Resumen MATLAB práctica 3:

Polos y ceros de una función de transferencia en MATLAB:

```
% Extraer polos: polos = pole(G)
% Extraer ceros: ceros = zero(G)
% Mapa de polos y ceros: [polos,ceros] =  pzmap(G1,G2)
```

EJ. Realiza ahora una única gráfica de los polos y ceros de la función de transferencia en lazo cerrado GLC(s) del sistema de la figura, con la misma P(s) y con C(s) = K, para tres valores de la constante K, K=1, K=3 y K=10.

```
P = tf([1 1],conv([1 -1],conv([1 2],[1 4])))
```

```
P =
 
          s + 1
  ---------------------
  s^3 + 5 s^2 + 2 s - 8
 
Continuous-time transfer function.
```

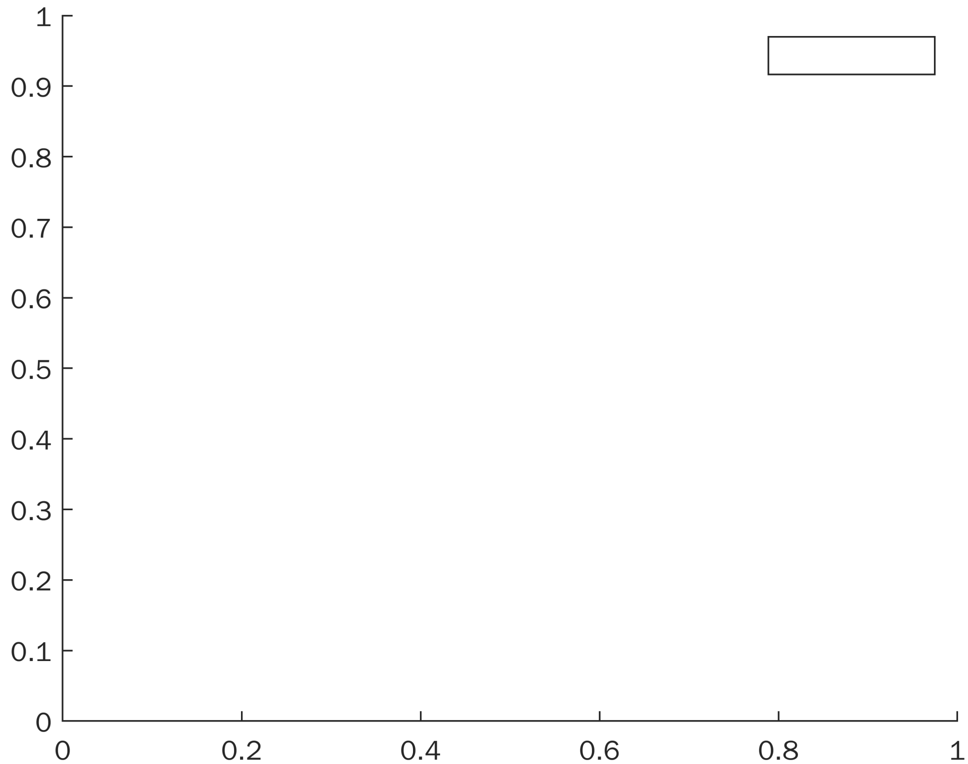
```
K = [1 3 10];
figure;hold on
for i=1:length(K)
    GLC = feedback(series(K(i),P),1)
end
```

```
GLC =
 
          s + 1
  ---------------------
  s^3 + 5 s^2 + 3 s - 7
 
Continuous-time transfer function.


GLC =
 
         3 s + 3
  ---------------------
  s^3 + 5 s^2 + 5 s - 5
 
Continuous-time transfer function.


GLC =
 
        10 s + 10
  ----------------------
  s^3 + 5 s^2 + 12 s + 2
 
Continuous-time transfer function.
```

```
legend('K=1','K=3','K=10')
```

```
Warning: Ignoring extra legend entries.
```

```
hold off
```



¿Influye el valor de K en la posición de los polos, de los ceros o de ambos en el sistema en lazo cerrado? Justifica tu respuesta.

Al modificar C(s) cambia el polinomio característico de la función de transferencia en lazo cerrado: "1 + C(s) * P(s)"

Estudio de la respuesta transitoria del sistema de control:

Entrada impulso unitario: impulse(G)

Entrada escalón unitario: step(G)

Definición de intervalo: impulse(G,X) (desde 0 hasta X) o definiendo:

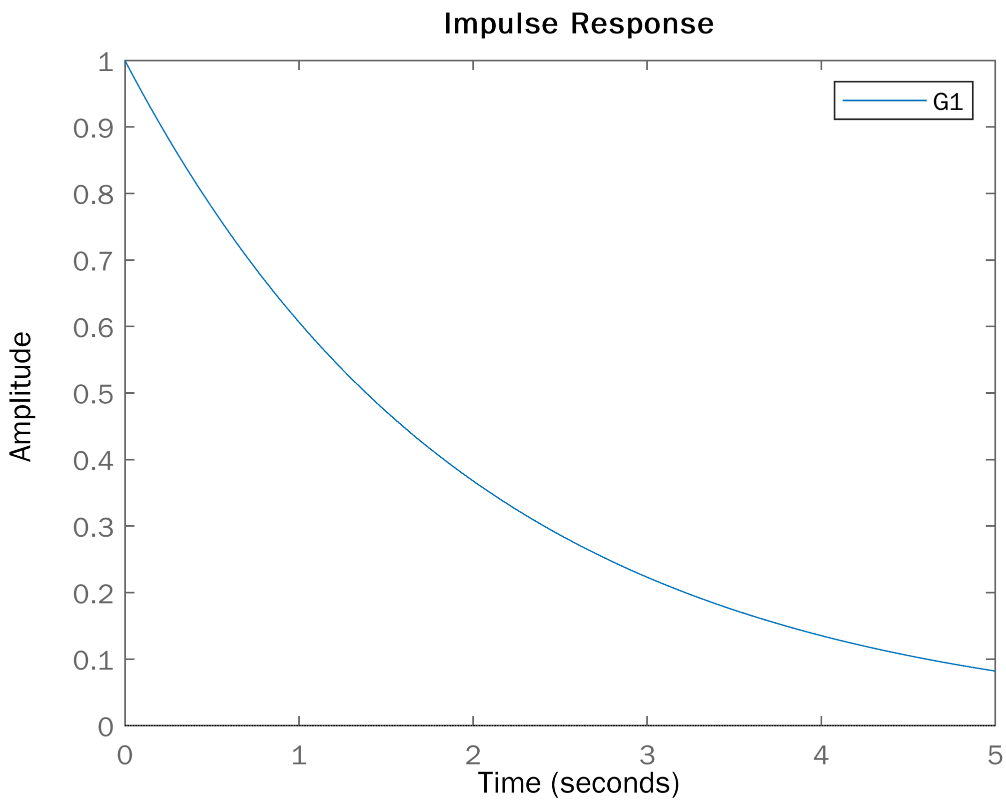
intervalo = 10:0.1:20; // step(G,intervalo)

EJ. Realiza una gráfica que incluya la respuesta al impulso unitario de todas ellas hasta un valor final de t = 5 s. Incluye la leyenda.

```
G1 = zpk([],[-0.5],1)
```

```
G1 =

     1
  -------
  (s+0.5)

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

```
figure; hold on
impulse(G1,5)
legend('G1'); hold off
```



Extracción de parámetros característicos de la respuesta temporal:

Gracias a este comando podemos extraer: Tiempo de establecimiento (settling time), tiempo de subida (rise time), valor final del estacionario (y oo), sobreelongación (overshoot) y tiempo de pico (peak time).

```
% respuesta_G = stepinfo(G)
```

EJ. Calcular el tiempo de establecimiento en una banda del 0.5% y el tiempo de subida del 5 al 95%.

```
% stepinfo(G, 'SettlingTimeThreshold', 0.005, 'RiseTimeThreshold', [0.05 0.95])
```

Tiempo de subida: respuesta_G1_RiseTime;

Tiempo de establecimiento: respuesta_G1.SettlingTime;

```
% G1 = zpk([],[-0.5],1);
% respuesta_G1 = stepinfo(G1);
% tr(1) = respuesta_G1_RiseTime;
% ts(1) = respuesta_G1.SettlingTime;
```

EJ. Utilizando comandos de control de flujo (for, while, if) realiza un script de MATLAB que calcule el valor que debe tener el polo de un sistema de primer orden para que su tiempo de subida (del 10 al 90%) sea igual a 1 ms.

```
close all
clear
clc
P = 3000;
G = tf(1,[1 P]);
respuestaG = stepinfo (G,'SettlingTimeThresold',0.005,'RiseTimeThresold',[0.1 0.9]);
i = 0;
respuestaG.RiseTime
```

```
ans = 7.3234e-04
```

```
while respuestaG.RiseTime <= 0.001 && i<1000
    i = i+100;
    G = tf(i,[1 P-i]);
    respuestaG = stepinfo(G,'SettlingTimeThresold',0.005,'RiseTimeThresold',[0.1 0.9]);
    respuestaG.RiseTime
end
```

```
ans = 7.5759e-04
ans = 7.8465e-04
ans = 8.1371e-04
ans = 8.4500e-04
ans = 8.7880e-04
ans = 9.1542e-04
ans = 9.5522e-04
ans = 9.9864e-04
ans = 0.0010
```

```
a = P - i
```

```
a = 2100
```

EJ. Repite el apartado anterior para extraer la frecuencia natural de un sistema de segundo orden con respuesta críticamente amortiguada que permite que el tiempo de asentamiento del sistema sea igual a 10 ms.

```
close all
clear
clc
Wn = 3000;
zeta = 1;
G = tf(1,[1 2*Wn Wn^2]);
respuestaG = stepinfo (G,'SettlingTimeThresold',0.005,'RiseTimeThresold',[0.1 0.9]);
i = 0;
respuestaG.SettlingTime
```

```
ans = 0.0025
```

```
while respuestaG.RiseTime <= 0.01 && i<3000
    i = i+1;
    Wn = Wn - 1;
```

```
    G = tf(1,[1 2*Wn Wn^2]);
    respuestaG = stepinfo(G,'SettlingTimeThresold',0.05,'RiseTimeThresold',[0.1 0.9]);
    respuestaG.SettlingTime;
end
Wn
```

```
Wn = 374
```

```
respuestaG
```

```
respuestaG = struct with fields:
         RiseTime: 0.0100
    TransientTime: 0.0158
     SettlingTime: 0.0158
      SettlingMin: 7.1150e-06
      SettlingMax: 7.1492e-06
        Overshoot: 0
       Undershoot: 0
             Peak: 7.1492e-06
         PeakTime: 0.1100
```

Respuesta a otro tipo de entradas: Rampa unitaria, aceleración unitaria, etc:

```
% lsim(G,entrada,tiempo)
% tiempo: Vector que debe contener un intervalo de valores
% entrada: vector de señal de entrada (aceleracion = tiempo^2)
```

EJ. Realiza una gráfica con la respuesta temporal de los sistemas de primer orden del ejercicio práctico 3 (una vez ajustadas las K) a una entrada rampa de pendiente igual a 3.

```
% figure
% t = 0:i:20;
% for i = 1:5
%     lsim(G2(i),3.*t,t)
%     hold on;
% end
% figure;
```

Error de posición en estado estacionario:

```
% Amplitud (configurable) del escalón
% Amplitud = 2;
% Almacenamos respuesta hasta t_final en el vector y
% [y,t] = step(Amplitud*GLC,t_final)
% Error absoluto, con su signo
% ess_pos_abs = Amplitud - y(end)
% Error relativo, en porcentaje, con signo
% ess_pos_rel = 100*(Amplitud - y(end)) / Amplitud
```