



UNIVERSIDAD NACIONAL DE CÓRDOBA  
FACULTAD DE CIENCIAS EXACTAS, FÍSICAS Y NATURALES  
CÁTEDRA DE SISTEMAS DE CONTROL I

**Ejercicios (26 al 29) Unidad 4**

Nombre: Monja Ernesto Joaquín

DNI: 43.873.728

Año 2024

## Problema 26:

$G1(s)$ :

```
close all; clear all; history -c; clc;
pkg load control;

s = tf('s');
G1 = 10/((s + 10)*(s + 100));

% Requerimientos: ess = 0 ; Ts minimo ; Mp = 0 (respuesta criticamente amort.)
% Se utiliza un compensador PI para cancelar el polo dominante, esto es:
Ti = 1/10;
PI = (s + 1/Ti)/s;
FdTLA = PI*G1;
rlocus(FdTLA)

% Para una respuesta criticamente amortiguada, se elige un punto donde las
% raices sean reales, negativas e iguales, por lo que segun el rlocus, este
% punto es:
s1 = -50;
Kp = 1/abs((s1 + 1/Ti)/s1 * 10/((s1 + 10)*(s1 + 100))) % Kp = 250
PI = Kp*PI;
FdTLC = minreal(feedback(G1*PI, 1))
% Se verifican estos valores con:
pole(FdTLC) % Se ven los polos dobles en s = -50
step(G1, FdTLC)
```

$G2(s)$ :

```
close all; clear all; history -c; clc;
pkg load control;

s = tf('s');
G2 = 0.8/((0.1*s + 1)*(0.05*s + 1));
pole(G2) % Se tiene que los polos estan en -10 y -20

% Requerimientos: ess = 0 ; Ts minimo ; Mp = 0 (respuesta criticamente amort.)
% Se utiliza un compensador PI para cancelar el polo dominante, esto es:
Ti = 1/10;
PI = (s + 1/Ti)/s;
FdTLA = PI*G2;
rlocus(FdTLA)

% Para una respuesta criticamente amortiguada, se elige un punto donde las
% raices sean reales, negativas e iguales, por lo que segun el rlocus, este
% punto es:
s1 = -10;
G2PI = minreal(G2*PI);
Kp = 1/abs( (160/(s1^2 + 20*s1)) ) % Kp = 0.6250
PI = Kp*PI;
FdTLC = minreal(feedback(G2*PI, 1))
% Se verifican estos valores con:
pole(FdTLC) % Se ven los polos dobles en s = -10
step(G2, FdTLC, 10)
```

### G3(s):

```
close all; clear all; history -c; clc;
pkg load control;

s = tf('s');
G3 = 125/(s^2 + 33*s + 200);
pole(G3)      % Se tiene que los polos estan en -8 y -25

% Requerimientos: ess = 0 ; Ts minimo ; Mp = 0 (respuesta criticamente amort.)
% Se utiliza un compensador PI para cancelar el polo dominante, esto es:
Ti = 1/8;
PI = (s + 1/Ti)/s;
FdTLA = PI*G3;
rlocus(FdTLA)
% Para una respuesta criticamente amortiguada, se elige un punto donde las
% raices sean reales, negativas e iguales, por lo que segun el rlocus, este
% punto es:
s1 = -12.5;
G3PI = minreal(G3*PI);
Kp = 1/abs(125/(s1^2 + 25*s1))      % Kp = 1.25
PI = Kp*PI;
FdTLC = minreal(feedback(G3*PI, 1))
% Se verifican estos valores con:
pole(FdTLC)      % Se ven los polos dobles en s = -12.5
step(G3, FdTLC, 10)
```

### G4(s):

```
close all; clear all; history -c; clc;
pkg load control;

s = tf('s');
G4 = 1.24/(s + 1)^3;
pole(G4)      % Se tiene que los polos estan en -1

% Requerimientos: ess = 0 ; Ts minimo ; Mp = 0 (respuesta criticamente amort.)
% Se utiliza un compensador PI para cancelar el polo dominante, esto es:
Ti = 1/1;
PI = (s + 1/Ti)/s;
FdTLA = PI*G4;
rlocus(FdTLA)
% Para una respuesta criticamente amortiguada, se elige un punto donde las
% raices sean reales, negativas e iguales, por lo que segun el rlocus, este
% punto es:
s1 = -0.3333;
G4PI = minreal(G4*PI)
Kp = 1/abs(1.24/(s1^3 + 2*s1^2 + s1))      % Kp = 0.1195
rlocusx(FdTLA)      % Kp = 0.12 aproximadamente
PI = 0.1195*PI;
FdTLC = minreal(feedback(G4*PI, 1))
% Se verifican estos valores con:
pole(FdTLC)      % Se cancelo uno de los polos y los otros tienen parte
                  % real igual a -0.33
step(G4, FdTLC, 30)
```

### G5(s):

```
close all; clear all; history -c; clc;
pkg load control;

s = tf('s');
G5 = 20*(s + 60)/((s + 10)*(s + 20));
pole(G5)      % Se tiene que los polos estan en -10 y -20

% Requerimientos: ess = 0 ; Ts minimo ; Mp = 0 (respuesta criticamente amort.)
% Se utiliza un compensador PI para cancelar el polo dominante, esto es:
Ti = 1/10;
PI = (s + 1/Ti)/s;
FdTLA = PI*G5;
rlocus(FdTLA)
% Para una respuesta criticamente amortiguada, se elige un punto donde las
% raices sean reales, negativas e iguales, por lo que segun el rlocus, este
% punto es:
s1 = -109;
G5PI = minreal(G5*PI)
Kp = 1/abs((20*s1 + 1200)/(s1^2 + 20*s1))      % Kp = 9.8990
rlocusx(FdTLA)                                % Kp = 9.9 aproximadamente
PI = 9.8990*PI;
FdTLC = minreal(feedback(G5*PI, 1))
% Se verifican estos valores con:
pole(FdTLC)      % se observan polos dobles casi iguales uno del otro
step(G4, FdTLC, 30)
```

### G6(s):

```
close all; clear all; history -c; clc;
pkg load control;

s = tf('s');
G6 = (800*s + 32000)/(s^3 + 330*s^2 + 29000*s + 600000);
pole(G6)      % Se tiene que los polos estan en -30, -100 y -200

% Requerimientos: ess = 0 ; Ts minimo ; Mp = 0 (respuesta criticamente amort.)
% Se utiliza un compensador PI para cancelar el polo dominante, esto es:
Ti = 1/30;
PI = (s + 1/Ti)/s;
FdTLA = PI*G6;
rlocus(FdTLA)
% Para una respuesta criticamente amortiguada, se elige un punto donde las
% raices sean reales, negativas e iguales, por lo que segun el rlocus, este
% punto es:
s1 = -146.82;
G6PI = minreal(G6*PI)
Kp = 1/abs((800*s1 + 3.2e+04)/(s1^3 + 300*s1^2 + 2e+04*s1))  % Kp = 4.2778
rlocusx(FdTLA)                                              % Kp = 4.28 aprox.
PI = 4.2778*PI;
FdTLC = minreal(feedback(G6*PI, 1))
% Se verifican estos valores con:
pole(FdTLC)      % Se tiene que dos polos son practicamente iguales,
                  % cumpliendo asi el requerimiento de psita.
step(G6, FdTLC, 5)
```

## Problema 27:

$G1(s)$ :

```
close all; clear all; history -c; clc;
pkg load control;

s = tf('s');
G1 = 10/((s + 10)*(s + 100));
% Se añade un compensador PI para cancelar el polo dominante:
Ti = 1/10;
PI = (s + 1/Ti)/s;

% Para un sobrepasamiento máximo del 4% corresponde a un cierto valor de psita
% el cual se puede calcular a continuación:
psita = sqrt((log(0.04)^2)/(pi^2 + log(0.04)^2))    % psita = 0.7156
% Se verifica donde coinciden los valores de psita con el LdR mediante el rlocus
rlocus(minreal(PI*G1)); sgrid(psita, [20 40 60 80])

s1 = -50 + j*48.8;
Kp = 1/abs( (s1 + 1/Ti)/s1 * 10/((s1 + 10)*(s1 + 100)) )    % Kp = 488.14
% Verificamos este resultado mediante el rlocusx():
rlocusx(G1*PI)                                              % Kp = 488 aprox.
```

$G2(s)$ :

```
close all; clear all; history -c; clc;
pkg load control;

s = tf('s');
G2 = 0.8/((0.1*s + 1)*(0.05*s + 1));
% Se añade un compensador PI para cancelar el polo dominante:
Ti = 1/10;
PI = (s + 1/Ti)/s;

% Para un sobrepasamiento máximo del 4% corresponde a un cierto valor de psita
% el cual se puede calcular a continuación:
psita = sqrt((log(0.04)^2)/(pi^2 + log(0.04)^2))    % psita = 0.7156
% Se verifica donde coinciden los valores de psita con el LdR mediante el rlocus
rlocus(minreal(PI*G2)); sgrid(psita, [20 40 60 80])

s1 = -10 + j*9.76;
G2PI = minreal(G2*PI)
Kp = 1/abs(160/(s1^2 + 20*s1))                        % Kp = 1.2204
% Verificamos este resultado mediante el rlocusx():
rlocusx(G2*PI)                                          % Kp = 1.22 aprox.
```

$G3(s)$ :

```

close all; clear all; history -c; clc;
pkg load control;

s = tf('s');
G3 = 125/(s^2 + 33*s + 200);
% Se añade un compensador PI para cancelar el polo dominante:
Ti = 1/8;
PI = (s + 1/Ti)/s;

% Para un sobrepasamiento máximo del 4% corresponde a un cierto valor de psita
% el cual se puede calcular a continuación:
psita = sqrt((log(0.04)^2)/(pi^2 + log(0.04)^2))    % psita = 0.7156
% Se verifica donde coinciden los valores de psita con el LdR mediante el rlocus
rlocus(minreal(PI*G3)); sgrid(psita, [20 40 60 80])

s1 = -12.5 + j*12.2;
G3PI = minreal(G3*PI);
Kp = 1/abs(125/(s1^2 + 25*s1))    % Kp = 2.4407
% Verificamos este resultado mediante el rlocusx():
rlocusx(G3*PI)    % Kp = 2.44 aprox.

```

$G4(s)$ :

```

close all; clear all; history -c; clc;
pkg load control;

s = tf('s');
G4 = 1.24/(s + 1)^3;
% Se añade un compensador PI para cancelar el polo dominante:
Ti = 1/1;
PI = (s + 1/Ti)/s;

% Para un sobrepasamiento máximo del 4% corresponde a un cierto valor de psita
% el cual se puede calcular a continuación:
psita = sqrt((log(0.04)^2)/(pi^2 + log(0.04)^2))    % psita = 0.7156
% Se verifica donde coinciden los valores de psita con el LdR mediante el rlocus
rlocus(minreal(PI*G4)); sgrid(psita, [20 40 60 80])

s1 = -0.294 + j*0.287;
G4PI = minreal(G4*PI);
Kp = 1/abs(1.24/(s1^3 + 2*s1^2 + s1))    % Kp = 0.1924
% Verificamos este resultado mediante el rlocusx():
rlocusx(G4*PI)    % Kp = 0.19 aprox.

```

$G5(s)$ :

```

close all; clear all; history -c; clc;
pkg load control;

s = tf('s');
G5 = 20*(s + 60)/((s + 10)*(s + 20));
% Se añade un compensador PI para cancelar el polo dominante:
Ti = 1/10;
PI = (s + 1/Ti)/s;

% Para un sobrepasamiento máximo del 4% corresponde a un cierto valor de psita
% el cual se puede calcular a continuación:
psita = sqrt((log(0.04)^2)/(pi^2 + log(0.04)^2))    % psita = 0.7156
% Se verifica donde coinciden los valores de psita con el LdR mediante el rlocus
rlocus(minreal(PI*G5)); sgrid(psita, [20 40 60 80])

sl = -48.8 + j*47.7;
G5PI = minreal(G5*PI)
Kp = 1/abs((20*sl + 1200)/(sl^2 + 20*sl))    % Kp = 3.8802
% Verificamos este resultado mediante el rlocusx():
rlocusx(G5*PI)    % Kp = 3.89 aprox.

```

G6(s):

```

close all; clear all; history -c; clc;
pkg load control;

s = tf('s');
G6 = (800*s + 32000)/(s^3 + 330*s^2 + 29000*s + 600000);
% Se añade un compensador PI para cancelar el polo dominante:
Ti = 1/30;
PI = (s + 1/Ti)/s;

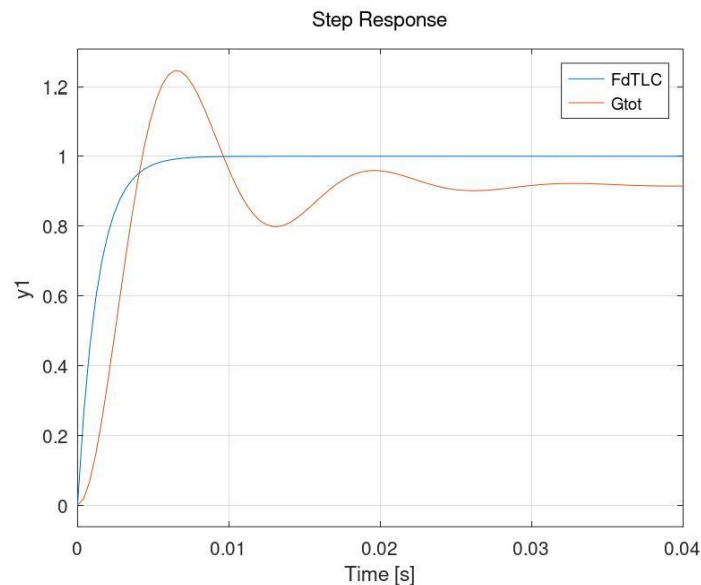
% Para un sobrepasamiento máximo del 4% corresponde a un cierto valor de psita
% el cual se puede calcular a continuación:
psita = sqrt((log(0.04)^2)/(pi^2 + log(0.04)^2))    % psita = 0.7156
% Se verifica donde coinciden los valores de psita con el LdR mediante el rlocus
rlocus(minreal(PI*G6)); sgrid(psita, [20 40 60 80])

sl = -137 + j*133.7;
G6PI = minreal(G6*PI)
Kp = 1/abs((800*sl + 3.2e+04)/(sl^3 + 300*sl^2 + 2e+04*sl))    % Kp = 29.702
% Verificamos este resultado mediante el rlocusx():
rlocusx(G6*PI)    % Kp = 29.7 aprox.

```

## **Problema 28:**

Respuesta al impulso del sistema sin compensar contra el compensado:



Código:

```
close all; clear all; history -c; clc;
pkg load control;

s = tf('s');

% Planteo la función de transferencia la cual voy a compensar como:
G1 = 100/(s + 100);
G2 = 2340/(s + 212.8);
Gtot = minreal(feedback(G1*G2, 1))

% Vemos el valor de los polos de esta función Gtot:
pole(Gtot)      % Se observan polos en: s = -156.4 +- j*480.44

% Luego, dado que se nos pide utilizar un compensador PID, este tiene una
% función de transferencia como la que se muestra a continuación:
% PID = Kp*(Ti*Td*s^2 + Ti*s + 1)/(Ti*s) = Kp*(s^2 + s/Td + 1/(Ti*Td))/Td*Ti*s

% Donde se tendra que los ceros dobles de este PID cancelaran a los polos
% dominantes de la función Gtot, por lo tanto se tendra que:
% s^2 + s/Td + 1/(Ti*Td) = s^2 + 312.8*s + 2553e+05
% De aquí se pueden deducir los valores de las constantes Td y Ti como:
Td = 1/312.8          % Td = 3.1969e-03
Ti = 1/(2.553e+05*Td) % Ti = 1.2252e-03

% Por lo tanto resulta que el controlador PID tiene la siguiente forma:
PID = (Ti*Td*s^2 + Ti*s + 1)/(Ti*s)

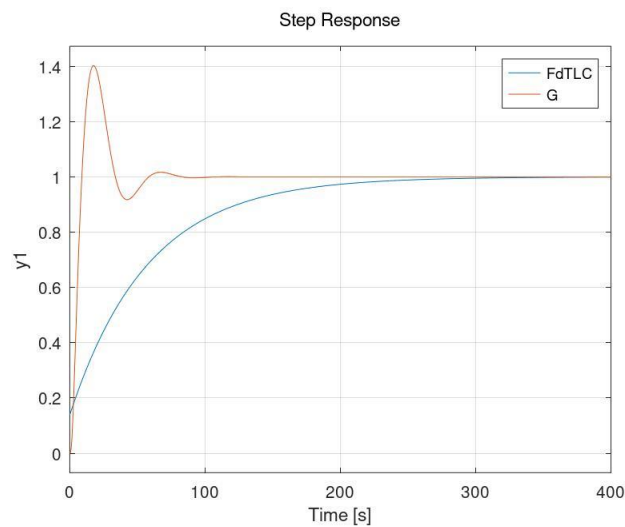
% Resta solo calcular el valor de Kp, por lo que asumiendo un sobrepasamiento
% maximo del 4% (psita = 0.707) se tiene que:
FdTLA = minreal(PID*Gtot);
rlocus(FdTLA); sgrid(0.707, [20 40 60 80])

% Se escoge arbitrariamente un Kp = 1 de modo tal que la función de transferencia
% a lazo cerrado sea la siguiente:
FdTLC = minreal(feedback(FdTLA, 1));
step(FdTLC, Gtot)
```



## Problema 29:

Respuesta al impulso del sistema sin compensar contra el compensado:



Código:

```
close all; clear all; history -c; clc;
pkg load control;

s = tf('s');

% Utilizando la resolucio del ejercicio 4 de la Unidad 3, se obtuvo que:
% Ma = 500; Ms = 50; Kr = 10; Kc = 50; Ba = 80;
% Xa_W = simplify((Kc*(Ba*s + Kr))/((Ma*Ms)*s^4 + (Ba*(Ma + Ms))*s^3 +
%      + (Ma*(Kc+Kr)+Kr*Ms)*s^2 + (Ba*Kc)*s + (Kc*Kr)))

G = (8*s + 1)/(s^4*50 + s^3*88 + s^2*61 + s*8 + 1)

% Veamos como responde este sistema a un impulso y veamos donde estan sus ceros
% y polos:

step(G); pole(G)
% Se trata de un sistema con cierto sobrepasamiento, se observa tambien que sus
% 4 polos son complejos conjugados de modo donde:
s1 = -0.0641 + j*0.1268;
s2 = -0.0641 - j*0.1268;
s3 = -0.8159 + j*0.5700;
s4 = -0.8159 - j*0.5700;

%% DISEÑO DEL CONTROLADOR PID:
% Se tiene que utilizaremos los ceros del PID para cancelar los polos complejos
% conjugados mas cerca del origen, estos son s1 y s2, por lo tanto diremos que
% (s + s1)*(s + s2) = s^2 + 0.1282*s + 0.02018705
Greal = ((8*s + 1))/(s^2 + 0.1282*s + 0.02018705)

% Ahora, dado que el PID cancelara estos dos polos, se tendra que:
% s^2 + 0.1282*s + 0.02018705 = s^2 + s/Td + 1/Ti*Td
Td = 1/0.1282 % Td = 7.8003
Ti = 1/(0.02018705*Td) % Ti = 6.3506
PID = (s^2 + s/Td + 1/(Ti*Td))/(Td*Ti*s)
FdTLA = minreal(PID*Greal);
rlocus(FdTLA); sgrid(0.707, [20 40 60 80])

% Se escoge un Kp arbitrario igual a 1, por lo tanto:
FdTLC = minreal(feedback(FdTLA, 1));
step(FdTLC, G)
```