 1],conv([1 2 0],[1 2 2]));
H5 = 1;
rlocus(G5*H5)
```



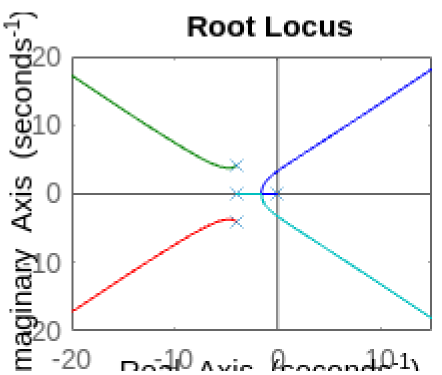

```
% Ejercicio práctico 4.7
G6 = tf([1 1],conv([1 2 0 0],[1 3]));
H6 = 1;
rlocus(G6*H6)
```



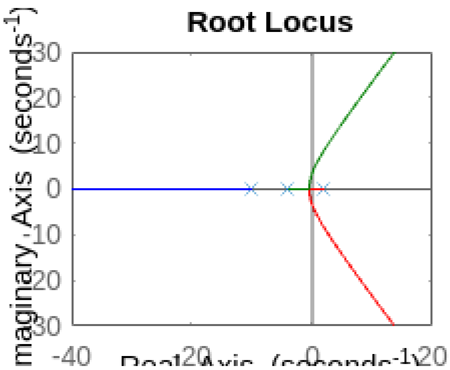

```
% Ejercicio práctico 4.8
G7 = tf([1 3],conv([1 1 0],[1 4 5]));
H7 = 1;
rlocus(G7*H7)
```



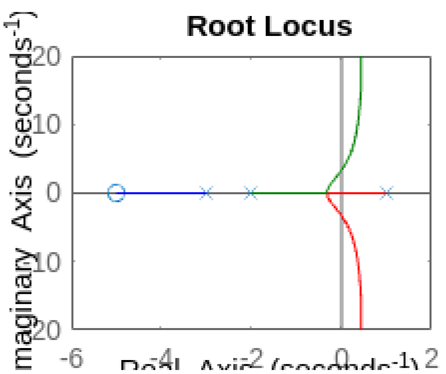

```
% Ejercicio práctico 4.9
G8 = tf(1,conv([1 4 0],[1 8 32]));
H8 = 1;
rlocus(G8*H8)
```



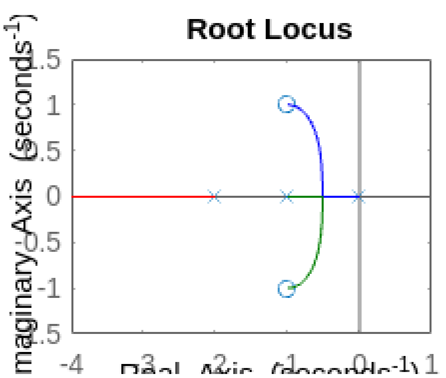

```
% Ejercicio práctico 4.10
G9 = tf(1,[1 8 -20]);
H9 = tf(1,[1 4]);
rlocus(G9*H9)
```



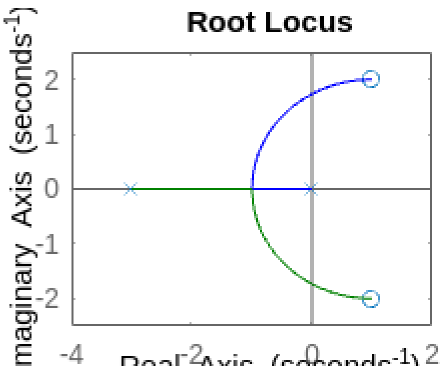

```
% Ejercicio práctico 4.11
G10 = tf(1,[1 1 -2]);
H10 = tf([1 5],[1 3]);
rlocus(G10*H10)
```




```
% Ejercicio práctico 4.12
G11 = tf([1 2 2],[1 2]);
H11 = tf(1,[1 1 0]);
rlocus(G11*H11)
```




```
% Ejercicio práctico 4.13
G12 = tf([1 -2 5],[1 3 0]);
H12 = 1;
rlocus(G12*H12)
```

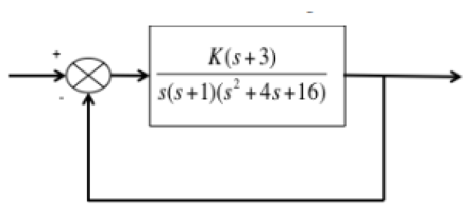


2. Para todas las figuras anteriores, determina gráficamente el punto del LDR en el que la ganancia toma el valor de K =1.

3. Si en la gráfica aparecen puntos de ruptura o reencuentro, halla el valor de K asociado a los mismos. Comprueba alguno de ellos realizando los cálculos manualmente.

<u>Ejercicio práctico 2: Lugar de las raíces con MATLAB para un vector de valores de K.</u>

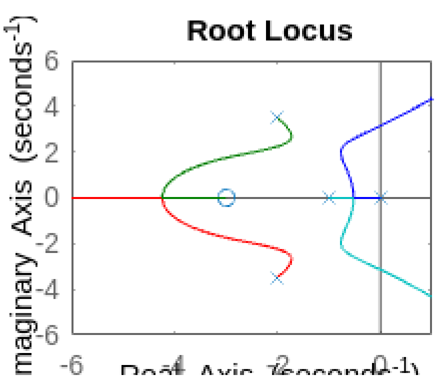
1. Dibuja la gráfica del LDR del sistema de la figura:




```
G = tf([1 3],conv([1 1 0],[1 4 16]));
H = 1;
rlocus(G*H)
```

2. Modifica los ejes de la gráfica para visualizarla entre -6 y 1 (en el eje real) y -6 y 6 (en el eje imaginario).

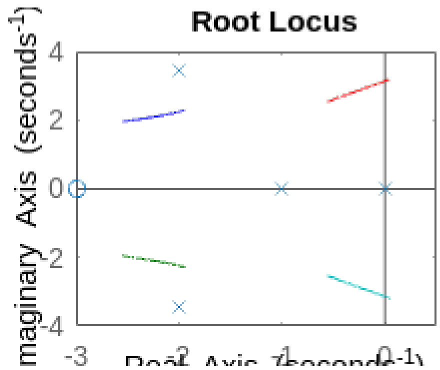
```
axis([-6 1 -6 6])
```



3. Calcula el valor de K para el que el sistema sea hace inestable y el valor de la frecuencia de oscilación de la respuesta del sistema para esa K en el límite de la estabilidad.

```
K = 20:0.01:35;
```

```
rlocus(G*H,K)
```



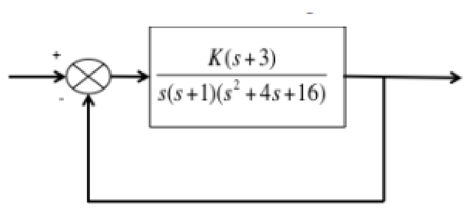

```
polosLC = rlocus(G*H,33)
```

```
polosLC = 4×1 complex
  -2.4902 + 1.9754i
  -2.4902 - 1.9754i
  -0.0098 + 3.1303i
  -0.0098 - 3.1303i
```

1.1 Cálculo gráfico de todos los polos en lazo cerrado mediante "rlocfind"

Ejercicio práctico 3: Uso de rlocfind para ampliar las capacidades de rlocus en MATLAB

Para el sistema de control en lazo cerrado del ejercicio anterior:



1. Encuentra gráficamente la posición de todos los polos en lazo cerrado del sistema para el límite de la estabilidad (sistema oscilante o no amortiguado).

```
G = tf([1 3],conv([1 1 0],[1 4 16]));
H = 1;
rlocus(G*H)
```

2. Calcula mediante rlocfind el valor de K para el que uno de los polos está en s = -5. Indica si el sistema de control será estable en ese caso, razonando tu respuesta.

```
[K,PolosLC] = rlocfind(G*H,-5)
```

2. Diseño de sistemas mediante el Lugar de las Raíces

2.1 Lugar de las raíces con ζ constante y ω_n constante

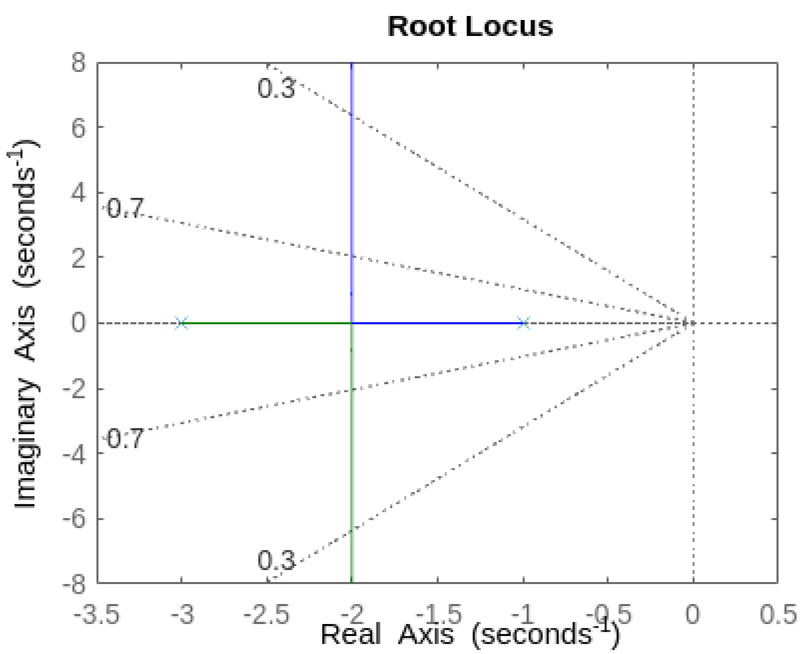
Ejercicio práctico 4: Uso de sgrid para dibujar líneas de coeficiente de amortiguamiento constante

1. Construye un sistema de control en lazo cerrado con H(s) = 1 y donde la función de transferencia en lazo abierto G(s) tenga dos polos situados en s = -1 y s = -3, y ningún cero.

```
G = zpk([],[-1 -3],1);
H=1;
rlocus(G*H)
```

2. Superpón al LDR las líneas de coeficientes de amortiguamiento $\zeta = 0.3$ y $\zeta = 0.7$ y calcula el valor de K (controlador de tipo proporcional) que permite que los polos en lazo cerrado se correspondan con dichos coeficientes. Reajusta los límites de la gráfica si es necesario.

```
chi = [0.3 0.7];
sgrid(chi,[])
ylim([-8 8])
```



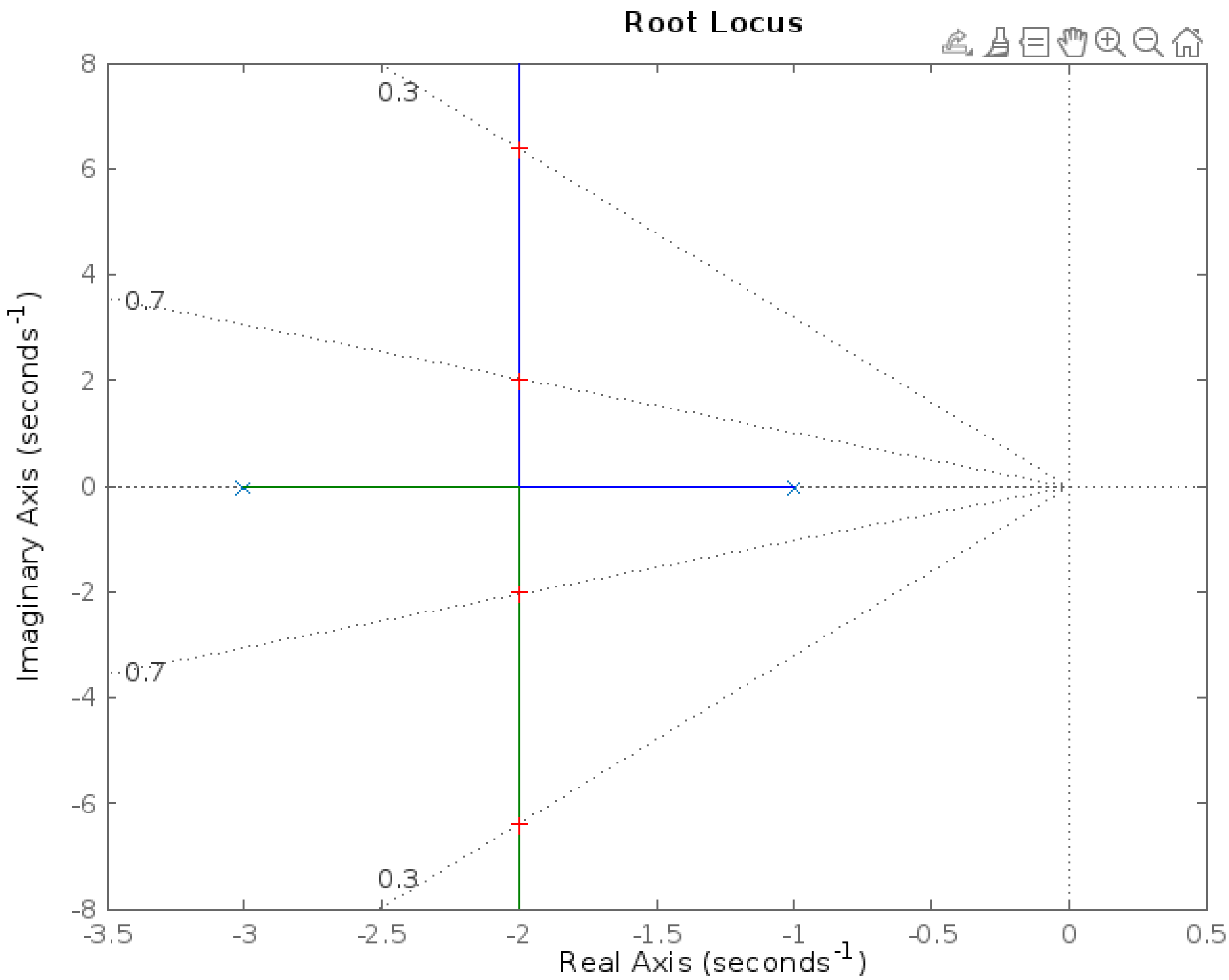
3. Extrae del LDR la posición de los polos en lazo cerrado (LC) para cada valor de K y la sobreelongación que tendrá la respuesta al escalón del sistema. Calcula el coeficiente de amortiguamiento (manualmente) utilizando la posición de los polos en LC, y comprueba con la expresión de Mp la estimación de sobreelongación extraída de la gráfica del LDR.

```
k_chi03 = rlocfind(G*H)
```

```
Select a point in the graphics window
selected_point = -2.0052 + 6.3891i
k_chi03 = 41.8211
```

```
k_chi07 = rlocfind(G*H)
```

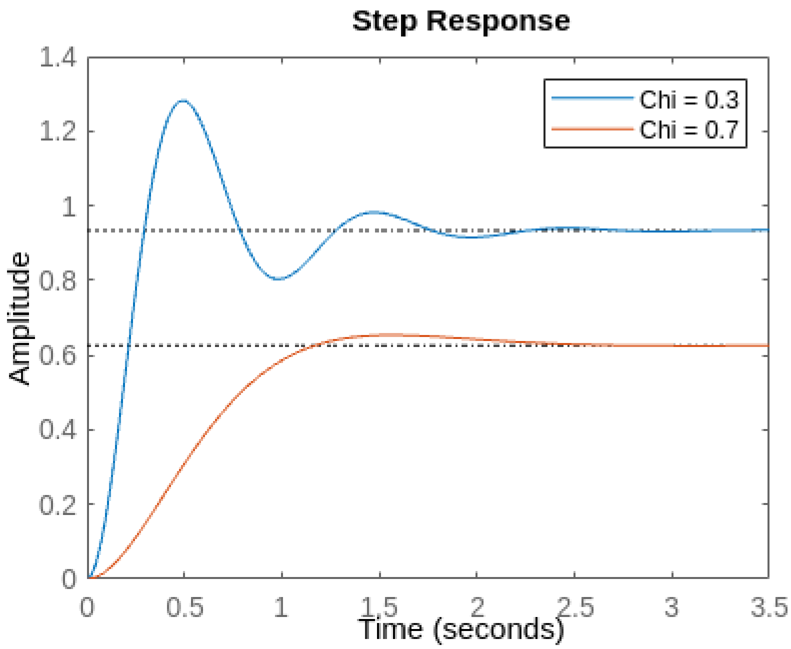
```
Select a point in the graphics window
```




```
selected_point = -1.9983 + 2.0090i
k_chi07 = 5.0363
```

4. Conocida la posición de los polos en LC para cada valor de K, estima el valor del tiempo de asentamiento del sistema.

```
G = zpk([],[-1 -3],1);
H=1;
step(feedback(k_chi03*G,H),feedback(k_chi07*G,H))
legend('Chi = 0.3','Chi = 0.7')
```



5. Realiza una gráfica de la respuesta al escalón del sistema en LC con ambos valores de K y comprueba en ambos casos los valores de sobreelongación y tiempo de asentamiento estimados mediante el LDR (recuerda, para la banda del 1.8%). ¿Coinciden?

6. ¿Con qué valor de K el sistema presenta un mayor error frente a la entrada escalón? ¿Por qué?

Ejercicio práctico 5: Uso de sgrid para dibujar circunferencias de ω_n constante

1. Construye ahora un sistema de control en lazo cerrado similar al del ejercicio anterior pero con G(s) teniendo dos polos situados en s = 0 y s = -3, y ningún cero.

```
G = zpk([],[0 -3],1);
H = 1;
rlocus(G*H)
```

2. Repite el ejercicio anterior y realiza una nueva gráfica del LDR para el mismo sistema, calculando ahora los tres valores de K que nos permiten intersectar con las circunferencias de las tres frecuencias naturales siguientes: ω_n=1, ω_n=1,5 y ω_n=3.

```
wn = [1 1.5 3];
sgrid([],wn)
ylim([-5 5])
```

3. ¿Qué tipo de respuesta transitoria es esperable para cada uno de los valores de K? ¿Cuál de los tres será el sistema con un transitorio más largo? ¿Por qué?

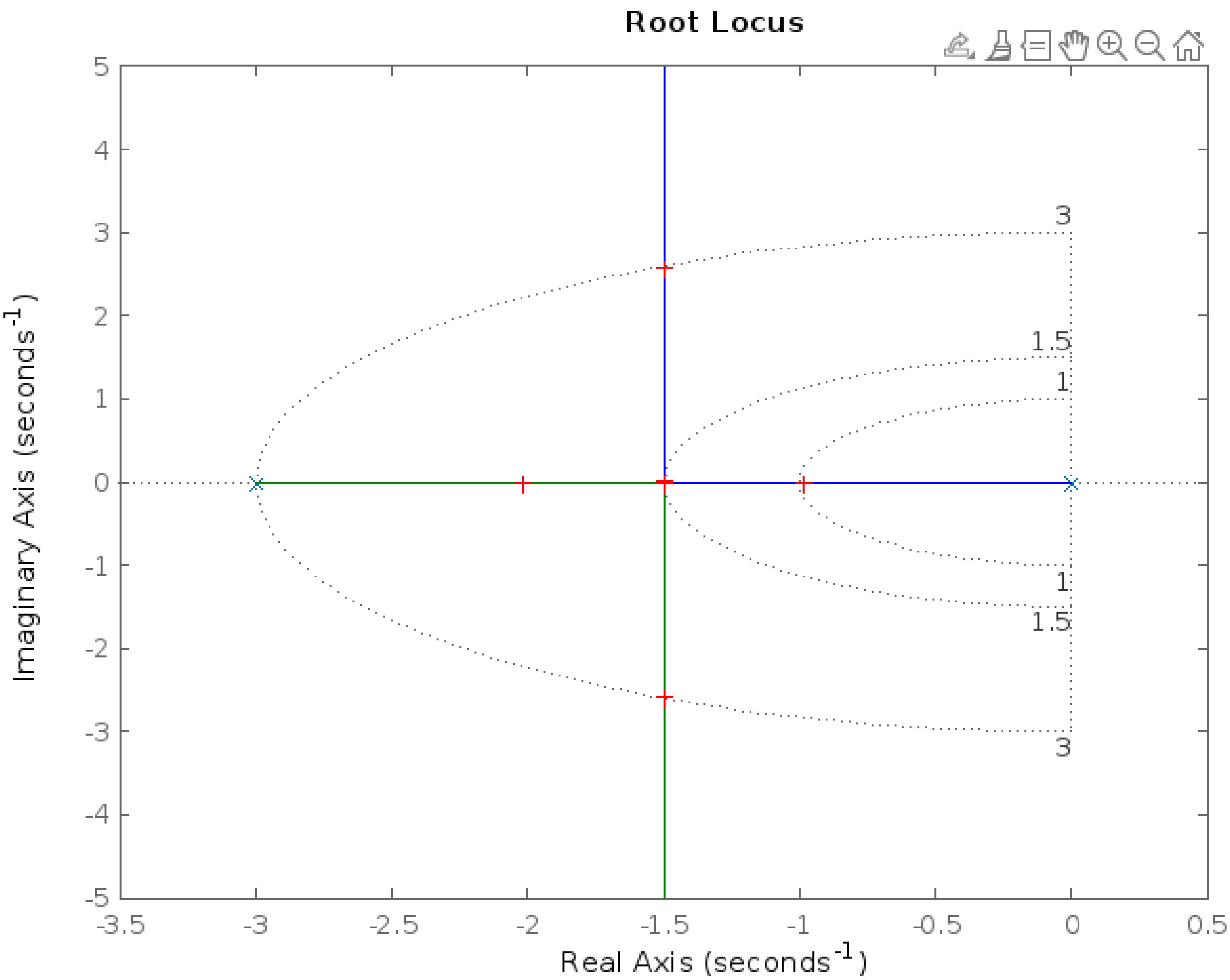
```
k_wn1 = rlocfind(G*H)
```

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

```
selected_point = -0.9842 + 0.0115i
k_wn1 = 1.9841
```

```
k_wn15 = rlocfind(G*H)
```

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



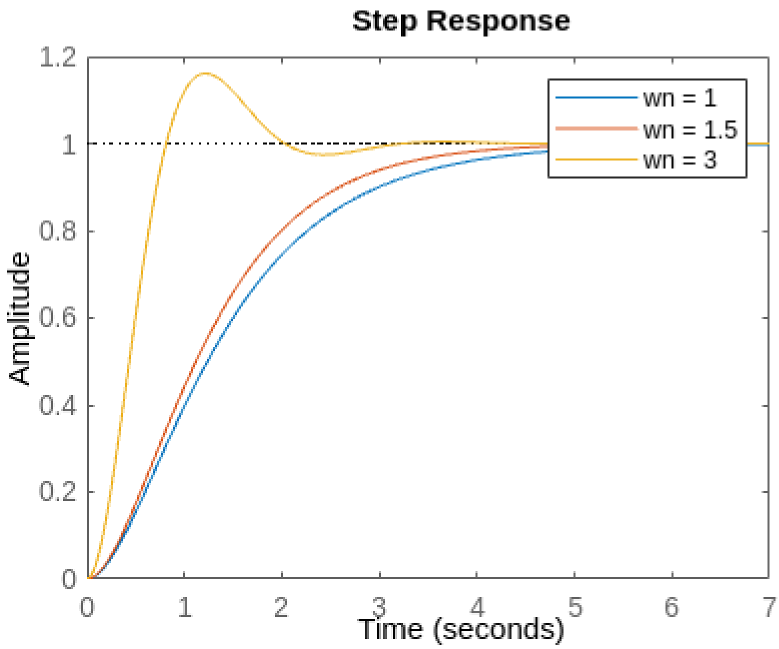

```
selected_point = -1.4895 + 0.0115i
k_wn15 = 2.2500
```

```
k_wn3 = rlocfind(G*H)
```

```
Select a point in the graphics window
selected_point = -1.5105 + 2.5803i
k_wn3 = 8.9079
```

4. Con respecto a la respuesta estacionaria, ¿alguno de los tres valores de K hará que el sistema sea inestable? ¿Se puede prever el error frente a la entrada escalón simplemente observando la gráfica del LDR?

```
step(feedback(k_wn1*G,H),feedback(k_wn15*G,H),feedback(k_wn3*G,H))
axis([0 7 0 1.2])
legend('wn = 1','wn = 1.5','wn = 3')
```



5. Realiza en una misma gráfica la respuesta al escalón del sistema en LC para los tres valores de K hallados y comprueba las respuestas de los apartados 3 y 4.

2.2 Diseño de sistemas de control mediante el LDR

Ejercicio práctico 6: Diseño de sistemas con ceros mediante el LDR

1. Construye un sistema de control en lazo cerrado donde la función en lazo abierto G(s) tenga dos polos situados en s = 0 y s = -1, y un cero situado en s = -3.

```
G = zpk(-3,[0 -1],1);
H = 1;
```

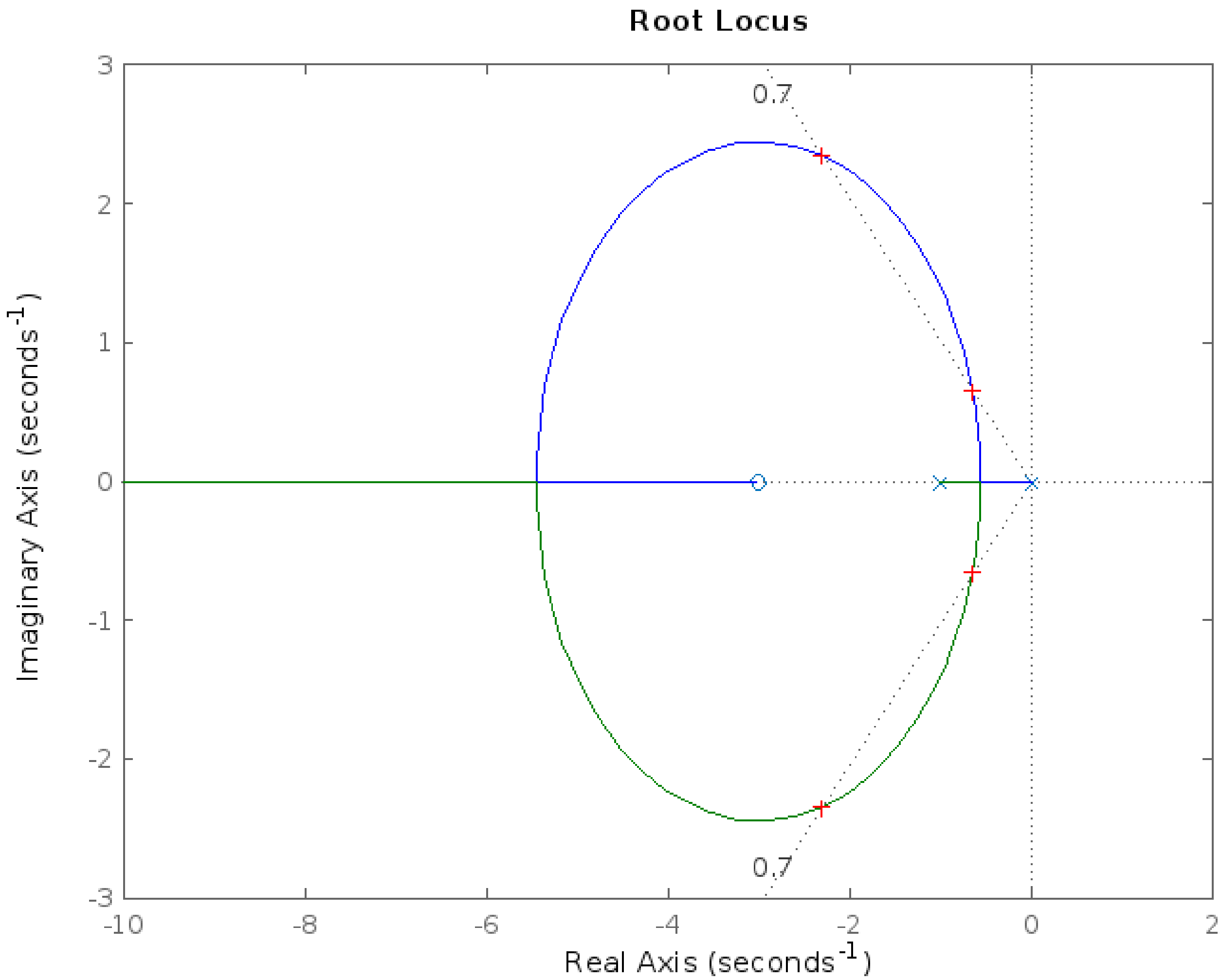
2. Dibuja la gráfica del lugar de las raíces del sistema. Extrae los valores de K que permiten intersectar con la línea de coeficiente de amortiguamiento constante $\zeta = 0.7$. Anota en cada caso la posición de los polos en LC y la sobreelongación máxima asociada con dicho coeficiente de amortiguamiento.

```
rlocus(G*H)
sgrid(0.7,[])
k1 = rlocfind(G*H)
```

```
Select a point in the graphics window
selected_point = -0.6421 + 0.6537i
k1 = 0.2791
```

```
k2 = rlocfind(G*H)
```

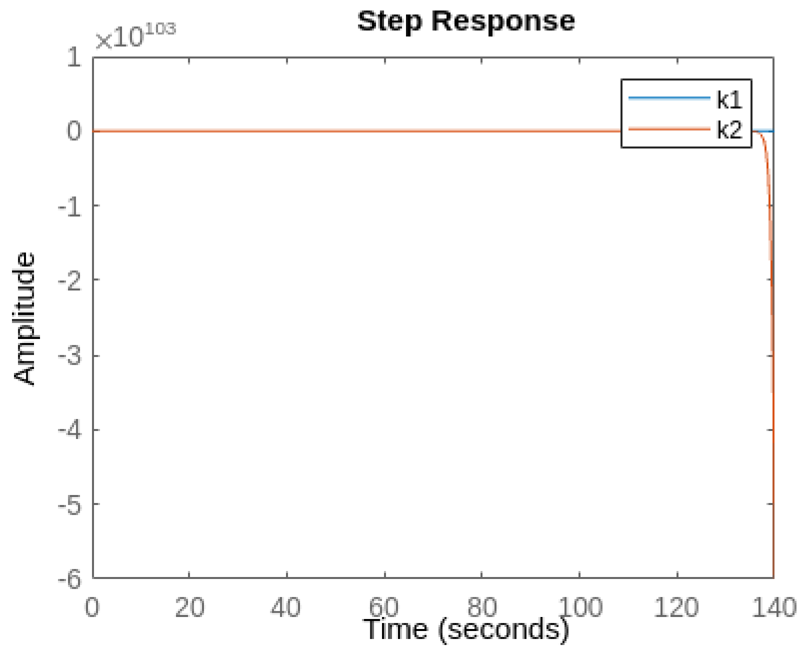
```
Select a point in the graphics window
```




```
selected_point = -2.3053 + 2.3601i
k2 = 3.6166
```

3. Razona qué diferencias y qué similitudes esperas en la respuesta al escalón utilizando cada uno de los valores de K anteriores.

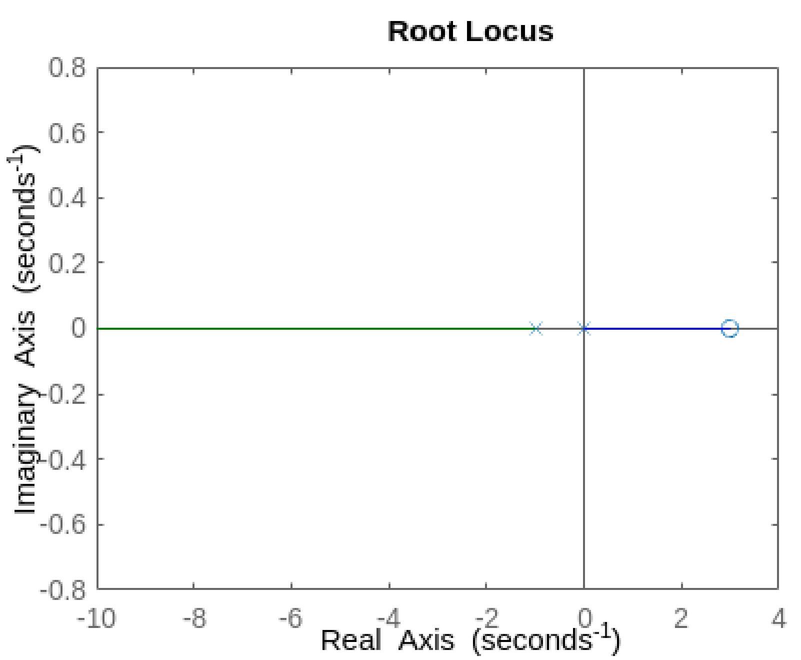
```
step(feedback(k1*G,H),feedback(k2*G,H))
legend('k1','k2')
```



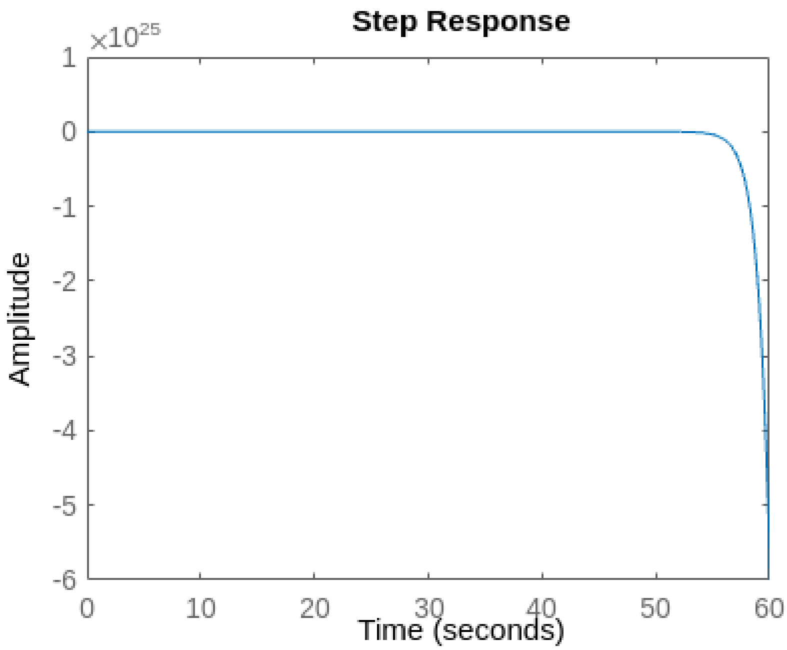
4. Representa la respuesta al escalón unitario del sistema completo para ambos valores de K y comprueba las afirmaciones que has hecho en el apartado 3. ¿Se cumplen las predicciones de sobreelongación máxima en ambos casos? ¿Por qué?

5. Resitúa el cero en s = +3 y vuelve a realizar la gráfica de la respuesta al escalón del sistema, para cualquier valor de K. Justifica en resultado en base al nuevo LDR.

```
G = zpk(3,[0 -1],1);
H = 1;
rlocus(G*H)
```



```
step(feedback(G*H,1))
```



2.3 Diseño de sistemas de orden superior mediante el LDR

Ejercicio práctico 7: Máquina-herramienta con control en velocidad

Para controlar la velocidad de una máquina herramienta se dispone de un sistema formado por un actuador que acciona directamente (en cascada) el motor de la misma. El sistema acepta órdenes de velocidad (no de posición) y:

1. El modelo del actuador, según su hoja de características, es un sistema de primer orden con ganancia estática de valor 1.2 y constante de tiempo de 2.56 segundos.

```
A = tf(1.2,[2.56 1]);
```

2. El motor dispone de un sensor para medir la velocidad del mismo. Del modelo motorsensor se ha obtenido un modelo empírico, dando como resultado un sistema de segundo orden con un valor de K de 0.1, y dos polos en las posiciones -0.3 y -0.1.

```
M = zpk([],[-0.1 -0.3],0.1);
G_la = A*M
```

```
G_la =

          0.046875
  --------------------------
  (s+0.3906) (s+0.3) (s+0.1)

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

Construye las funciones de transferencia del actuador y del motor-sensor en MATLAB y:

a) Obtén la evolución de la salida (velocidad), ante una entrada escalón unitario, cuando el sistema se encuentra en lazo abierto. Calcula el error en estado estacionario. ¿Es un error de velocidad?

```
step(G_la)
```

b) Para el sistema de control en lazo cerrado con realimentación unitaria, obtén la evolución de la salida (velocidad) ante una entrada escalón unitario, con una ganancia de lazo K unitaria. Determina de nuevo el error en estado estacionario y compara con el anterior. ¿Es este error nulo? ¿Por qué? ¿Cómo podría ser siempre nulo?

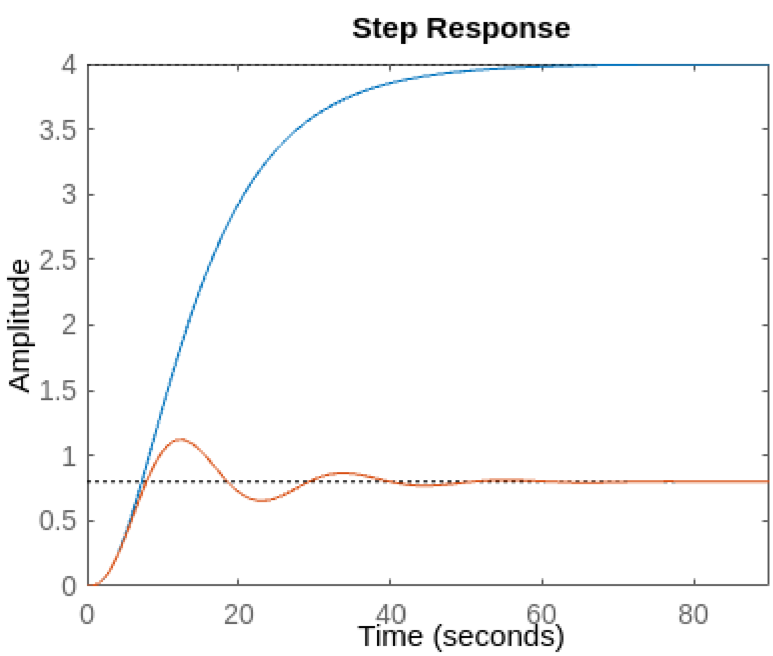
```
K = 1;
G_lc = feedback(K*G_la,1)
```

```
G_lc =

                0.046875
  -----------------------------------
  (s+0.6427) (s^2 + 0.1479s + 0.09117)

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

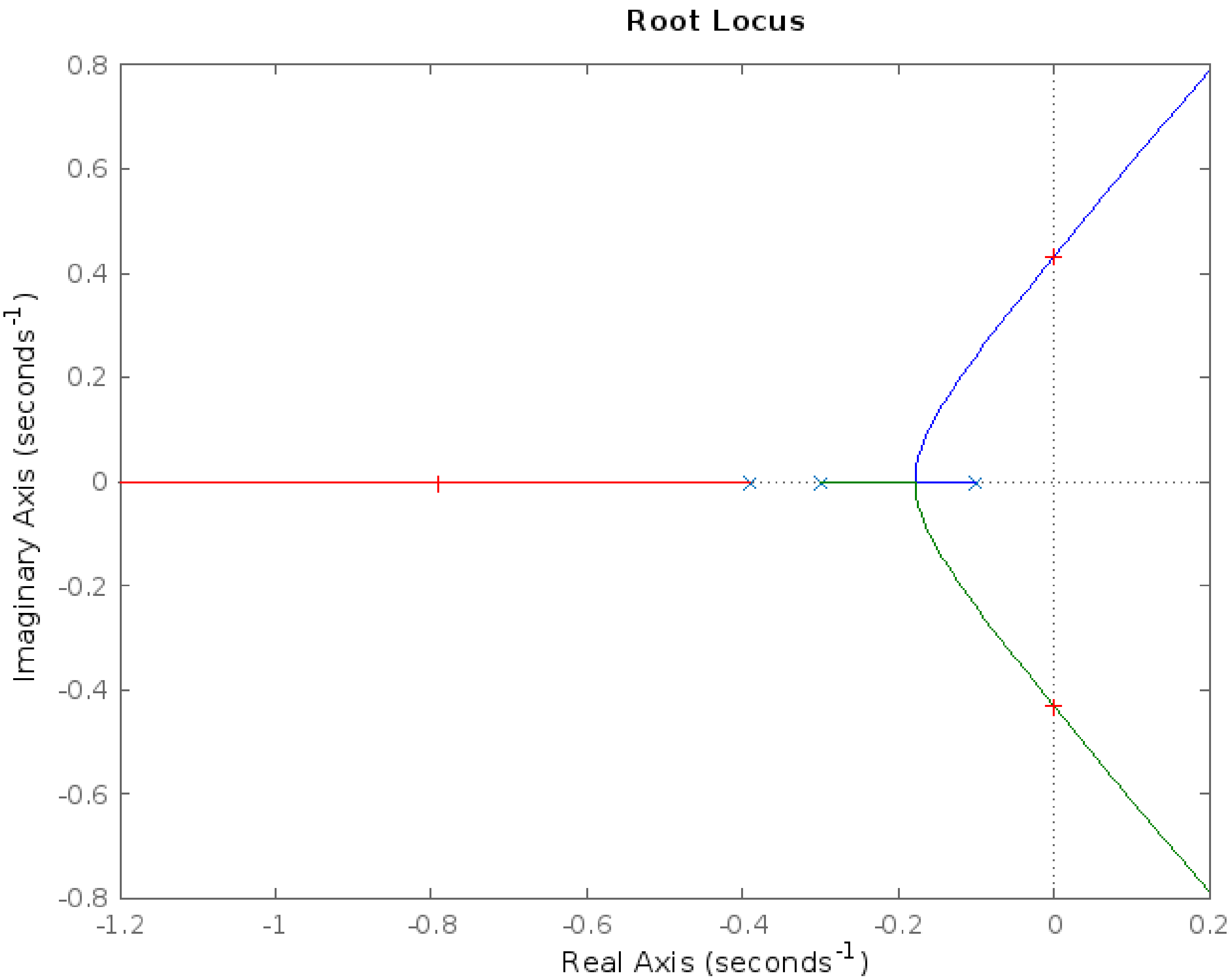
```
hold on
step(G_lc)
hold off
```



c) Dibuja el lugar de las raíces e indica si es posible aumentar el valor de la ganancia de lazo K al sistema para mejorar la respuesta en estado estacionario sin que éste se convierta en inestable. Justifica tu respuesta.

```
rlocus(G_la)
k_critica = rlocfind(G_la)
```

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




```
selected_point = -0.0002 + 0.4312i
k_critica = 2.8846
```

d) Representa la respuesta del sistema ante una entrada escalón unitario para dos valores de K > 1. Relaciona la frecuencia de oscilación y sobreelongación máxima de estas dos respuestas con la nueva posición de los polos dominantes en lazo cerrado según el lugar de las raíces. Calcula la posición de polo rápido del sistema para dichos valores de K. ¿Tiene alguna influencia en la respuesta temporal?