

Cancelación de perturbaciones deterministas por asignación de polos.

Seguimiento de referencia con 1 grado libertad.

(c) 2018, Antonio Sala Piqueras, Universitat Politècnica de València. Todos los derechos reservados.

Presentación en vídeo en <http://personales.upv.es/asala/YT/V/drconstrml.html>

Modelado

modelo del proceso (masa-muelle-amortiguador):

```
A=[0 1;-5 -1]; B=[0;2];C=[1 0];D=0;  
sys=ss(A,B,C,D);  
tf(sys)
```

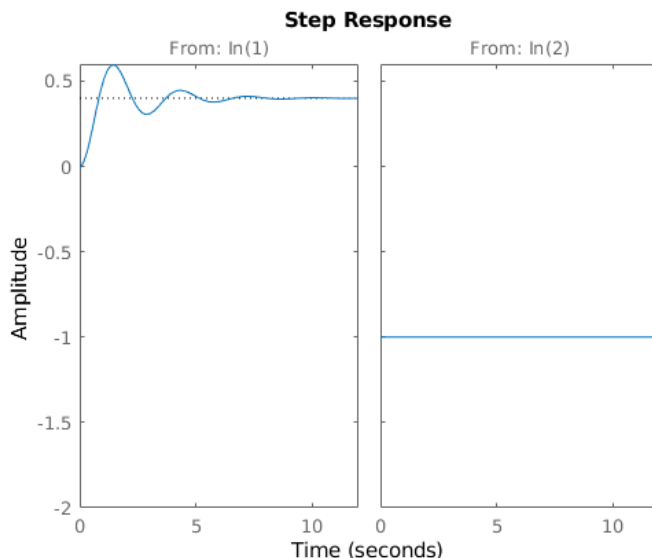
ans =

```
      2  
-----  
s^2 + s + 5
```

Continuous-time transfer function.

efecto de las perturbaciones sobre el estado:

```
F=[0;0]; E=-1;%efecto en la salida, se desea y-r=0.  
%sólo hay un "C": objetivos de control y sensores idénticos.  
%modelo con 2 entradas:  
syspert=ss(A,[B F],C,[D E]);  
%las entradas son u, ref; la salida es error Cx-ref.  
step(syspert)
```



Cancelación de perturbaciones si fuera conocida:

```
Qu=inv(dcgain(sys)) %la perturbación r se cancela con u=inv(ganancia)*r,
```

```
Qu = 2.5000
```

```
%el estado final, iria a:
```

```
Qx=-inv(A)*B*Qu
```

```
Qx =  
    1  
    0
```

Modelo ampliado:

```
Abig=[A F;zeros(1,2) 0]
```

```
Abig =  
    0     1     0  
   -5    -1     0  
    0     0     0
```

```
Bbig=[B;0]
```

```
Bbig =  
    0  
    2  
    0
```

```
Cbig=[C E]
```

```
Cbig =  
    1     0    -1
```

```
sysbig=ss(Abig,Bbig,Cbig,0);  
rank(ctrb(Abig,Bbig))
```

```
ans = 2
```

```
rank(observ(Abig,Cbig))
```

```
ans = 3
```

Diseño del control por asignación de polos

Ganancia controlador:

```
t_est_deseado=0.5;  
polosdeseadosc=-4/t_est_deseado*[1 1.25]
```

```
polosdeseadosc =  
   -8   -10
```

```
K=place(A,B,polosdeseadosc)
```

```
K =  
   37.5000    8.5000
```

Ganancia ampliada, incorporando cancelación de perturbación estacionaria:

```
%K=[0 0];%bucle abierto
Kbig=[K -(Qu+K*Qx) ]
```

```
Kbig =
    37.5000    8.5000   -40.0000
```

Diseño del observador

```
polosdeseadosobs=-4/t_est_deseado*[1.5 1.75 2]
```

```
polosdeseadosobs =
   -12   -14   -16
```

```
Lbig=place(Abig',Cbig',polosdeseadosobs)';
```

formamos el controlador por realimentación de la salida:

```
controlador=reg(sysbig,Kbig,Lbig);
zpk(controlador)
```

```
ans =
```

```
From input "y1" to output "u1":
-7454.5 (s^2 + 5.924s + 14.42)
-----
      s (s^2 + 59s + 1356)
```

```
Input groups:
      Name      Channels
Measurement      1
```

```
Output groups:
      Name      Channels
Controls      1
```

Continuous-time zero/pole/gain model.

tiene un integrador, como se esperaba intuitivamente.

Simulación del resultado:

```
clp=feedback(sys*(-controlador),1);
zpk(clp)
```

```
ans =
```

```
From input "y1" to output:
 14909 (s^2 + 5.924s + 14.42)
-----
(s+16) (s+14) (s+12) (s+10) (s+8)
```

```
Input groups:
      Name      Channels
Measurement      1
```

Continuous-time zero/pole/gain model.

```
eig(clp)'
```

```
ans =  
    -16.0000   -14.0000   -12.0000   -10.0000    -8.0000
```

Ante un escalón en esa perturbación de entrada, la respuesta es:

```
dcgain(clp)
```

```
ans = 1.0000
```

```
step(clp), grid on
```

