

Final control 2:

1) Considere el sistema a lazo cerrado con realimentación unitaria.

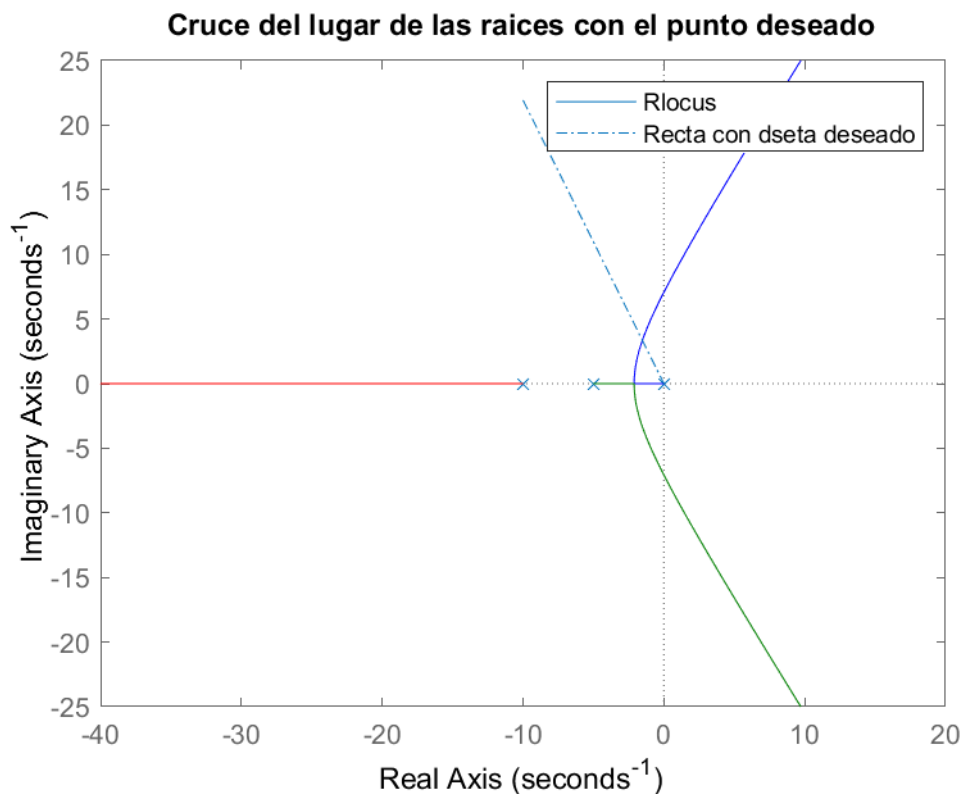
```
%% Examen final control 2
% Juan Agustin Avila
% Reg 26076
% DNI 35995356

%%Punto 1:
%Considere el sistema a lazo cerrado con realimentación
unitaria.

G=zpk([], [0 -5 -10], 1)
```

a) Hallar el valor de K para que la sobreelongación sea del 20%.

```
% Punto a: Halle el valor de K para que la sobreelongacion sea
del 20%
stepinfo(feedback(G,1)) %Obtengo las especificaciones temporales
OS=20; %defino el overshoot
a=-(1/pi)*log(20/100); %defino a para calcular dseta
dseta=a/(sqrt(1+a^2)); %calculo dseta
x=-10:0.1:0; %genero una recta con la pendiente
y=-1/dseta*x; %de dseta para ver el cruce
rlocus(G);hold on; %grafico el lugar de las raices
plot(x,y,'-.'); %Y luego grafico la recta
title("Cruce del lugar de las raices con el punto deseado");
legend("Rlocus","Recta con dseta deseado");
saveas(1,"lugardelasraices.png");
%sd=ginput(); %graficamente determino el punto de
cruce
%sd=sd(1)+sd(2)*i %y paso el valor obtenido a complejo
% El punto sd perteneciente al lugar de las raices es
sd = -1.5405 + 3.3070i;
%%
K=1/abs(evalfr(G,sd)); %Analizo el K, siendo la inversa de la
ganancia del
%lugar de las raices en ese punto
%K=K*.96; %para ajustar el K de ser necesario
R=rlocus(G,K); %lugar de los polos para la ganancia K,
observo que contiene
%a mi punto sd
G2=K*G; %Genero una nueva funcion con la ganancia K
```



b) Para el sistema con el valor de K hallado anteriormente determinar el tiempo del primer pico T_p y constante estática del error al escalón K_1 .

```

%% Punto b: Para el sistema con el valor de K hallado
anteriormente
% determinar el tiempo del primer pico  $T_p$  y constante
% estática del error al escalón  $K_1$ .
FTLC=feedback(G2,1); %Realimento la nueva funcion
info=stepinfo(FTLC); %Y analizo la respuesta al escalon
Tp=info.PeakTime; %Tp es el tiempo de pico
disp("El tiempo del primer pico es de "+Tp);
rampa=tf([1 0],1); %Hago una funcion con un cero en cero para
cancelar el polo y analizar
%la ganancia ante una entrada rampa
Kv=dcgain(minreal(G2*rampa)) %La planta es tipo 1, anulo el polo
en cero y analizo la ganancia estatica

```

El tiempo del primer pico es de 1.0472

$K_v = 3.1717$

2) Diseñar un compensador de adelanto-atraso con el lugar de las raíces que cumpla con las siguientes especificaciones:

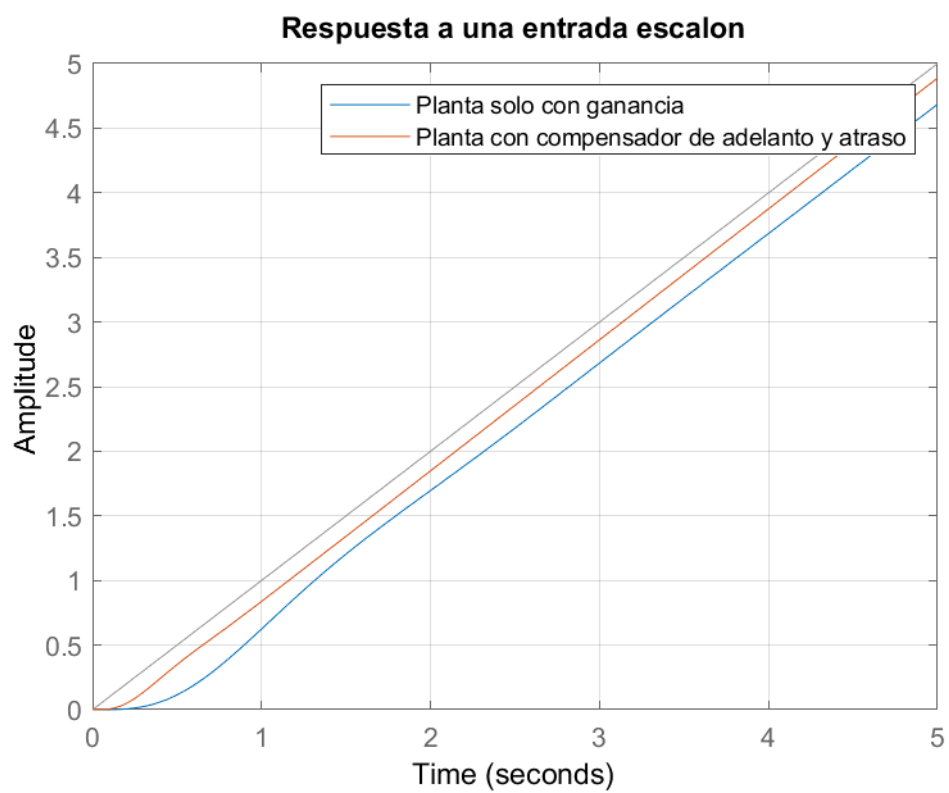
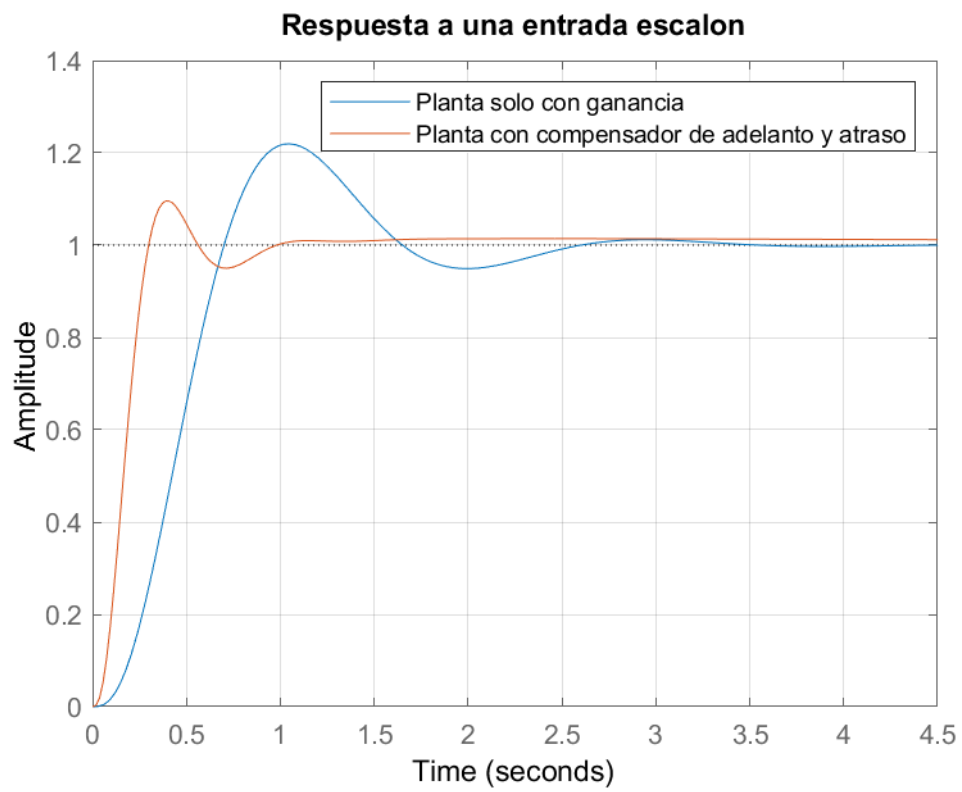
- Máxima sobreelongación: La misma obtenida en el punto 1 (20%).
- Tiempo de primer pico: Tres veces menor que el hallado en el punto 1.
- Constante estática del error al escalón: Cinco veces mayor la determinada en el punto 1.

```

%% Punto 2:Diseñar un compensador de adelanto-atraso con el
lugar de las
% raíces que cumpla con las siguientes especificaciones:
% a) Máxima sobreelongación: La misma obtenida en el punto 1
(20%).
% b) Tiempo de primer pico: Tres veces menor que el hallado en
el punto 1.
% c) Constante estática del error al escalón: Cinco veces mayor
la determinada en el punto 1.
Tpn=Tp/3;           %Hago que el nuevo Tp sea 3 veces menor
Kc=5*Kv;           %La nueva ganancia debe ser 5 veces mayor
%el dseta es el mismo que en el punto anterior
wn=pi/(Tpn*sqrt(1-dseta^2)); %wn cuando la especificacion es
el tiempo de pico
sd=-dseta*wn+abs((wn*sqrt(1-dseta^2)))*1i; %Calculo el nuevo
punto sd

%% No pude realizarlo con lgdl, asi que hice primero uno de
adelanto y luego uno de atraso
titamax=phase(sd)*180/pi;
fase=180-phase(evalfr(G,sd))*180/pi ; %obtengo la ganancia de
fase necesaria en el punto
Gplanta=abs(evalfr(G,sd)); %obtengo la ganancia en
el punto
titapolo=(titamax-fase)/2; %obtengo angulo del polo
titacero=(titamax+fase)/2; %obtengo angulo del cero
polo= real(sd)-imag(sd)/tan(titapolo*pi/180); %obtengo posicion
del polo
cero= real(sd)-imag(sd)/tan(titacero*pi/180);%obtengo posicion
del cero
pz=zpk(cero,polo,1); %armo la ft sin la ganancia
gpz=abs(evalfr(pz,sd)); %evaluo la ganancia para ese punto
Kc=abs(1/(gpz*Gplanta));%Y obtengo la ganancia del controlador
Gadelanto=pz*Kc
%compensador de atraso
alfa=K;
T=10;
Gatraso=zpk((-1/T),(-1/(alfa*T)),1)
Gcomp=Gadelanto*Gatraso
Gtotal=Gcomp*G;
FTLC=feedback(Gtotal,1);
%% pruebas de la planta con y sin compensador
stepinfo(FTLC)
[p z]=pzmap(FTLC)
figure(1),step(feedback(G2,1)),hold,step(FTLC),grid,title("Respu
esta a una entrada escalon")
legend('Planta solo con ganancia','Planta con compensador de
adelanto y atraso');
saveas(1,'Respuestaescalon.png');
x=0:.01:5;y=x;
figure(2),
lsim(feedback(G2,1),y,x),hold;
lsim(FTLC,y,x),grid,title("Respuesta a una entrada rampa");
legend('Planta solo con ganancia','Planta con compensador de
adelanto y atraso');
saveas(2,'Respuestarampa.png');

```



Gadelanto =

$$3270.9 (s+2.954)$$

(s+34.61)

Gatraso =

(s+0.1)

(s+0.0006306)

Gcomp =

3270.9 (s+2.954) (s+0.1)

(s+34.61) (s+0.0006306)

stepin =

struct with fields:

RiseTime: 0.1873

SettlingTime: 0.8783

SettlingMin: 0.9007

SettlingMax: 1.0959

Overshoot: 9.5920

Undershoot: 0

Peak: 1.0959

PeakTime: 0.3912

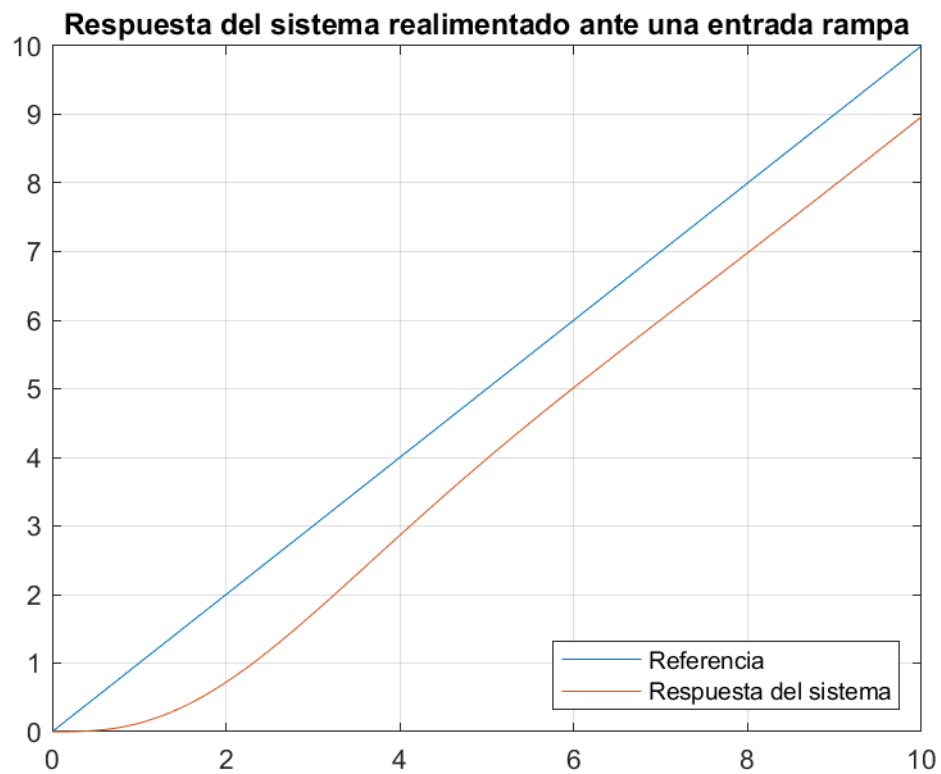
3) Para la misma planta diseñar en espacio de estados un servosistema tipo 1 con el esquema de Ogata con los mismos polos a lazo cerrado que el diseño anterior.

```
% Punto 3: servosistema tipo 1 con esquema de ogata
[A,B,C,D]=ssdata(G) %Obtengo las matrices de la planta G
Gcc1b=ss2ss(ss(G),obsv(ss(G))); %Convierto la planta G a su
representacion FCC1b
[Acc,Bcc,Ccc,Dcc]=ssdata(Gcc1b) %Obtengo las matrices de esta
planta
lambdas=[sd conj(sd) real(sd)*4]; %Uso el par complejo
conjugado SD
K=place(Acc,Bcc,lambdas)
Gcomp=ss(Acc-Bcc*K(1),Bcc*K(1),Ccc,Dcc) %genero el sistema con
la realimentacion de estados
t=0:.01:10;
```

```

y=t;
[Y,t,X]=lsim(Gcomp,y,t); %lo simulo para una entrada rampa
figure(3)
plot(t,y,t,Y);grid, title("Respuesta del sistema realimentado
ante una entrada rampa")
legend("Referencia","Respuesta del
sistema",'Location','SouthEast');
saveas(3,'realimestados.png')
tf(Gcomp);

```



K = 1.0e+03 *

1.8857 0.2223 0.0127

Gcomp =

A =

	x1	x2	x3
x1	0	1	0
x2	0	0	1
x3	-1886	-1936	-1901

B =

u1

x1 0

x2 0

x3 1886

C =

x1 x2 x3

y1 1 0 0

D =

u1

y1 0