

Cancelaciones en un controlador IMC (discreto), relación con controlabilidad/observabilidad de un sistema en bucle cerrado

© 2021, Antonio Sala Piqueras, Universitat Politècnica de València. Todos los derechos reservados.

Presentación en vídeo: <http://personales.upv.es/asala/YTV/cancobsctr.html>

Este código funcionó correctamente con Matlab R2021b

Objetivo: Las cancelaciones polo-cero entre distintos elementos (en este ejemplo, entre plantas y controladores) hacen cosas no observables o no controlables desde determinadas entradas y salidas... pero desde otras quizás no contempladas (o condiciones iniciales no nulas), sí pueden verse sus efectos.

Table of Contents

Modelo.....	1
Diseño (no recomendable) de control IMC discreto.....	2
Prestaciones en bucle cerrado.....	2
Controlabilidad y observabilidad del bucle cerrado.....	4
Forma canónica modal.....	6
Conclusión.....	9

Modelo

```
Ac=[0 1 ; -3 -0.2]; Bc=[0;3]; C=[1 0];  
sysc=ss(Ac,Bc,C,0);sysc.InputName='ureal';sysc.OutputName='y';  
tf(sysc) %vemos la TF, pero operaremos en repr. interna.
```

ans =

From input "ureal" to output "y":

3

s^2 + 0.2 s + 3

Continuous-time transfer function.

Vamos a discretizarlo:

```
Ts=0.2;  
sysd=c2d(sysc,Ts,'zoh');  
zpk(sysd)
```

ans =

From input "ureal" to output "y":

0.05862 (z+0.9867)

(z^2 - 1.844z + 0.9608)

Sample time: 0.2 seconds

Discrete-time zero/pole/gain model.

Diseño (no recomendable) de control IMC discreto

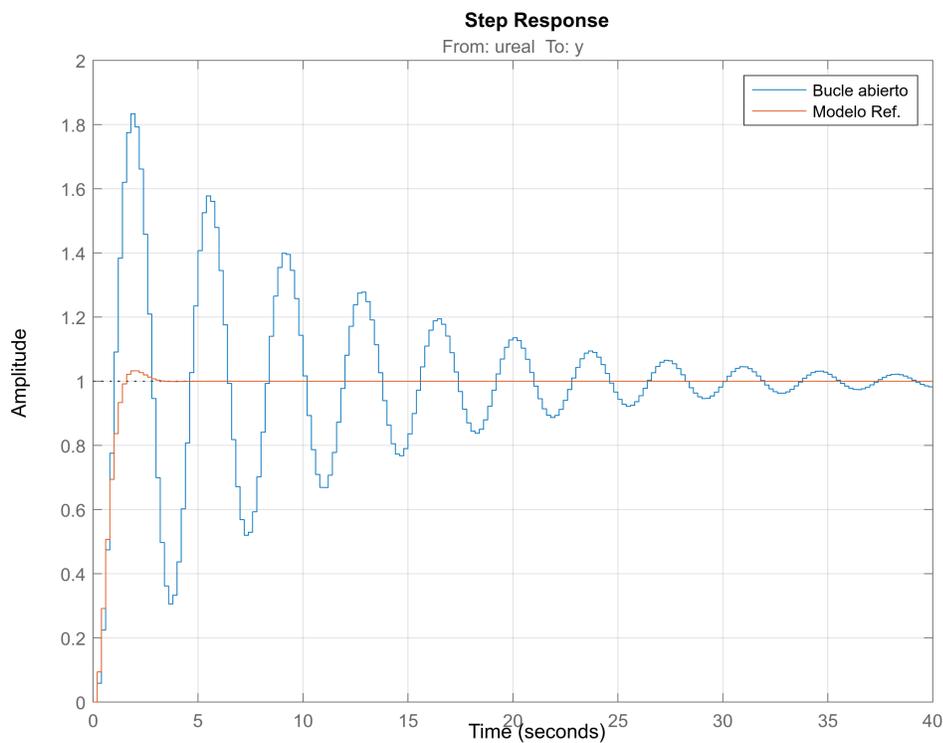
```
s=tf('s');
ModeloRefContinuo=6/(s^2+3.6*s+6);
ModeloRef=c2d(ModeloRefContinuo,Ts,'zoh');
zpk(ModeloRef)
```

```
ans =
```

```
0.093915 (z+0.7858)
-----
(z^2 - 1.319z + 0.4868)
```

```
Sample time: 0.2 seconds
Discrete-time zero/pole/gain model.
```

```
step(sysd,ModeloRef,40), grid on, legend("Bucle abierto", "Modelo Ref.")
```



```
Q=minreal(ModeloRef/sysd); %parámetro de Youla del control por modelo interno
```

```
2 states removed.
```

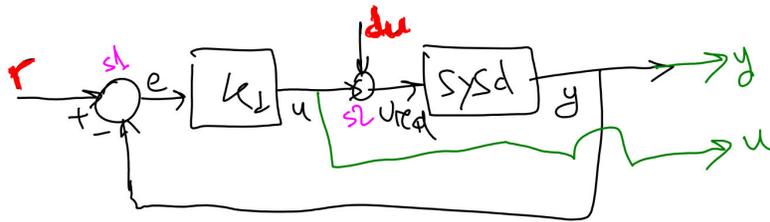
El regulador IMC será:

```
K_IMC=minreal(feedback(Q,-sysd)); size(K_IMC)
```

```
2 states removed.
```

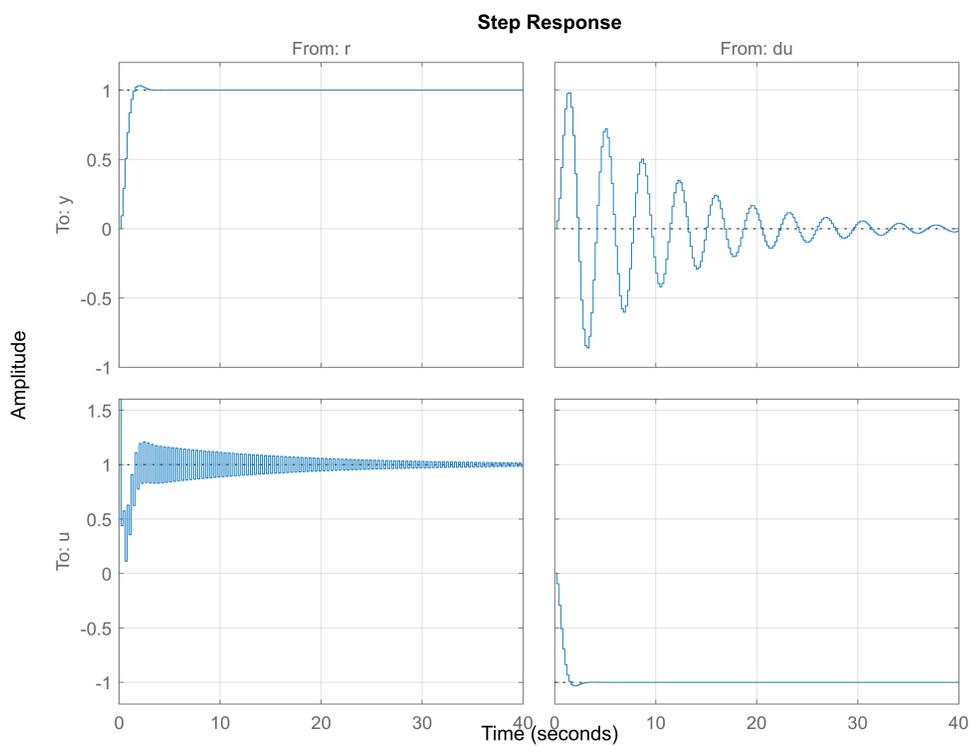
```
State-space model with 1 outputs, 1 inputs, and 3 states.
```

Prestaciones en bucle cerrado

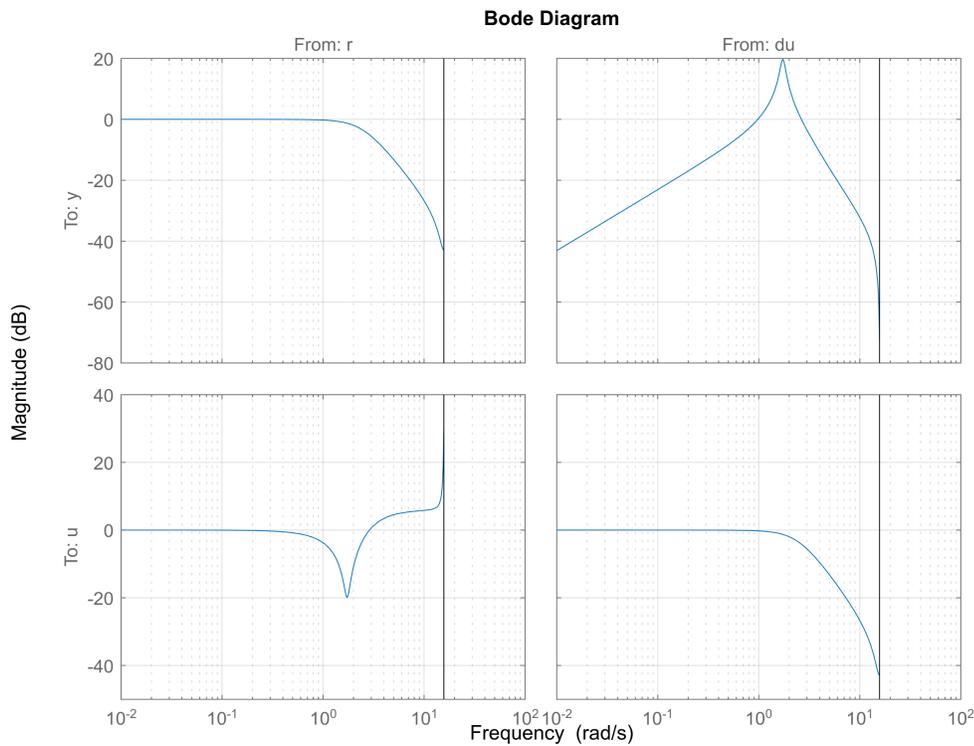


Cerramos bucle con "connect", ante perturbación de entrada.

```
K_IMC.InputName='e';
K_IMC.OutputName='u';
s1=sumblk('e=r-y');s2=sumblk('ureal=u+du');
BucleCerrado=connect(s1,s2,K_IMC,sysd,{'r','du'},{'y','u'});
step(BucleCerrado,40), grid on
```



```
bodemag(BucleCerrado), grid on
```



No hay buena respuesta por la cancelación planta/controlador:

```
zpk(sysd)
```

```
ans =
```

```
From input "ureal" to output "y":
  0.05862 (z+0.9867)
-----
(z^2 - 1.844z + 0.9608)
```

```
Sample time: 0.2 seconds
Discrete-time zero/pole/gain model.
```

```
zpk(K_IMC)
```

```
ans =
```

```
From input "e" to output "u":
  1.6021 (z+0.7858) (z^2 - 1.844z + 0.9608)
-----
(z+0.9867) (z-1) (z-0.413)
```

```
Sample time: 0.2 seconds
Discrete-time zero/pole/gain model.
```

Controlabilidad y observabilidad del bucle cerrado

¿Se podía detectar estas cosas raras que ocurren en el step/bodemag basándose en controlabilidad/observabilidad?

```
size(BucleCerrado)
```

```
State-space model with 2 outputs, 2 inputs, and 5 states.
```

```
hsvd(BucleCerrado) ' %todo es "controlable" y "observable"
```

```
ans = 1x5
    16.5671    5.3074    4.9136    1.0647    1.0295
```

```
svd(ctrb(BucleCerrado.A,BucleCerrado.B)) ' %todo es "controlable"
```

```
ans = 1x5
    18.2982    1.8117    0.7756    0.6924    0.3097
```

```
rank(ctrb(BucleCerrado.A,BucleCerrado.B)) ' %todo es "controlable"
```

```
ans = 5
```

```
svd(observ(BucleCerrado.A,BucleCerrado.C)) ' %todo es "observable"
```

```
ans = 1x5
    2.4441    1.0511    1.0113    0.3258    0.0977
```

```
rank(observ(BucleCerrado.A,BucleCerrado.C)) ' %todo es "observable"
```

```
ans = 5
```

El sistema sea "controlable" y "observable" (esto es, todo lo que pasa por dentro lo mueve "ro d_u ", y se ve en "yo u ")... o sea, que lo que hemos visto en step/bodemag es "todo lo que hay que ver": si algo raro pasa internamente en el sistema, en alguna de las 4 gráficas se ha de ver: no necesitamos un diagrama de bloques más complicado con otras entradas o salidas para estudiar este bucle.

Pero, podemos sacar más conclusiones usando esto de controlabilidad y observabilidad, para entender en más detalle lo que está pasando.

```
svd(ctrb(BucleCerrado.A,BucleCerrado.B(:,1))) ' %2 estados no son "controlables" desde "du"
```

```
ans = 1x5
    18.2606    1.6432    0.6953    0.0000    0.0000
```

```
svd(ctrb(BucleCerrado.A,BucleCerrado.B(:,2))) ' %1 estado no es "controlable" desde "du"
```

```
ans = 1x5
    1.3917    0.8307    0.1404    0.0366    0.0000
```

```
svd(observ(BucleCerrado.A,BucleCerrado.C(1,:))) ' %1 estado no es "observable" desde "y"
```

```
ans = 1x5
    1.4575    0.9042    0.1298    0.0163    0.0000
```

```
svd(observ(BucleCerrado.A,BucleCerrado.C(2,:))) ' %2 estados no son "observables" desde "y"
```

```
ans = 1x5
    2.0728    0.9746    0.1091    0.0000    0.0000
```

Esto indica que si sólo nos fijamos en una de las entradas o en una de las salidas, nos dejaríamos algo "en el tintero" sobre la dinámica interna de este bucle.

Para ver qué estados (o combinaciones de ellos) no son controlables u observables desde qué entradas o salidas, pasemos a forma modal.

Forma canónica modal

```
eBC=eig(BucleCerrado)'
```

```
eBC = 1x5 complex
    -0.9867 + 0.0000i    0.9222 - 0.3323i    0.9222 + 0.3323i    0.6595 - 0.2276i ...
```

```
abs(eBC)
```

```
ans = 1x5
    0.9867    0.9802    0.9802    0.6977    0.6977
```

Pasando a forma canónica modal (diagonalizando; bueno, los pares de polos complejos de 2 en 2 con `[re im;-im re]`):

```
[Canon_sys, Ti]=canon(BucleCerrado,'modal');
```

A priori, los estados pueden ser mezcla de estados de planta y controlador... no supone inconveniente, porque las entradas y salidas no cambian:

```
Ti
```

```
Ti = 5x5
    -0.6177    -0.3920    -0.4214    -1.1494    0.1169
     0.6168     0.9943     0.7118     1.8156    -0.7373
    -0.0112     0.9765    -0.4531     1.3843     1.0927
    -0.3371     0.3634    -0.8146    -1.4371     0.6959
    -0.1491    -0.6289    -0.0081    -0.2298    -0.4691
```

Esta es la forma canónica final:

```
Canon_sys
```

```
Canon_sys =
```

```
A =
```

	x1	x2	x3	x4	x5
x1	-0.9867	0	0	0	0
x2	0	0.9222	0.3323	0	0
x3	0	-0.3323	0.9222	0	0
x4	0	0	0	0.6595	0.2276
x5	0	0	0	-0.2276	0.6595

```
B =
```

	r	du
x1	-2.284	-6.496e-16
x2	-5.58e-15	-0.3186

```

x3  3.859e-16    0.711
x4  -0.7539     0.3169
x5  -0.1214    -0.2839

```

```

C =
      x1      x2      x3      x4      x5
y  2.398e-17  0.05694  0.5175  -0.2456  0.7516
u   0.1916  -2.304e-15  1.11e-15   0.771   1.191

```

```

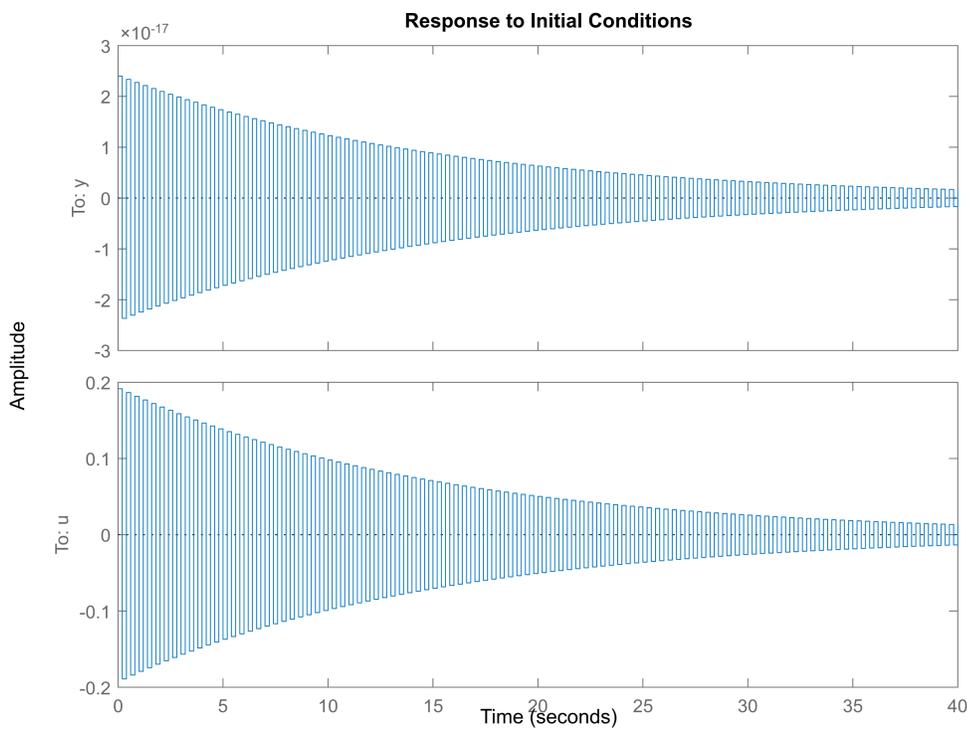
D =
      r      du
y      0      0
u  1.602    0

```

Sample time: 0.2 seconds
Discrete-time state-space model.

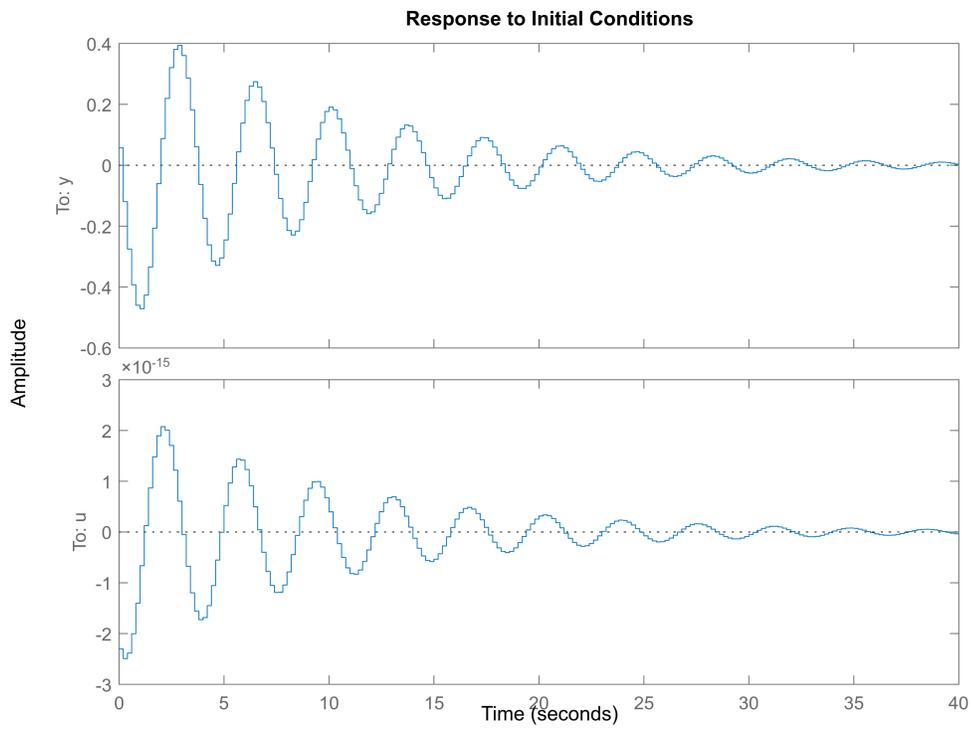
Las condiciones iniciales en cada uno de los nuevos estados dan lugar a los siguientes transitorios:

```
initial(Canon_sys, [1;0;0;0;0],40) %condiciones iniciales coincidentes con autovectores
```



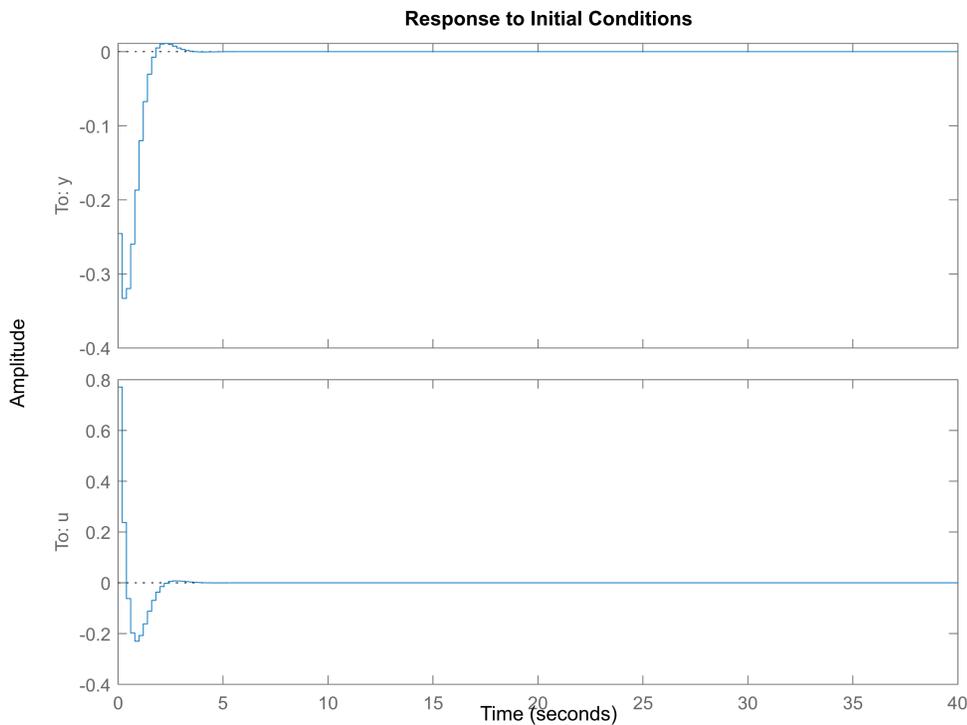
El polo se ve en "u" pero no en "y".

```
initial(Canon_sys, [0;1;0;0;0],40) %condiciones iniciales coincidentes con autovectores
```



El polo se ve en "y" pero no en "u".

```
initial(Canon_sys, [0;0;0;1;0], 40) %condiciones iniciales coincidentes con autovectores
```



Esta dinámica se ve tanto en "y" como en "u".

Conclusión

- Hay dos modos (polos) que tardan 40 segundos... y que no se hace control para amortiguarlos... Los componentes de condiciones iniciales en esas direcciones darán una dinámica mala, también si la excitan los ruidos de proceso (pert. entrada): al final de los transitorios sólo quedarán esos modos... la referencia no los excita, pero las condiciones iniciales fuera de equilibrio, posibles perturbaciones y errores de modelado que siempre habrá en la práctica, es posible que sí.
- Hay otro modo no observable en la salida, pero que da lugar a una acción de control de alta frecuencia. Tampoco es deseable.

Esto justifica las reglas usuales "de libro" sobre IMC o cualquier otro regulador basado en cancelación (p.ej. *mixed sensitivity*): (a) No uses IMC en procesos para los que desees cambiar "mucho" la dinámica de bucle abierto porque cambiará ante referencia, pero no ante "initial" o "perturb. entrada"... **no canceles polos cerca del límite de inestabilidad**; (b) **No canceles ceros cerca del límite de inestabilidad**, porque te destrozará la respuesta en acción de control.