Teoría de Control Realimentado

Curso: 2002/03 21.01.2003
Prueba Final Gerrit Färber

Programa Doctorado: Automatización Avanzada y Robótica
Instituto de Organización y control de Sistemas Industriales (IOC)

1. Sea el sistema dado por

$$\begin{aligned} \dot{x}_1 &= x_2 + u \\ \dot{x}_2 &= -(2+a)x_2 - 2a\,x_1 + u \\ y &= x_2 \end{aligned}$$

con $a \in \mathbb{R}$.

(a) Escribid la función de transferencia de u a y. Esta va a ser la planta para el resto de este ejercicio.

Solución:
Encontrado mediante la transformada de Laplace:

$$\Rightarrow \quad \begin{aligned} & s \cdot X_1(s) = X_2(s) + U(s) \\ & s \cdot X_2(s) = -(2+a) \cdot X_2(s) - 2a \cdot X_1(s) + U(s) \\ & Y(s) = X_2(s) \end{aligned}$$

$$\Rightarrow \quad X_1(s) = \tfrac{X_2(s)+U(s)}{s}$$

$$\Rightarrow \quad s \cdot X_2(s) = -(2+a) \cdot X_2(s) - 2a \cdot \tfrac{X_2(s)+U(s)}{s} + U(s)$$

$$\Rightarrow \quad X_2(s) = \frac{\dfrac{-2a+s}{s}}{\dfrac{s^2+(2+a)+2a}{s}} \cdot U(s)$$

$$\Rightarrow \quad P(s) = \frac{Y(s)}{U(s)} = \frac{-2a+s}{s^2+(2+a)+2a}$$

Solución de 1-a: $\boxed{P(s) = \dfrac{-2a+s}{s^2+(2+a)+2a}}$

(b) Realimentamos con el controlador

$$C(s) = \frac{1}{s(s+1)}$$

Calculad el rango de valores de a tal que el sistema es internamente estable. Encontrad el margen de ganancia superior en función de a .

Solución:
Mediante Maple:

```
P:=(-2*alpha+s)/(s^2+(2+alpha)*s+2*alpha);
```

$$P := \frac{-2\,\alpha + s}{s^2 + (2+\alpha)\,s + 2\,\alpha}$$

```
C:=1/(s^2+s);
```

$$C := \frac{1}{s^2 + s}$$

1-b-1: Calculo del rango de alpha:
Función de transferencia G del sistema con retroalimentación:

```
G:=simplify((C*P)/(1+P*C));
```

$$G := -\frac{2\,\alpha - s}{s^4 + 3\,s^3 + 2\,s^2 + s^3\,\alpha + 3\,s^2\,\alpha + 2\,s\,\alpha - 2\,\alpha + s}$$

Aplicando el Criterio de Routh-Hurwitz:

```
a1:=3+alpha:
a2:=3*alpha+2:
a3:=2*alpha+1:
a4:=-2*alpha:
a5:=0:

b1:=simplify(-(1*a3-a2*a1)/(a1));
```

$$b1 := \frac{9\,\alpha + 5 + 3\,\alpha^2}{3 + \alpha}$$

```
b2:=simplify(-(1*a5-a1*a4)/(a1));
```

$$b2 := -2\,\alpha$$

```
c1:=simplify(-(a1*b2-b1*a3)/(b1));
```

$$c1 := \frac{37\,\alpha + 33\,\alpha^2 + 8\,\alpha^3 + 5}{9\,\alpha + 5 + 3\,\alpha^2}$$

```
c2:=0:
d1:=simplify(-(b1*c2-c1*b2)/(c1));
```

$$d1 := -2\,\alpha$$

Condición de Routh-Hurwitz con a0 > 0 (a0 =1)

a1, b1, c1, d1 >0

=> con a1 y d1: -3<alpha<0 (rango preliminar)

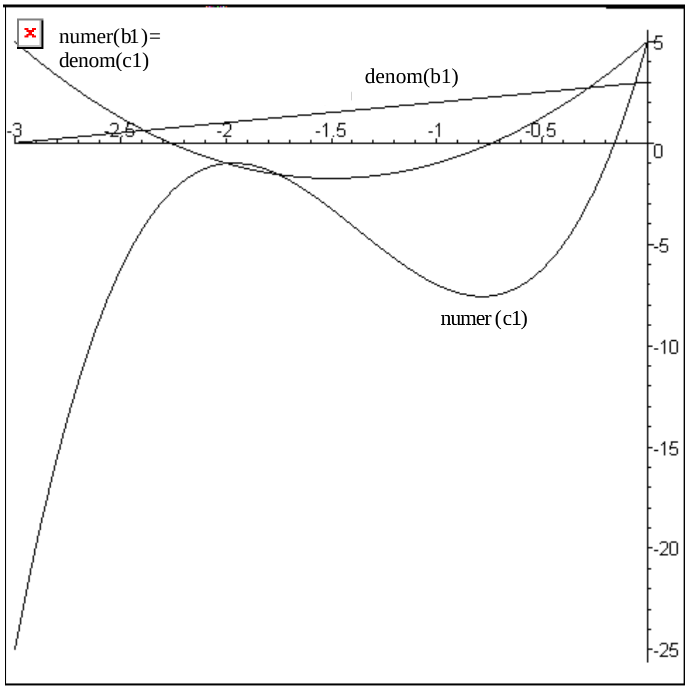
Aplicando este rango preliminar para dibujar el grafo de los denominadores y de los numeradores de b1 y de c1:

Según b1 y c1 alpha es valido si:

(denom(b1)>0 y numer(b1)>0) y (denom(c1)>0 y numer(c1)>0) o si

(denom(b1)<0 y numer(b1)<0) y (denom(c1)<0 y numer(c1)<0)

```
plot({denom(b1),numer(b1),denom(c1),numer(c1)},alpha=-3..0);
```



El grafico muestra que es suficiente encontrar el cero del numerador de c1 que está alrededor de -0.15, el cero tiene que ser real.

```
evalf(solve(numer(c1)));
```

-0.156026264 , -1.984486868 + 0.2598994556 *I*, -1.984486868 − 0.2598994556 *I*

=> rango de alpha:

-0.156026264 < alpha < 0

Solución de 1-b-1: <u>-0.156026264 < alpha < 0</u>

1-b-2: Calculao de margen de ganancia:
Multiplicando P con la variable k, se obtiene una nueva Función de Transferencia Gk:

```
Gk:=simplify((C*k*P)/(1+k*P*C));
```

$$Gk := -\frac{(2\,\alpha - s)\,k}{s^4 + 3\,s^3 + 2\,s^2 + s^3\,\alpha + 3\,s^2\,\alpha + 2\,s\,\alpha - 2\,k\,\alpha + k\,s}$$

```
s:=I*w:
```

```
Gk;
```

$$-\frac{(2\,\alpha - w\,I)\,k}{w^4 - 3\,I\,w^3 - 2\,w^2 - w^3\,\alpha\,I - 3\,w^2\,\alpha + 2\,I\,w\,\alpha - 2\,k\,\alpha + k\,w\,I}$$

Encontrar una k tal que la parte imaginaria de Gk sea cero y la parte real sea -1:

```
Gk_re:=sort(simplify(evalc(Re(Gk))),w);
```

$$Gk\_re := k\,(-3\,\alpha\,w^4 - 3\,w^4 + 6\,\alpha\,w^2 + 6\,\alpha^2\,w^2 + k\,w^2 + 4\,k\,\alpha^2) \Big/ (w^8 + 5\,w^6 + \alpha^2\,w^6 - 6\,k\,\alpha\,w^4 + 4\,w^4 - 6\,k\,w^4 + 5\,\alpha^2\,w^4 + 12\,\alpha^2\,k\,w^2 + k^2\,w^2 + 12\,k\,\alpha\,w^2 + 4\,\alpha^2\,w^2 + 4\,k^2\,\alpha^2)$$

```
Gk_im:=sort(simplify(evalc(Im(Gk))),w);
```

$$Gk\_im := -k\,(-w^4 + 2\,w^2 + 9\,\alpha\,w^2 + 2\,\alpha^2\,w^2 - 4\,\alpha^2)\,w \Big/ (w^8 + 5\,w^6 + \alpha^2\,w^6 - 6\,k\,\alpha\,w^4 + 4\,w^4 - 6\,k\,w^4 + 5\,\alpha^2\,w^4 + 12\,\alpha^2\,k\,w^2 + k^2\,w^2 + 12\,k\,\alpha\,w^2 + 4\,\alpha^2\,w^2 + 4\,k^2\,\alpha^2)$$

Encontrar los ceros de la parte imaginaria de Gk:

```
Gk_im_num:=numer(Gk_im);
```

$$Gk\_im\_num := -k\,w\,(-w^4 + 2\,w^2 + 9\,\alpha\,w^2 + 2\,\alpha^2\,w^2 - 4\,\alpha^2)$$

y aplicar los ceros a la función que representa la parte real de Gk:

```
zeros_Gk:=solve(Gk_im_num,w);
```

$$zeros\_Gk := 0, \frac{\sqrt{4 + 2\sqrt{4 + 36\,\alpha + 73\,\alpha^2 + 36\,\alpha^3 + 4\,\alpha^4} + 18\,\alpha + 4\,\alpha^2}}{2}, -\frac{\sqrt{4 + 2\sqrt{4 + 36\,\alpha + 73\,\alpha^2 + 36\,\alpha^3 + 4\,\alpha^4} + 18\,\alpha + 4\,\alpha^2}}{2}, \frac{\sqrt{4 - 2\sqrt{4 + 36\,\alpha + 73\,\alpha^2 + 36\,\alpha^3 + 4\,\alpha^4} + 18\,\alpha + 4\,\alpha^2}}{2}, -\frac{\sqrt{4 - 2\sqrt{4 + 36\,\alpha + 73\,\alpha^2 + 36\,\alpha^3 + 4\,\alpha^4} + 18\,\alpha + 4\,\alpha^2}}{2}$$

Seguimos solamente con "valor_Gk_re[2]" y "valor_Gk_re[4]", porque los demás son positivos o están incluidos.

```
evala(valor_Gk_re[1]);
```

$$1$$

```
evala(valor_Gk_re[2]);
```

$$2\,k/(2\,k-2\,\alpha^3-15\,\alpha^2-\sqrt{4+36\,\alpha+73\,\alpha^2+36\,\alpha^3+4\,\alpha^4}\,\alpha-25\,\alpha-3\sqrt{4+36\,\alpha+73\,\alpha^2+36\,\alpha^3+4\,\alpha^4}-6)$$

```
evala(valor_Gk_re[3]);
```

$$2\,k/(2\,k-2\,\alpha^3-15\,\alpha^2-\sqrt{4+36\,\alpha+73\,\alpha^2+36\,\alpha^3+4\,\alpha^4}\,\alpha-25\,\alpha-3\sqrt{4+36\,\alpha+73\,\alpha^2+36\,\alpha^3+4\,\alpha^4}-6)$$

```
evala(valor_Gk_re[4]);
```

$$2\,k/(2\,k-2\,\alpha^3-15\,\alpha^2+\sqrt{4+36\,\alpha+73\,\alpha^2+36\,\alpha^3+4\,\alpha^4}\,\alpha-25\,\alpha+3\sqrt{4+36\,\alpha+73\,\alpha^2+36\,\alpha^3+4\,\alpha^4}-6)$$

```
evala(valor_Gk_re[5]);
```

$$2\,k/(2\,k-2\,\alpha^3-15\,\alpha^2+\sqrt{4+36\,\alpha+73\,\alpha^2+36\,\alpha^3+4\,\alpha^4}\,\alpha-25\,\alpha+3\sqrt{4+36\,\alpha+73\,\alpha^2+36\,\alpha^3+4\,\alpha^4}-6)$$

Al final el número de funciones de k son dos:

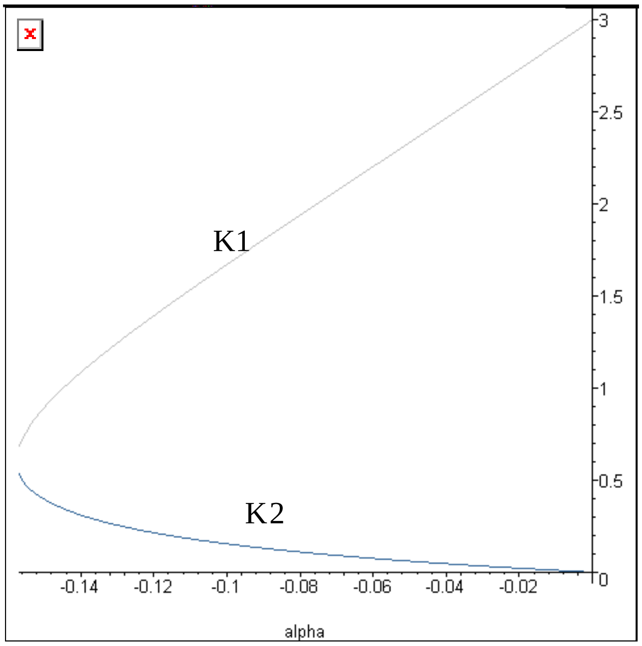
```
k1:=solve(evala(valor_Gk_re[2])=-1,k);
```

$$k1 := \frac{\alpha^3}{2}+\frac{15\,\alpha^2}{4}+\frac{\sqrt{4+36\,\alpha+73\,\alpha^2+36\,\alpha^3+4\,\alpha^4}\,\alpha}{4}+\frac{25\,\alpha}{4}+\frac{3}{2}+\frac{3\sqrt{4+36\,\alpha+73\,\alpha^2+36\,\alpha^3+4\,\alpha^4}}{4}$$

```
k2:=solve(evala(valor_Gk_re[4])=-1,k);
```

$$k2 := \frac{\alpha^3}{2}+\frac{15\,\alpha^2}{4}-\frac{\sqrt{4+36\,\alpha+73\,\alpha^2+36\,\alpha^3+4\,\alpha^4}\,\alpha}{4}+\frac{25\,\alpha}{4}+\frac{3}{2}-\frac{3\sqrt{4+36\,\alpha+73\,\alpha^2+36\,\alpha^3+4\,\alpha^4}}{4}$$

```
plot({k1,k2},alpha=-0.157..0);
```



=> Dibujando k1 y k2 en el rango de alpha se encuentra que k2 siempre es mas pequeño que k1, por lo tanto la margen de ganancia es la función k2.

Solución de 1-b-2:

```
k2;
```

$$\frac{\alpha^3}{2}+\frac{15\,\alpha^2}{4}-\frac{\sqrt{4+36\,\alpha+73\,\alpha^2+36\,\alpha^3+4\,\alpha^4}\,\alpha}{4}+\frac{25\,\alpha}{4}+\frac{3}{2}-\frac{3\sqrt{4+36\,\alpha+73\,\alpha^2+36\,\alpha^3+4\,\alpha^4}}{4}$$

(c) Sea α =-2. Diseñad un controlador tal que el sistema realimentado sea internamente estable y capaz de seguir señales en el rango de frecuencias angulares hasta 10 Hz con un error menor del 1%. Comprobad el controlador obtenido mediante simulación con diferentes señales y observad su degradación fuera del ancho de banda especificado.

Solución:

Aplicando alpha= -2 a la planta:

```
alpha:=-2:
P:=(-2*alpha+s)/(s^2+(2+alpha)*s+2*alpha);
```

$$P := \frac{4+s}{s^2-4}$$

```
s:=(1-p)/p:
P:=simplify(P);
```

$$P := -\frac{(3\,p+1)\,p}{-1+2\,p+3\,p^2}$$

Factorización Coprima:

```
n:=numer(P);
m:=denom(P);
```

$$n := -(3\,p+1)\,p$$

$$m := -1+2\,p+3\,p^2$$

```
r_1:=rem(m,n,p,q_1);q_1:=q_1;
```

$$r\_1 := -1+p$$

$$q\_1 := -1$$

```
r_2:=rem(n,r_1,p,q_2);q_2:=q_2;
```

$$r\_2 := -4$$

$$q\_2 := -3\,p-4$$

```
p:=1/(x+1);
term:=1/r_2*(N-q_2*(M-N*q_1)):
simplify (term);
```

$$-\frac{5\,N\,x+8\,N+7\,M+4\,x\,M}{4\,(x+1)}$$

```
N:=n:N:=simplify(N);
```

$$N := -\frac{4+x}{(x+1)^2}$$

```
M:=m:M:=simplify(M);
```

$$M := -\frac{x^2-4}{(x+1)^2}$$

Determinación de X y Y con (term= Y(s) * M(s) +X(s) * N(s)):

```
Y:=(-4*x-7)/(4*(x+1));
```

$$Y := \frac{-7-4x}{4x+4}$$

```
X:=(-5*x-8)/(4*(x+1));
```

$$X := \frac{-5x-8}{4x+4}$$

```
prueba:=simplify(Y*M+X*N);
```

$$prueba := 1$$

```
assign('s=s');assign('x=s');
```

Definimos W1 para que el controlador controle la planta hasta una frecuencia de 10Hz con un error máximo de 1% :

```
W1:=100/(1/20*Pi*s+1);
```

$$\mathrm{W1} := \frac{100}{\frac{\pi s}{20}+1}$$

La Norma infinita de W1*M*Y*(1-J[tau]) tiene que ser mas pequeño que 1.

```
calculo:=simplify(W1*M*Y*(1-J[tau]));
```

$$calculo := -\frac{500\,(s^2-4)\,(7+4s)\,(-1+J_\tau)}{(\pi s+20)\,(s+1)^3}$$

```
tau:=0.00155:  n:=1:
J[tau]:=(1/(s*tau+1)^n);
```

$$J_{0.00155} := \frac{1}{0.00155\;s+1}$$

```
Lsg:=calculo:
Lsg_p:=simplify(eval(Lsg,s=w*I)):
Lsg_n:=simplify(eval(Lsg,s=-w*I)):
Lsg_abs:=simplify(Lsg_n*Lsg_p):
Lsg_diff:=simplify(diff(Lsg_abs,w)):
Lsg_diff_z:=vector(17):
Lsg_diff_z:=solve(Lsg_diff,w):
Lsg_0:=0:
for k to 13 do
  Lsg_01[k]:=(eval(Lsg_abs,w=Lsg_diff_z[k])):
  if ((Lsg_0)<(Lsg_01[k])) then Lsg_0:=Lsg_01[k] end if;
end do:
Lsg_0;
```

$$0.9911463304$$

```
limit(Lsg_abs, w=infinity);
```

$$0.$$

Determinación del controlador, estabilizando y controlando la planta hasta 10Hz con un error máximo de 1%

```
assign('s=s');
Q:=1/N*Y*J[tau];
```

$$Q := -\frac{(s+1)^2\,(-7-4\,s)}{(4+s)\,(4\,s+4)\,(0.00155\ \ s+1)}$$

```
C:=sort(simplify((X+M*Q)/(Y-N*Q)),s);
```

$$C := \frac{0.03225806452(80155.\,s^3+240868.\,s^2+240992.\,s+80000.)}{(4.\,s+7.)\,(s+4.)\,s}$$

Evaluación del numerador y del denominador de C para la simulación en Matlab:

```
sort(expand(numer(C)),s);
```

$$2585.645162\,s^3+7769.935485\,s^2+7773.935485\,s+2580.645162$$

```
sort(expand(denom(C)),s);
```

$$4.\,s^3+23.\,s^2+28.\,s$$

Simulación con Matlab:

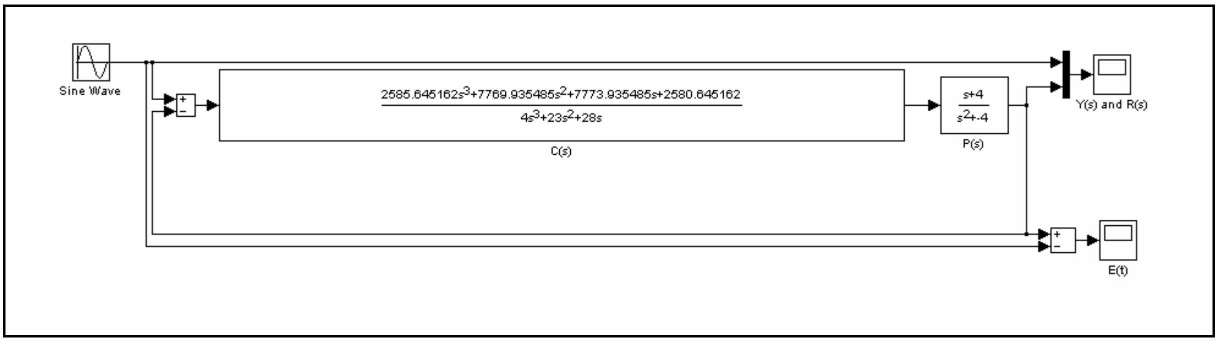
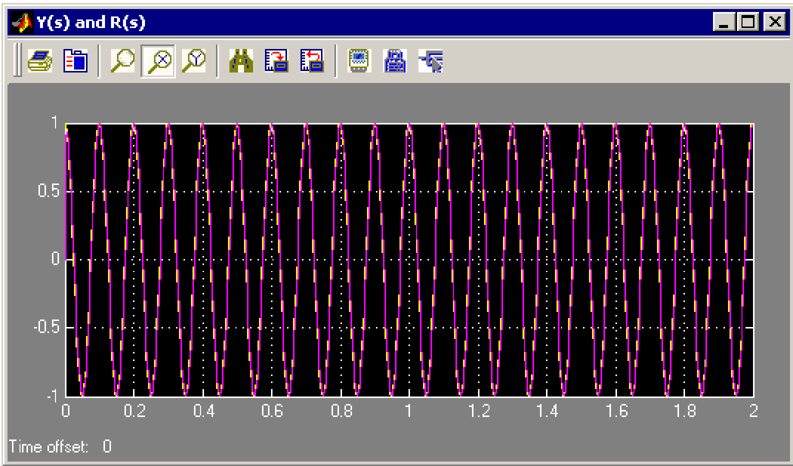
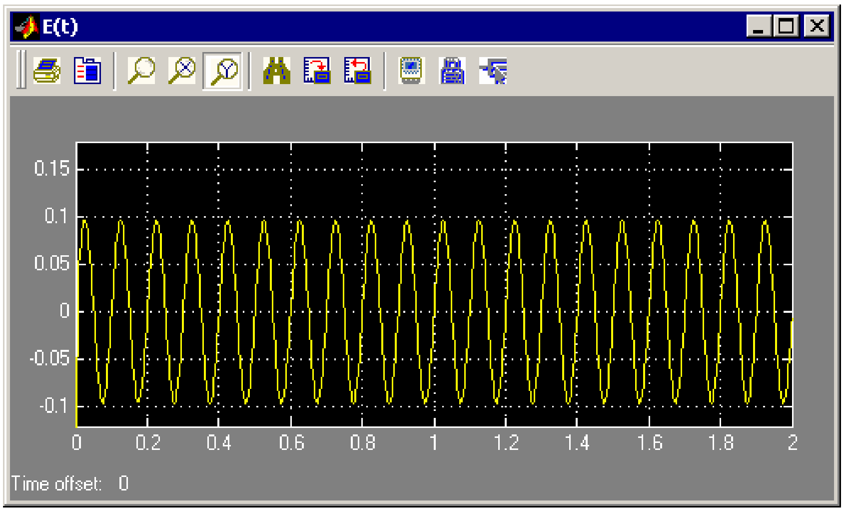


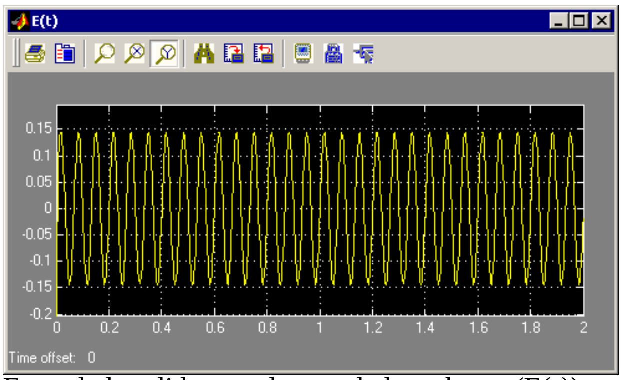
Diagrama de bloques de la planta con el controlador robusto.

El controlador controla la planta hasta una frecuencia de 10Hz con un error máximo de 1%:

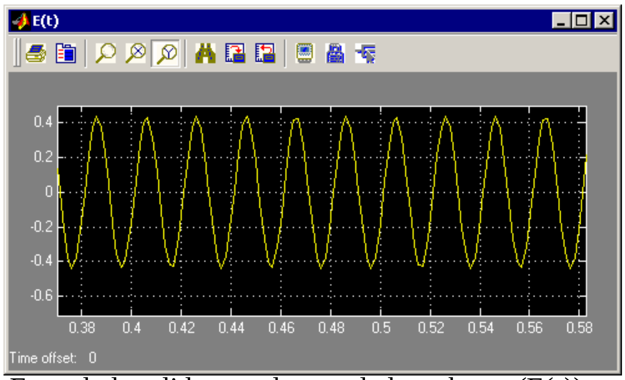


Salida de la simulación con el controlador robusto (Y(s) y R(s))
(f=10Hz)

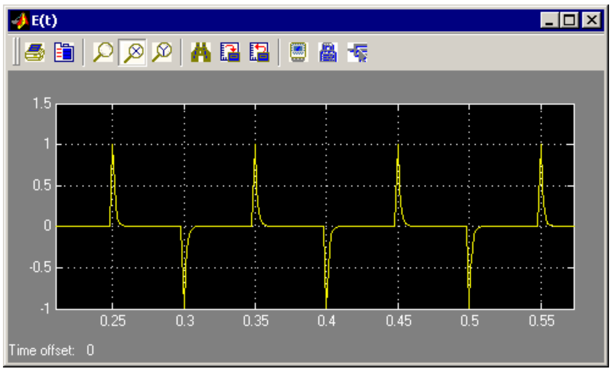




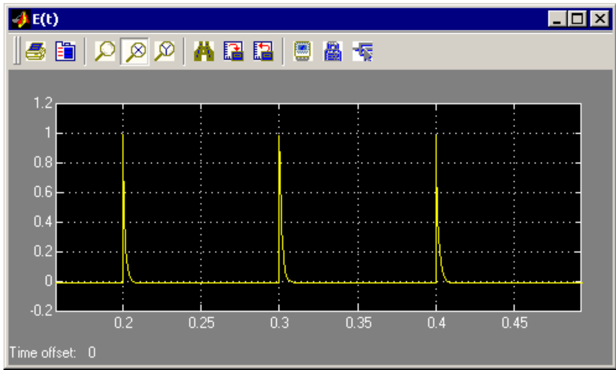
Error de la salida con el controlador robusto (E(s)).
(f = 15Hz)



Error de la salida con el controlador robusto (E(s)).
(f = 50Hz)



Error de la salida con el controlador robusto (E(s)).
(entrada: cuadrada con un periodo de T=0.1 seg)



Error de la salida con el controlador robusto (E(s)).
(entrada: triangular con un periodo de T=0.1 seg)

El controlador, controla la planta para señales senoidales hasta la frecuencia de 10Hz con un error máximo de 1%. Aplicando frecuencias mas grandes, el error aumenta.
Aplicando señales no senoidales como cuadrada o triangular, el error aumenta hasta 100%. Esto es por que las señales con saltos tienen frecuencias altas en donde está el salto.

2. Sea un modelo de incertidumbre multiplicativa de disco con planta nominal

$$P(s) = \frac{1}{1-s}$$

y función de incertidumbre

$$W_2(s) = \frac{a \cdot s}{0.01 \cdot s + 1}$$

con a > 0.

(a) Calculad el supremo de los valores de a tales que se puede conseguir estabilidad robusta.

Solución:

```
P:=1/(s-1):
W2:=a*s/(0.01*s+1):
C:=(s-1)/(s+1);
```

$$C := \frac{s-1}{s+1}$$

La norma infinita de (W2 x T) tiene que ser más pequeña que 1:

```
T:=simplify(P*C/(1+P*C));
```

$$T := \frac{1}{s+2}$$

```
prod:=W2*T;
```

$$prod := \frac{a\ s}{(0.01\ s+1)\,(s+2)}$$

Calculo de la norma infinita:

```
s:=w*I:
prod_p:=prod:
s:=-w*I:
prod_n:=prod:
prodabs:=simplify(prod_p*prod_n);
```

$$prodabs := \frac{10000.\ a^2\ w^2}{(-1.\ w^2 - 10000.\ )\,(-1.\ w^2 - 4.\ )}$$

```
prod_diff:=diff(prodabs,w);
```

$$prod\_diff := \frac{20000.\ a^2\ w}{(-1.\ w^2 - 10000.\ )\,(-1.\ w^2 - 4.\ )} + \frac{20000.\ a^2\ w^3}{(-1.\ w^2 - 10000.\ )^2\,(-1.\ w^2 - 4.\ )}$$
$$+ \frac{20000.\ a^2\ w^3}{(-1.\ w^2 - 10000.\ )\,(-1.\ w^2 - 4.\ )^2}$$

Los ceros de la derivada son los siguientes, con tres valores reales:

```
zero:=solve(prod_diff,w);
```

$$zero := 0., 14.14213562\ I, -14.14213562\ I, 14.14213562\ , -14.14213562$$

```
w:=zero[1]: simplify(prodabs);
```

$$0.$$

```
w:=zero[4]: test1:=simplify(prodabs);
```

$$test1 := 0.9611687814\ a^2$$

```
w:=zero[5]: test2:=simplify(prodabs);
```

$$test2 := 0.9611687814 a^2$$

```
solve(test1=1);solve(test2=1);
```

$$1.020000000\ , -1.020000000$$

$$1.020000000\ , -1.020000000$$

=> La norma subinfinita debe ser mas pequeña que 1
y positivo => El valor supremo de "a" es 1.02

Solución de 2-a: a =< 1.02

(b) Seleccionad un valor de a inferior al obtenido en el apartado anterior y diseñad un controlador robusto.

Solución:
Tomando "a": a=1.005 y cambiando el único cero de W2 para que T2 no tenga un cero en el eje imaginario.

```
P:=1/(s-1):
W2:=a*(s+1e-8)/(0.01*s+1);
```

$$W2 := \frac{a\ (s + 0.1\ 10^{-7})}{0.01\ s + 1}$$

```
a:=1.005;
```

$$a := 1.005$$

Step-1: Factorización Coprima:

```
s:=(1-p)/p:
P:=simplify(P);
```

$$P := -\frac{p}{-1 + 2\ p}$$

```
n:=numer(P);
m:=denom(P);
```

$$n := -p$$

$$m := -1 + 2\ p$$

```
r_1:=rem(m,n,p,q_1);q_1:=q_1;
```

$$r\_1 := -1$$

$$q\_1 := -2$$

```
p:=1/(x+1);
```

$$p := \frac{1}{x+1}$$

```
term:=1/r_1*(M-N*q_1):
simplify (term);
```

$$-M - 2\,N$$

```
N:=n:N:=simplify(N);
```

$$N := -\frac{1}{x+1}$$

```
M:=m:M:=simplify(M);
```

$$M := -\frac{x-1}{x+1}$$

Determinación de X y Y con (term= Y(s) * M(s) +X(s) * N(s)):

```
Y:=-1;
```

$$Y := -1$$

```
X:=-2;
```

$$X := -2$$

Step-2: Solución del problema model-matching:

```
T1:=simplify(W2*N*X);
```

$$T1 := \frac{0.2010000000\ \ 10^{-5}\,(0.100000000\ \ 10^{9}\,x + 1.)}{x^2 + 101.\ x + 100.}$$

```
T2:=simplify(-W2*N*M);
```

$$T2 := -\frac{0.1005000000\ \ 10^{-5}\,(x - 1.)\,(0.100000000\ \ 10^{9}\,x + 1.)}{(x^2 + 101.\ x + 100.\ )\,(x + 1.\ )}$$

Ceros de T2:

```
z:=solve(T2);
```

$$z := 1.,\ -0.1000000000\ \ 10^{-7}$$

Evaluado en T1:

```
b:=eval(T1,x=z[1]);
```

$$b := 0.9950495149$$

```
A:=simplify(1/(z[1]+Re(z[1])-Im(z[1])*I));
```

$$A := 0.5000000000$$

```
B:=simplify(b*conjugate(b)/(z[1]+conjugate(z[1])));
```

$$B := 0.4950617686$$

```
Gam_opt:= A^(-1/2)*B*A^(-1/2);
```

$$Gam\_opt := 0.9901235372$$

Solución de NP:

```
c:=z[1];
d:=b*1/Gam_opt;
```

$$c := 1.$$

$$d := 1.004975114$$

```
Aa:=(x-c)/(x+c);
```

$$Aa := \frac{x-1.}{x+1.}$$

```
G1:=1;
```

$$G1 := 1$$

```
Mb:=(Z+d)/(1+Z*d);
```

$$Mb := \frac{Z + 1.004975114}{1 + 1.004975114\ Z}$$

```
Z:=G1*Aa;
```

$$Z := \frac{x-1.}{x+1.}$$

```
G:=simplify(Mb);
```

$$G := \frac{0.1002487557\ 10^{10}\ x + 0.2487557\ 10^{7}}{0.1002487557\ 10^{10}\ x - 0.2487557\ 10^{7}}$$

Determinación de Q_im:

```
Qim:=(simplify((T1-Gam_opt*G)/T2));
```

$$Qim := 0.00009950248756\ (x+1.)\,(-0.1012462969\ 10^{22}\ x^2 + 0.1000074114\ 10^{22}\ x + 0.2462988786\ 10^{19} + 0.9925865258\ 10^{19}\ x^3) \Big/ ((0.100000000\ 10^{9}\ x + 1.)\ (x-1.)\,(0.1002487557\ 10^{10}\ x - 0.2487557\ 10^{7}))$$

Prueba de Gamma_optima:

La norma infinita de (T1-T2*Q)/Gam_opt debe ser mas pequeña que 1.

```
x:=w*I:
Lsg:=(T1-T2*Qim)/Gam_opt:
Lsg_Re:=simplify(evalc(Re(Lsg))):
Lsg_Im:=simplify(evalc(Im(Lsg))):
Lsg_abs:=simplify(sqrt(Lsg_Re^2+Lsg_Im^2)):
Lsg_diff:=simplify(diff(Lsg_abs,w))=0:
Lsg_diff_z:=vector(17):
Lsg_diff_z:=solve(Lsg_diff,w):
Lsg_0:=0:
for k to 17 do
  Lsg_01[k]:=abs(eval(Lsg_abs,w=Lsg_diff_z[k])):
  if ((Lsg_0)<(Lsg_01[k])) then Lsg_0:=Lsg_01[k] end if;
end do:
Lsg_0;
```

$$1.000000000$$

```
limit(Lsg_abs, w=infinity);
```

$$0.9999999992$$

Step-3: Elección de J[tau] tal que Q_im x J[tau] es propio y tau suficientemente pequeño de tal manera que (W2*N*(X+M*Qim*J[tau])) es mas pequeño que Gamma_opt:

```
n:=1:
tau:=0.00002:
J[tau]:=1/(tau*s+1)^n;
```

$$J_{0.00002} := \frac{1}{0.00002\ s + 1}$$

```
x:=s:
Lsg:=factor(simplify(W2*N*(X+M*Qim*J[tau]))):
s:=w*I:
Lsg_p:=Lsg:
s:=-w*I:
Lsg_n:=Lsg:
Lsg_abs:=Lsg_p*Lsg_n:
Lsg_diff:=simplify(diff(Lsg_abs,w))=0:
Lsg_diff_z:=vector(30):
Lsg_diff_z:=solve(Lsg_diff,w):
Lsg_0:=0:
for k to 9 do
  Lsg_01[k]:=abs(eval(Lsg_abs,w=Lsg_diff_z[k])):
  if ((Lsg_0)<(Lsg_01[k])) then Lsg_0:=Lsg_01[k] end if;
end do:
Lsg_0;
```

$$0.9886823459$$

```
limit(Lsg_abs, w=infinity);
```

$$0.$$

Calculo de Q=Q_im x J[tau]:

```
assign('s=s');
Q:=Qim*J[tau];
```

$$Q := 0.00009950248756\ (s + 1.)\ (-0.1012462969\ 10^{22}\ s^2 + 0.1000074114\ 10^{22}\ s + 0.2462988786\ 10^{19} + 0.9925865258\ 10^{19}\ s^3) \Big/ ((0.100000000\ 10^{9}\ s + 1.)\ (s - 1.)\ (0.1002487557\ 10^{10}\ s - 0.2487557\ 10^{7})\ (0.00002\ s + 1))$$

Calculación de C=(X+M*Q)/(Y-N*Q):

```
C:=sort(simplify((X+M*Q)/(Y-N*Q)),s);
```

$$C := 2.\ (s - 1.)\ (0.2479145586\ 10^{20}\ s^3 + 0.2493872937\ 10^{22}\ s^2 + 0.2475308817\ 10^{22}\ s + 0.6126837652\ 10^{19}) \Big/ (0.1002487557\ 10^{18}\ s^4 + 0.4962954873\ 10^{22}\ s^3 + 0.1225392777\ 10^{20}\ s^2 - 0.4963055370\ 10^{22}\ s - 0.1225367543\ 10^{20})$$

Evaluación del numerador y del denominador de C para la simulación en Matlab:

```
expand(numer(C));
```

$$0.4958291172\ 10^{20}\ s^4 + 0.4938162962\ 10^{22}\ s^3 - 0.37128240\ 10^{20}\ s^2 - 0.4938363958\ 10^{22}\ s - 0.1225367530\ 10^{20}$$

```
expand(denom(C));
```

$$0.1002487557\ 10^{18}\ s^4 + 0.4962954873\ 10^{22}\ s^3 + 0.1225392777\ 10^{20}\ s^2 - 0.4963055370\ 10^{22}\ s - 0.1225367543\ 10^{20}$$

Simulación con Matlab:

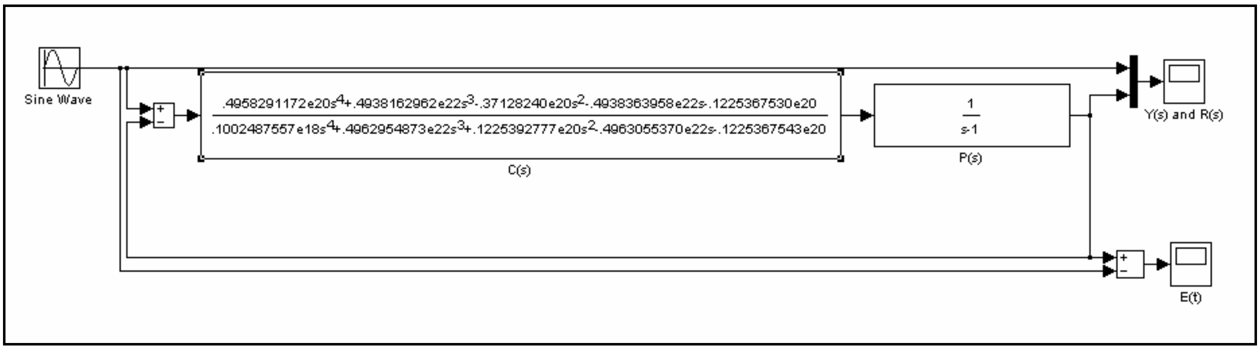
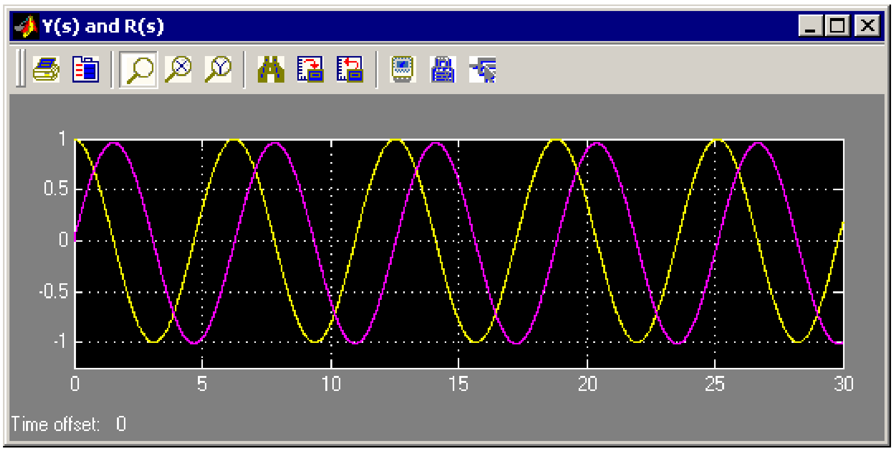


Diagrama de bloque de la planta con el controlador robusto.



Salida del la simulación con el controlador robusto (Y(s) y R(s)).

(c) Para el valor de *a* escogido, calculad una planta, distinta de la nominal, que pertenezca al modelo de incertidumbre. Para dicha planta, comprobad mediante simulación la bondad del controlador obtenido.

Solución:

La variación de la planta se calcula mediante una función Delta que tiene su norma infinita mas pequeña que 1.

Planta original:

```
P:=1/(s-1):
a:=1.005:
W2:=a*s/(0.01*s+1):
```

Función Delta con norma infinita mas pequeña que 1:

```
Delta:=1/(s+3);
```

$$\Delta := \frac{1}{s+3}$$

Prueba que la función Delta tiene su norma infinita mas pequeña que 1:

```
Lsg:=Delta:
s:=w*I:
Lsg_p:=Lsg:
s:=-w*I:
Lsg_n:=Lsg:
Lsg_abs:=Lsg_p*Lsg_n:
Lsg_diff:=simplify(diff(Lsg_abs,w))=0:
Lsg_diff_z:=vector(30):
Lsg_diff_z:=solve(Lsg_diff,w):
Lsg_0:=0:
for k to 9 do
       Lsg_01[k]:=abs(eval(Lsg_abs,w=Lsg_diff_z[k])):
  if ((Lsg_0)<(Lsg_01[k])) then Lsg_0:=Lsg_01[k] end if;
end do:
Lsg_0;
```

$$\frac{1}{9}$$

```
limit(Lsg_abs, w=infinity);
```

$$0$$

Calculo de la Planta nueva (Planta_tilde):

```
assign('s=s');
P_tilde:=simplify((1+Delta*W2)*P);
```

$$P\_tilde := \frac{0.5000000000\ (2.\ s^2 + 407.\ s + 600.)}{(s+3.)\,(s+100.)\,(s-1.)}$$

Evaluación del numerador y del denominador de P_tilde para la simulación en Matlab:

```
sort(expand(numer(P_tilde)),s);
```

$$1.000000000\ s^2 + 203.5000000\ s + 300.0000000$$

```
sort(expand(denom(P_tilde)),s);
```

$$s^3 + 102.\ s^2 + 197.\ s - 300.$$

Simulación con Matlab:

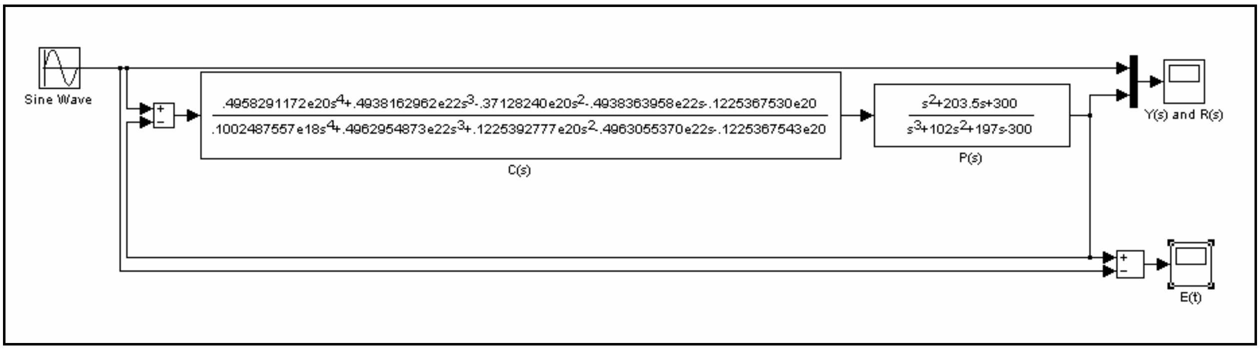
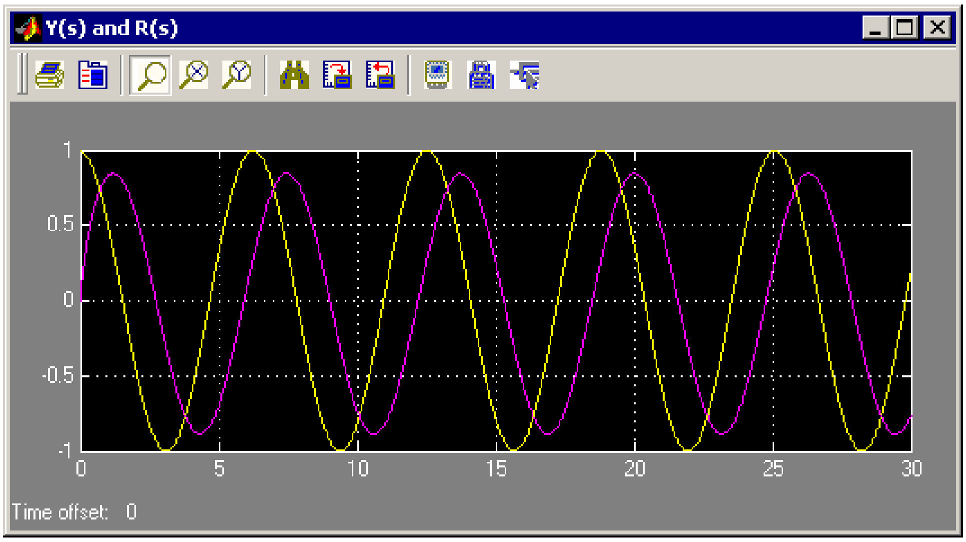
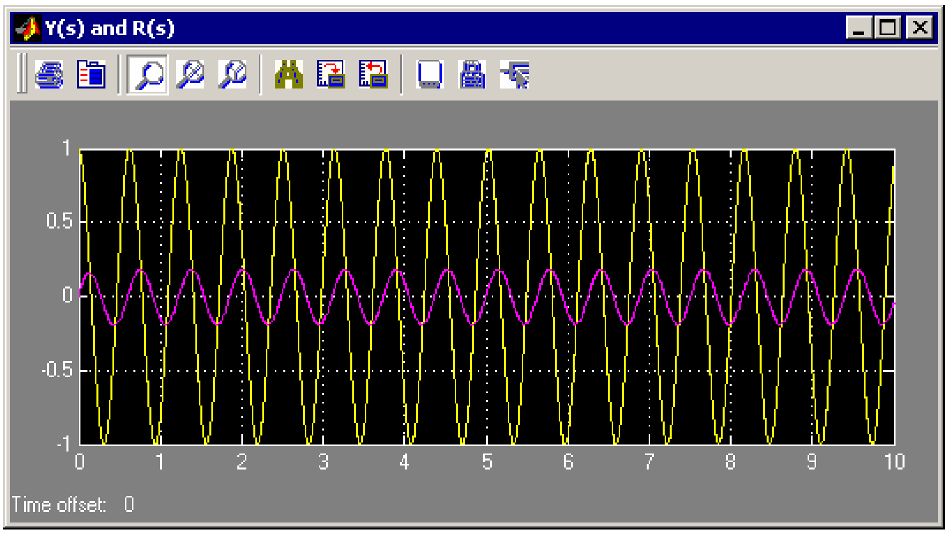


Diagrama de bloques de la planta con el controlador robusto.



Salida del la simulación con el controlador robusto (Y(s) y R(s)) con una frecuencia baja.



Salida del la simulación con el controlador robusto (Y(s) y R(s)) con una frecuencia mas alta. En este caso, la perturbación de la salida es más pequeña, pero sigue seyendo estable.