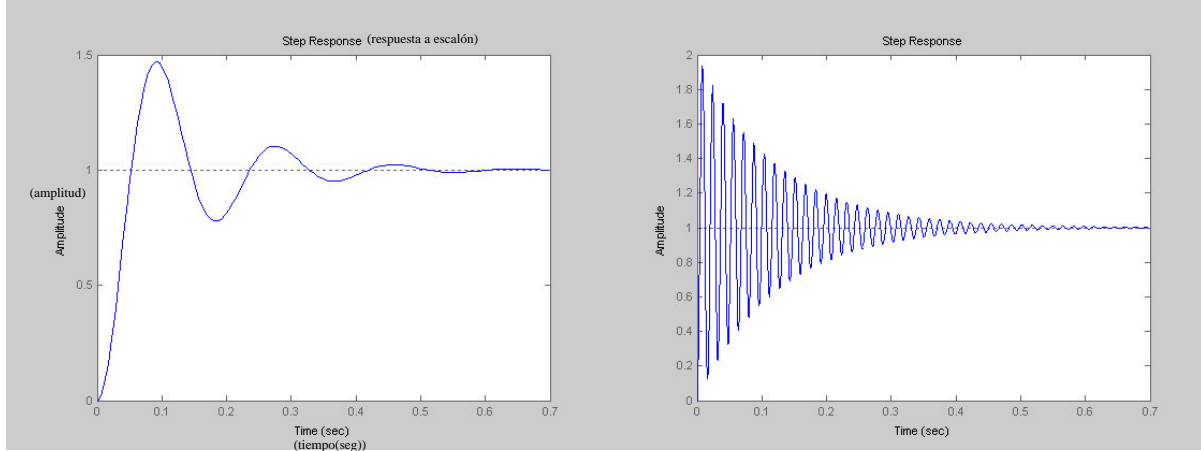


Práctica 7: soluciones

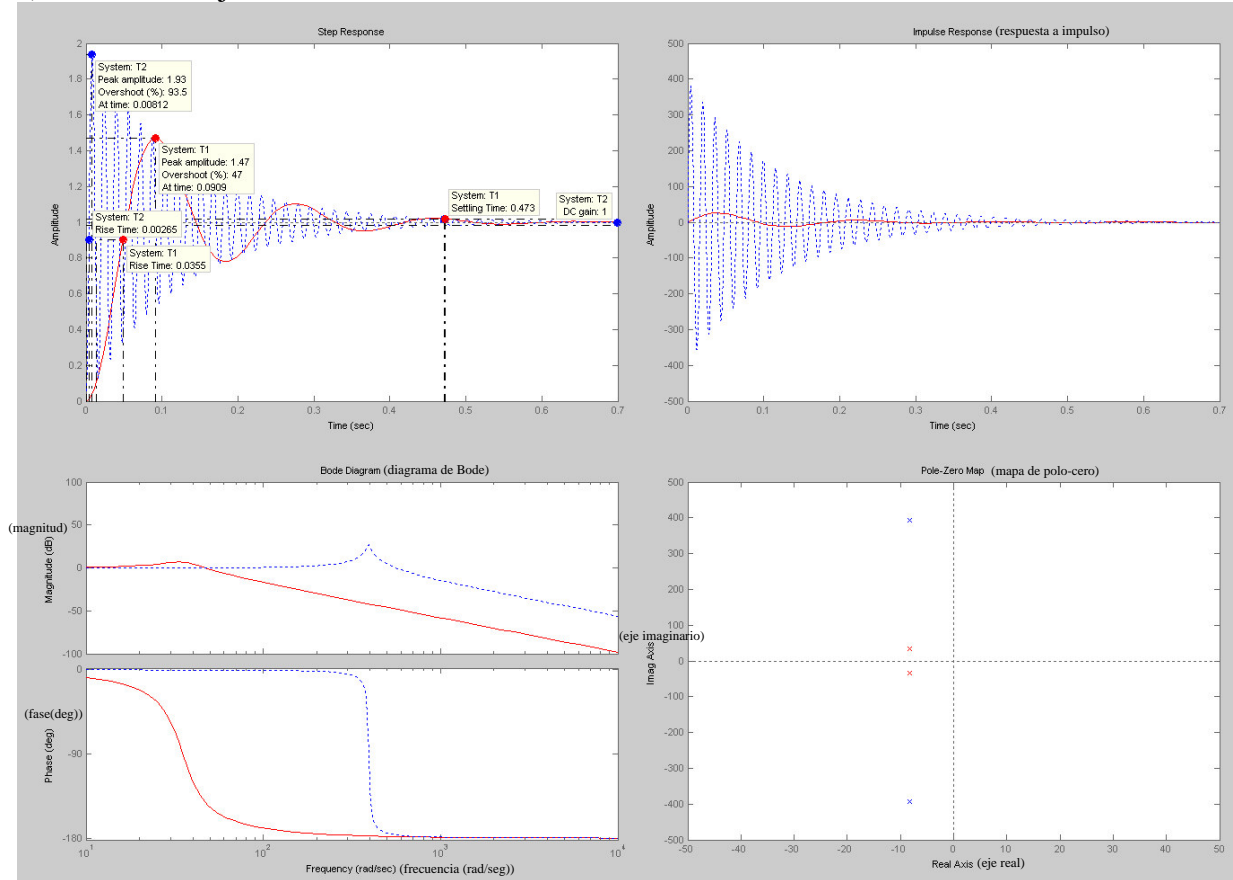
a) véase el código de Matlab incluido al final del documento

b) F1

F2



c) F1 – línea roja continua; F2 – línea azul discontinua;



d) i) polos de bucle abierto para $K * F1$

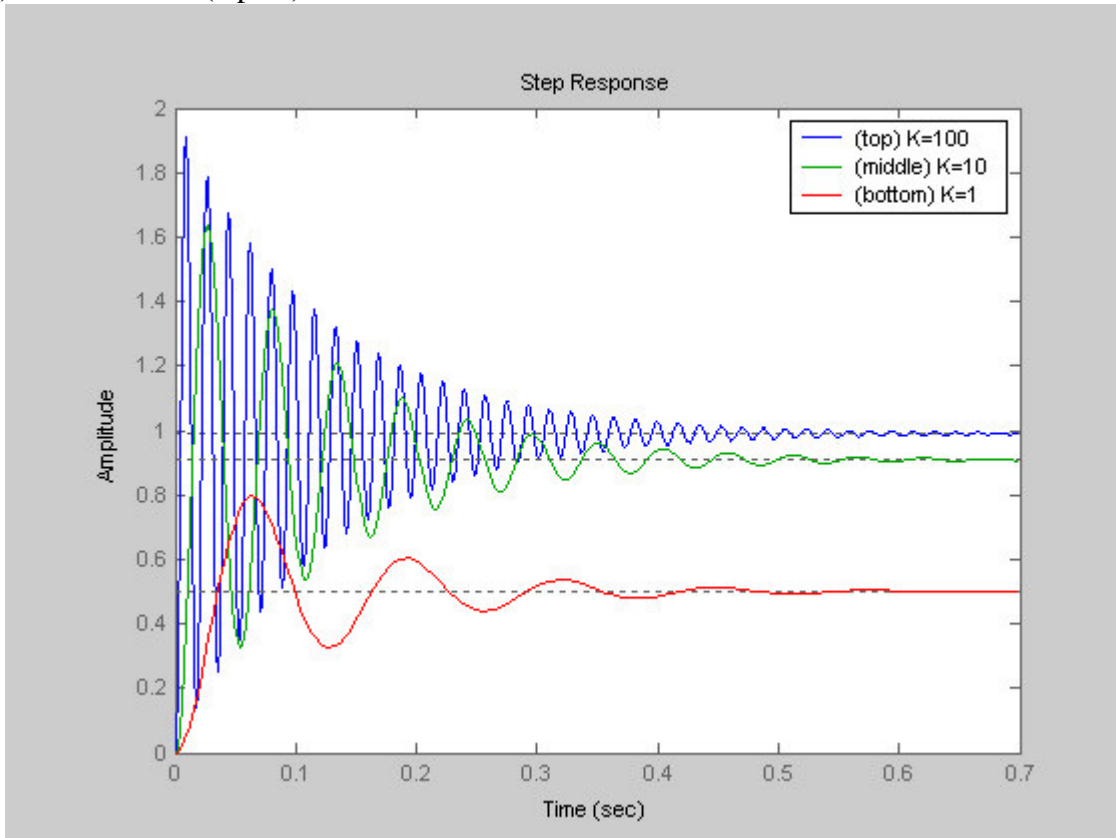
K=100	K=10	K=1
-8.2353 +34.2715i	-8.2353 +34.2715i	-8.2353 +34.2715i
-8.2353 -34.2715i	-8.2353 -34.2715i	-8.2353 -34.2715i

ii)
$$\frac{1056}{0.85 s^2 + 14 s + 2112} \quad (K=1)$$

polos de bucle cerrado para $K * F1$

K=100	K=10	K=1
1.0e+002 *		
-0.0824 + 3.5413i	-0.0824 + 1.1661i	-0.0824 + 0.4916i
-0.0824 - 3.5413i	-0.0824 - 1.1661i	-0.0824 - 0.4916i

iii) Hecho. `rltool(KpFB)`.



iv) Al aumentar la ganancia proporcional (K_p) en el sistema de retroalimentación, la frecuencia de las oscilaciones aumenta radicalmente (debido a que la disminución de amortiguación de la parte imaginaria de los polos es mayor), el % de sobreexceso aumenta poco a poco, el tiempo de estabilización permanece invariable(ya que las partes reales de los polos no están en movimiento), el tiempo de subida y el tiempo pico disminuyen, y el error de estado estacionario se aproxima a cero cuando K_p se acerca a infinito.

e) i)
$$\frac{1056 s + 1056}{0.85 s^2 + 1070 s + 2112} \quad (K=1)$$

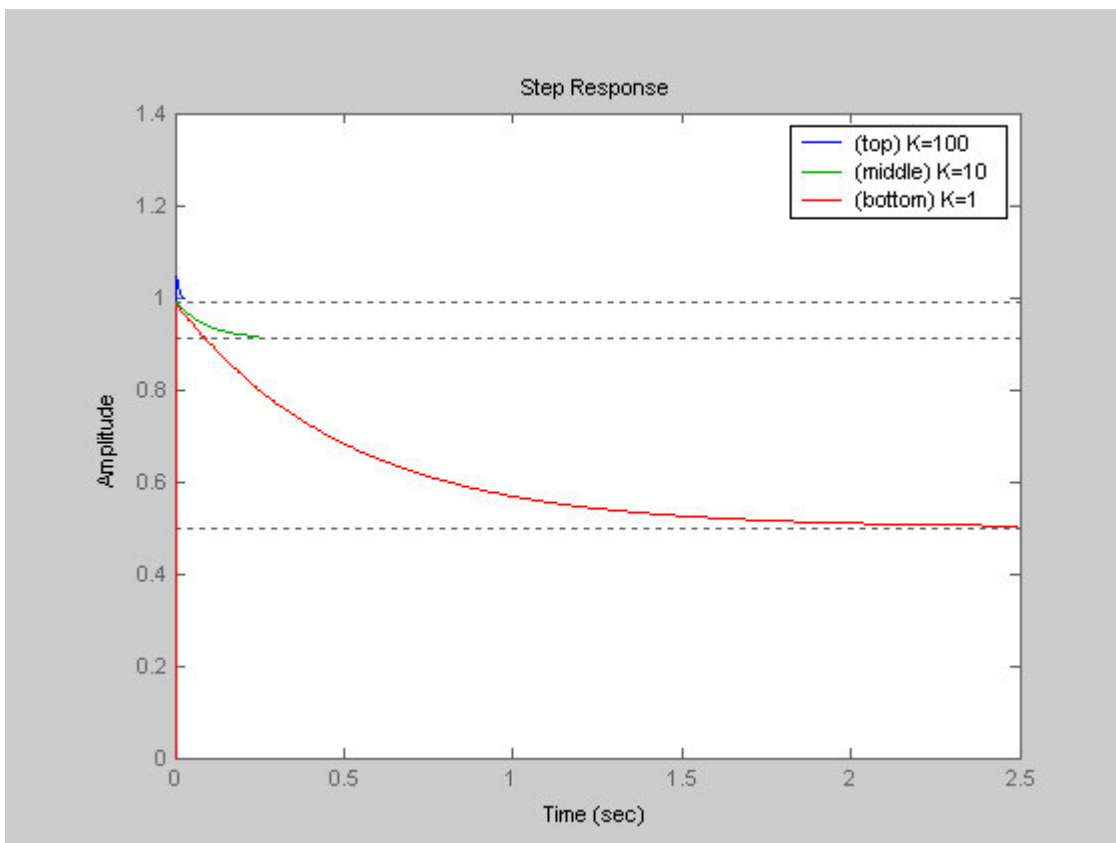
ii) polos de bucle cerrado para $(s + K) * F1$

K=100 1.0e+003 *	K=10	K=1
-1.1497	-1.2479	-1.2568
-0.1091	-0.0110	-0.0020

polos de bucle cerrado para $(s + K) * F1$

K=100	K=10	K=1
-100	-10	-1

iii) rltool(KdFB)



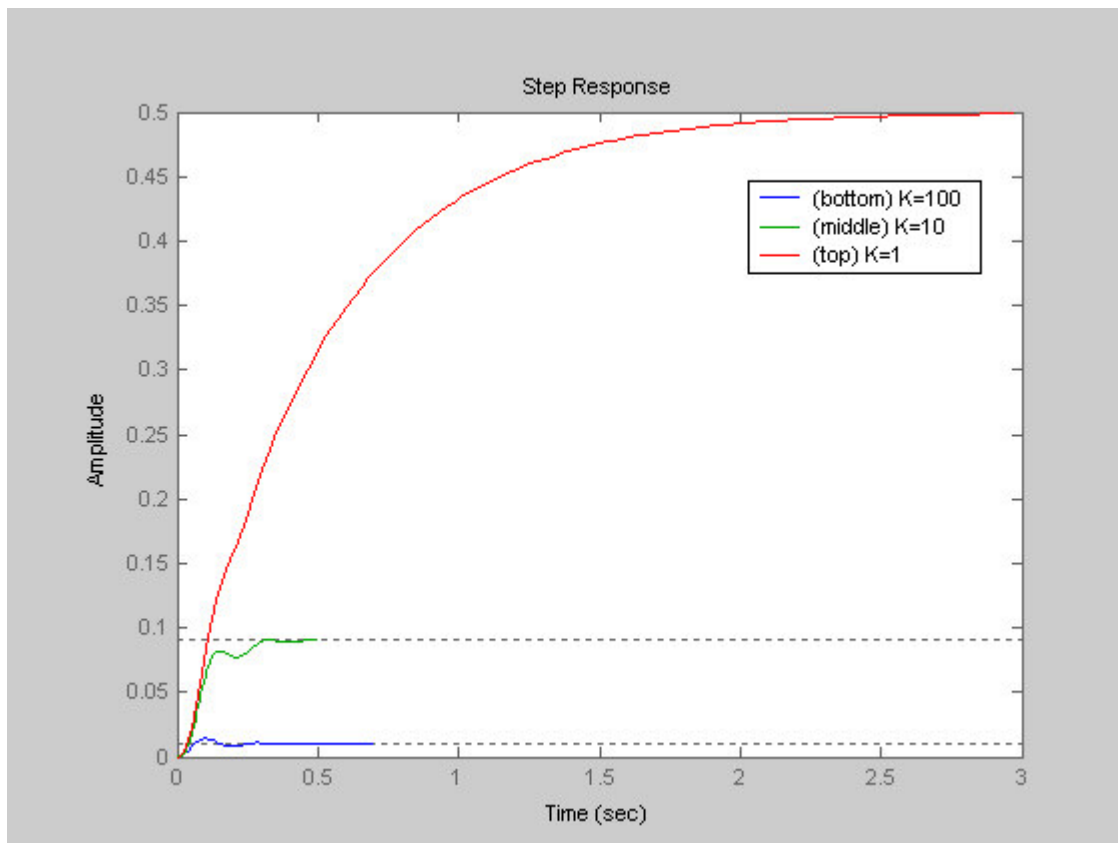
iv) añadiendo un cero disminuye el error de estado estacionario, aumenta el tiempo de estabilización, disminuyen los tiempos de subida y de pico radicalmente y se eliminan las oscilaciones. Aumentando K aumenta aún más el error de estado estacionario y disminuye el % de sobreexceso del sistema.

f) i)
$$\frac{1056}{0.85 s^3 + 14.85 s^2 + 1070 s + 2112} \quad (K=1)$$

ii) polos de bucle cerrado para $1/(s + K) * F1$

K=100	K=10	K=1
1.0e+002 *		
-1.0013	-0.0771 + 0.3431i	-0.0772 + 0.3417i
-0.0817 + 0.3444i	-0.0771 - 0.3431i	-0.0772 - 0.3417i
-0.0817 - 0.3444i	-0.1105	-0.0202

no hay ceros para este sistema.



Si se añade un polo aumentan los tiempos de subida y de pico y el tiempo de estabilización, proporcionando una amortiguación y eliminando las oscilaciones. No se recomienda aumentar más K. No obstante, con los valores que posee aumentan las oscilaciones, el % de sobreexceso y el error de estado estacionario, pero disminuye el tiempo de estabilización,

```
%----- Código de Matlab
```

```
% a) defina las funciones de transferencia utilizando TF();
```

```
F1 = TF([1056],[.85 14 1056]);  
F2 = TF([132000],[.85 14 132000]);
```

```
% b) visualice la respuesta a escalón utilizando STEP();
```

```
figure(1)  
STEP(F1)  
figure(2)  
STEP(F2)
```

```
% c) juegue con estos valores en el visor LTI.
```

```
LTIView({'step','impulse','bode','pzmap'},F1,'r',F2,'b:')
```

```
% d) ganancia proporcional
```

```
kk = [100 10 1];
```

```
para i=1:length(kk)
```

```
    Kp = TF([kk(i)],[1]);  
    KpF = series(Kp,F1);  
    ppo(:,i)=pole(KpF);  
    zpo(:,i)=zero(KpF);
```

```
    KpFB = feedback(KpF,1);  
    ppc(:,i)=pole(KpFB);  
    zpc(:,i)=zero(KpFB);
```

```
    figure(3)  
    STEP(KpFB);  
    hold on;
```

```
    Kd = TF([1 kk(i)],[1]);  
    KdF = series(Kd,F1);  
    KdFB = feedback(KdF,1)  
    pdc(:,i)=pole(KdFB);  
    zdc(:,i)=zero(KdFB);
```

```
    figure(4)  
    STEP(KdFB);  
    hold on;
```

```
    Ki = TF([1],[1 kk(i)]);
```

```
KiF = series(Ki,F1);  
KiFB = feedback(KiF,1);  
pic(:,i)=pole(KiFB);  
zic(:,i)=zero(KiFB);
```

```
figure(5)  
STEP(KiFB);  
hold on;
```

fin

```
'open-loop poles for K * F1'  
fprintf('\tK=100\t\t\t\tK=10\t\t\t\tK=1\n')  
disp(ppo);
```

```
'closed-loop poles for K * F1'  
fprintf('\tK=100\t\t\t\tK=10\t\t\t\tK=1\n')  
disp(ppc);
```

```
'closed-loop poles for (s + K) * F1'  
fprintf('\tK=100\t\t\t\tK=10\t\t\t\tK=1\n')  
disp(pdc);
```

```
'closed-loop zeros for (s + K) * F1'  
fprintf('\tK=100\t\t\t\tK=10\t\t\t\tK=1\n')  
disp(zdc);
```

```
'closed-loop poles for 1/(s + K) * F1'  
fprintf('\tK=100\t\t\t\tK=10\t\t\t\tK=1\n')  
disp(pic);
```

```
'closed-loop zeros for 1/(s + K) * F1'  
fprintf('\tK=100\t\t\t\tK=10\t\t\t\tK=1\n')  
disp(zic);
```