

Desacoplamiento SVD, ejemplo Matlab

(c) Antonio Sala Piqueras, 2023. Universitat Politècnica de València. Todos los derechos reservados.

Presentación en vídeo del contenido en: <https://youtu.be/NKJybOfu7sk>

Este código ejecutó sin errores en Matlab R2022a

Modelado, análisis de controlabilidad entrada/salida

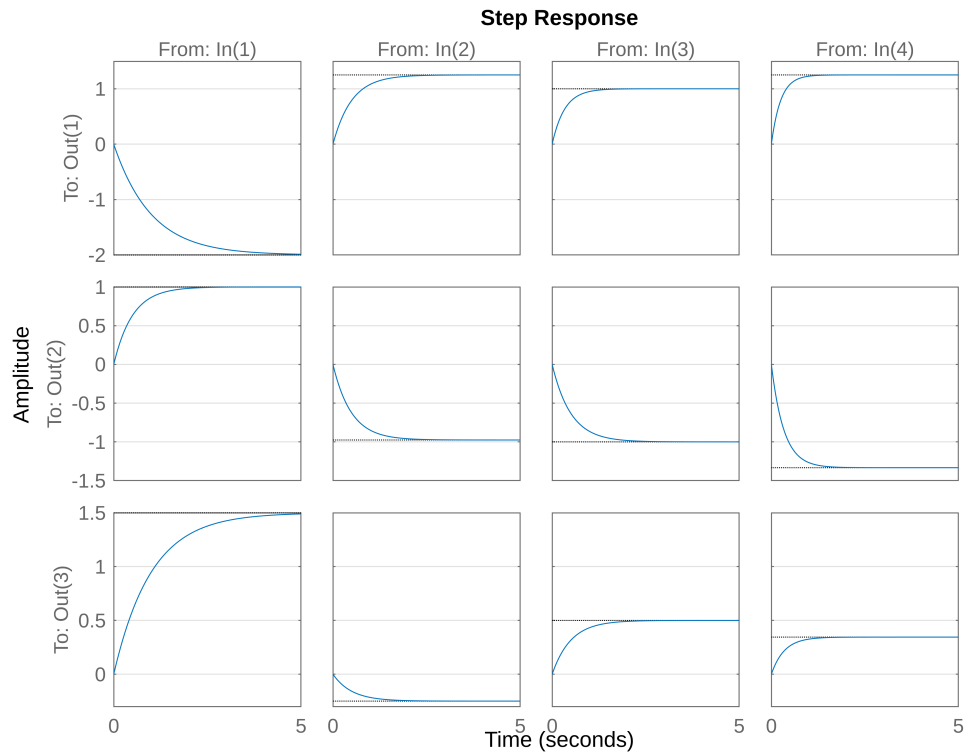
```
s=tf('s');  
G=[-2/(s+1) 2.5/(s+2) 3/(s+3) 5/(s+4); ...  
    2/(s+2) -2/(s+2.05) -2/(s+2) -4/(s+3); ...  
    1.5/(s+1) -0.5/(s+2) 1/(s+2) 1/(s+2.9)];  
%suponemos que ya está escalada, por simplicidad Ey=eye(3); Eu=eye(4)  
size(G)
```

Transfer function with 3 outputs and 4 inputs.

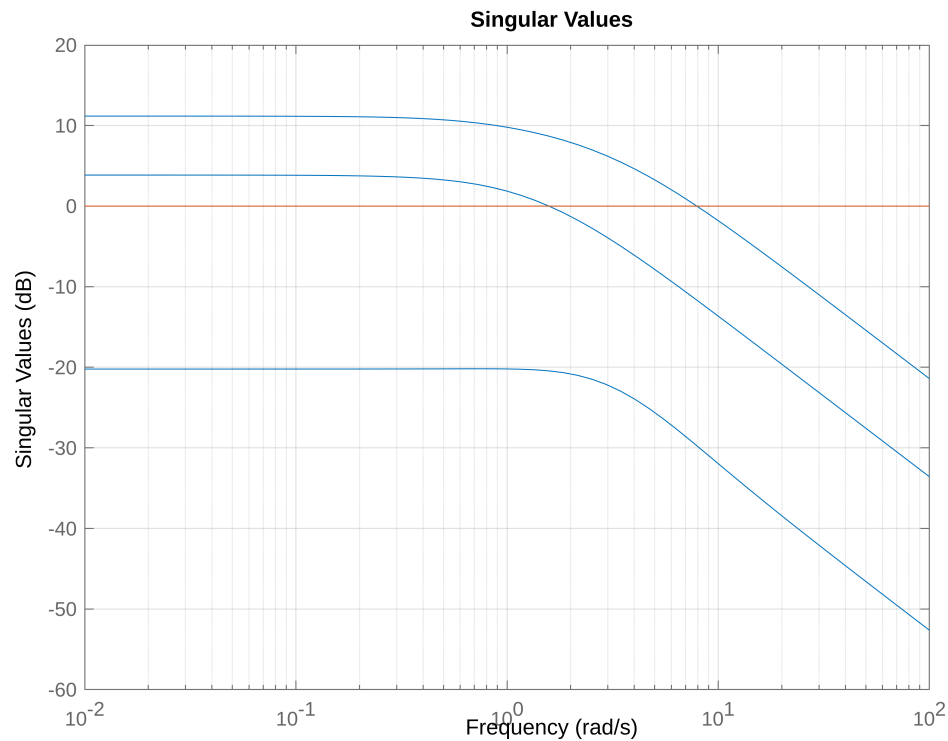
```
size(ss(G))
```

State-space model with 3 outputs, 4 inputs, and 12 states.

```
step(G,5), grid on
```



```
%análisis de controlabilidad entrada-salida:  
sigma(G, tf(1)), grid on
```



```
%pero NO vamos a hacer control basado en modelos, sino PID's con
%desacoplamiento, con lo que no reducimos orden y nos olvidamos de Gred.

%se pueden controlar DOS cosas, una hasta 7 rad/s, otra hasta 1.5 rad/s.
t_est_cosa1=pi/7
```

```
t_est_cosa1 = 0.4488
```

```
t_est_cosa2=pi/1.5
```

```
t_est_cosa2 = 2.0944
```

```
%a frec lenta, la ganancia es:
gananciaG=dcgain(G)
```

```
gananciaG = 3x4
   -2.0000    1.2500    1.0000    1.2500
    1.0000   -0.9756   -1.0000   -1.3333
    1.5000   -0.2500    0.5000    0.3448
```

```
% y el SVD, que da las dos direcciones (salidas) controlables en U (cols 1, 2)
% y los movimientos necesarios de actuadores en V (cols 1,2) es:
[U,S,V]=svd(gananciaG,'econ')
```

```
U = 3x3
    0.7873   -0.0366   -0.6155
   -0.5747   -0.4050   -0.7111
   -0.2233    0.9136   -0.3399
S = 3x3
```

```

    3.6190    0    0
    0    1.5586    0
    0    0    0.0975
V = 4x3
   -0.6864    0.6663    0.1030
    0.4423    0.0776    0.0959
    0.3455    0.5295   -0.7628
    0.4624    0.5193    0.6312

```

```

null(gananciaG)

```

```

ans = 4x1
   -0.2724
   -0.8883
    0.1360
    0.3437

```

```

%si controlamos tres cosas el condicionamiento es "fatal"
cond(gananciaG)

```

```

ans = 37.1212

```

Desacoplamiento SVD

```

nvariablescontroladas=2;
%si controlamos dos, es bueno
cond(S(1:nvariablescontroladas,1:nvariablescontroladas))

```

```

ans = 2.3220

```

```

%controlamos 2 cosas con 2 PID's y renunciamos a controlar la tercera.
Kp=1;
Ki=min(-eig(G))

```

```

Ki = 1

```

```

PIDs=eye(nvariablescontroladas)*(Kp*(1+Ki/s))

```

```

PIDs =

```

```

From input 1 to output...

```

```

    s + 1
1:  -----
    s

```

```

2:  0

```

```

From input 2 to output...

```

```

1:  0

    s + 1
2:  -----
    s

```

```

Continuous-time transfer function.

```

```

trozoU=U(:,1:nvariablescontroladas);
trozoV=V(:,1:nvariablescontroladas);
Ksvd=trozoV*PIDs*trozoU';

```

```
size(Ksvd)
```

Transfer function with 4 outputs and 3 inputs.

```
%cerremos el bucle
CL=minreal(feedback(minreal(G*Ksvd),eye(3)));
Acccontrol=minreal(feedback(Ksvd,G));
size(CL)
```

Transfer function with 3 outputs and 3 inputs.

```
% veamos como responde a una referencia, simulemos tres casos, dos casi alineados con
% las direcciones que sí se controlan, y uno que no
refe{1}=[1 -0.8 -0.4]';%fácil
refe{2}=[0 -.5 1]';%fácil
refe{3}=[0.9 1 .5]'; %difícil, no la controlamos

ref=refe{1}
```

```
ref = 3×1
    1.0000
   -0.8000
   -0.4000
```

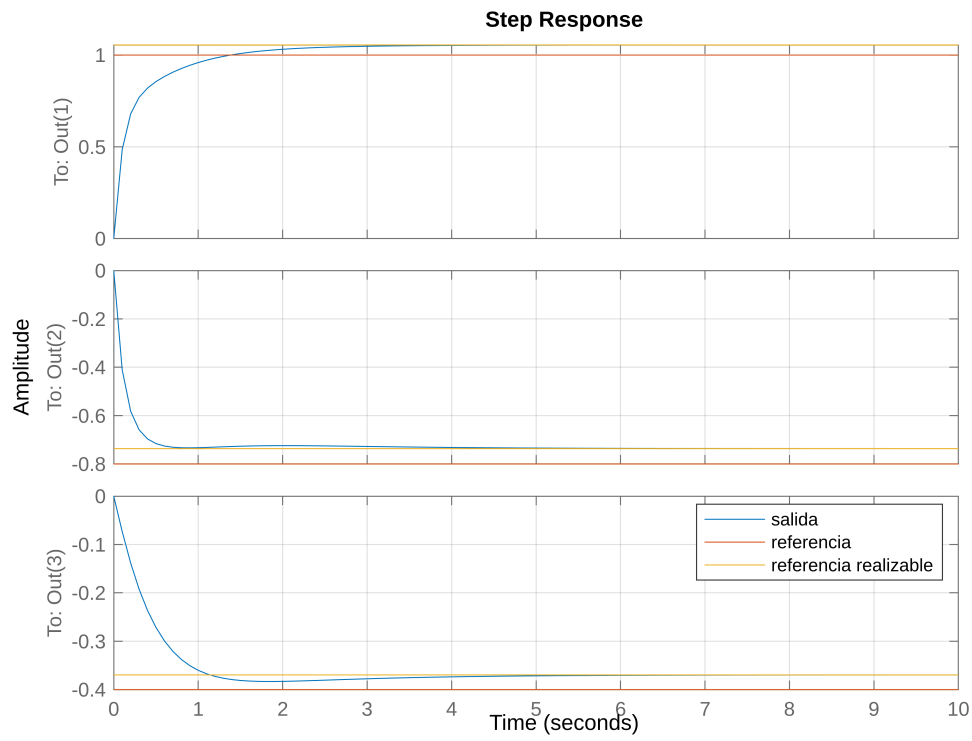
```
%en las nuevas coordenadas (componentes principales) la referencia vale casi todo en el
ref_nuevasvbles=U'*ref
```

```
ref_nuevasvbles = 3×1
    1.3364
   -0.0780
    0.0893
```

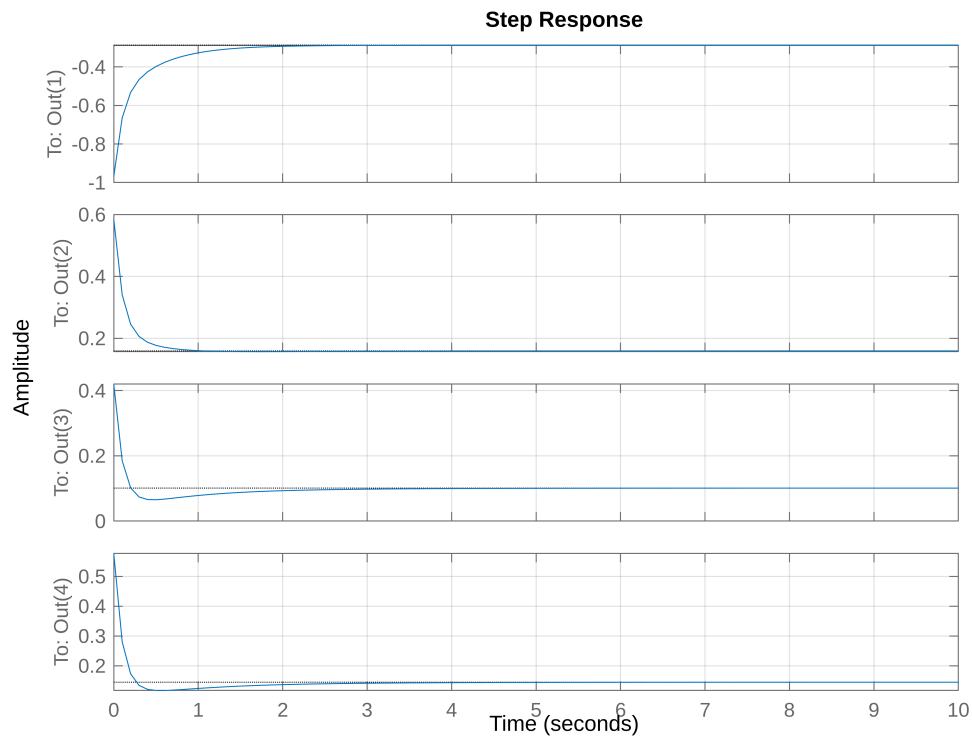
```
%solo los componentes en las dos direcciones a controlar serán objetivo del
%regulador.
refrealizable=trozoU*trozoU'*ref
```

```
refrealizable = 3×1
    1.0550
   -0.7365
   -0.3696
```

```
% la respuesta de los PID es:
step(CL*ref,tf(ref),tf(refrealizable),10), grid on
legend('salida','referencia','referencia realizable')
```



```
%y la acción de control
step(Acccontrol*ref,10), grid on
```



```
%segundo caso
```

```
ref=refe{2}
```

```
ref = 3×1
      0
 -0.5000
  1.0000
```

```
%en las nuevas coordenadas (componentes principales) la referencia vale:
```

```
ref_nuevasvbles=U'*ref
```

```
ref_nuevasvbles = 3×1
    0.0641
    1.1161
    0.0156
```

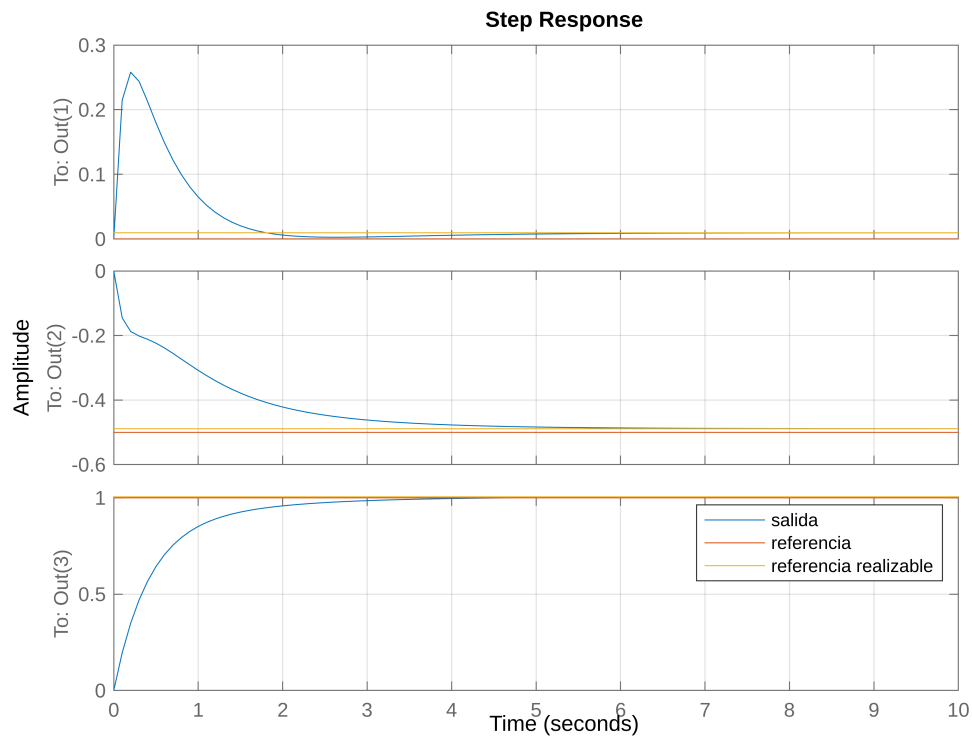
```
%solo los componentes en las dos direcciones a controlar serán objetivo del
%regulador. Escogiendo sólo esos dos y deshaciendo el cambio tenemos:
```

```
refrealizable=trozoU*trozoU'*ref
```

```
refrealizable = 3×1
    0.0096
   -0.4889
    1.0053
```

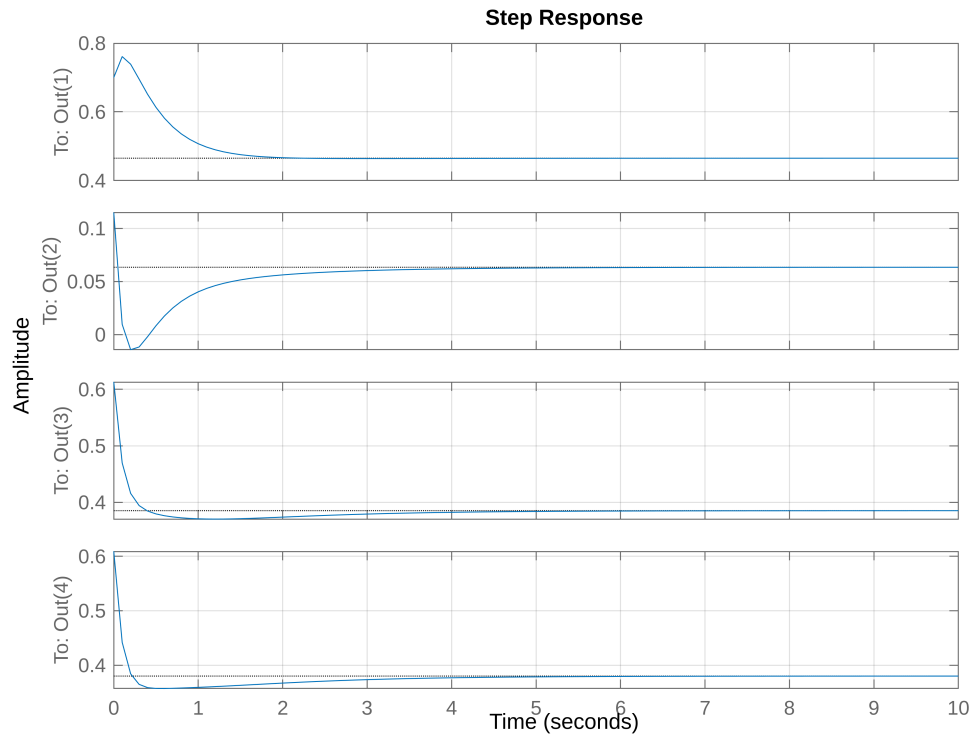
```
% la respuesta de los PID es:
```

```
step(CL*ref,tf(ref),tf(refrealizable),10), grid on
legend('salida','referencia','referencia realizable')
```



```
%y la acción de control
```

```
step(Acccontrol*ref,10), grid on
```



```
%seguir algo NO controlable???
```

```
ref=refe{3}
```

```
ref = 3×1  
    0.9000  
    1.0000  
    0.5000
```

```
%en las nuevas coordenadas (componentes principales) la referencia vale en el tercer co  
% porque no se puede por saturación y condicionamiento.
```

```
ref_nuevasvbles=U'*ref
```

```
ref_nuevasvbles = 3×1  
    0.0222  
    0.0188  
   -1.4350
```

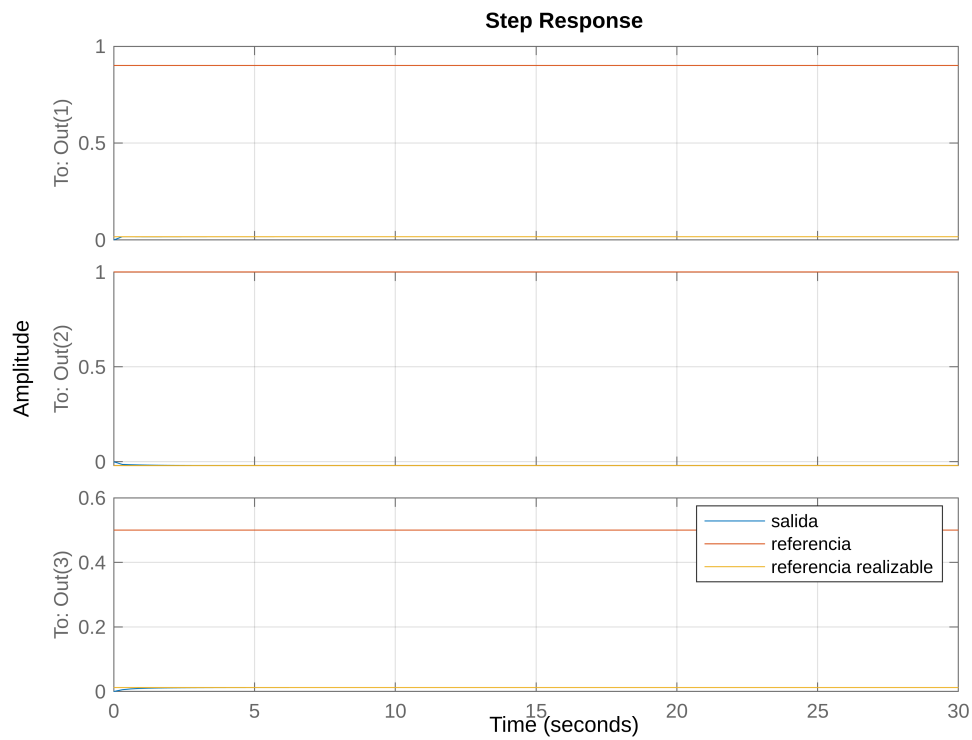
```
%solo los componentes en las dos direcciones a controlar serán objetivo del  
%regulador. Escogiendo sólo esos dos y deshaciendo el cambio tenemos:
```

```
refrealizable=trozoU*trozoU'*ref
```

```
refrealizable = 3×1  
    0.0168  
   -0.0204  
    0.0122
```

```
% la respuesta de los PIDs es:
```

```
step(CL*ref,tf(ref),tf(refrealizable),30), grid on
legend('salida','referencia','referencia realizable')
```



```
%en este tercer caso, intencionadamente, NO se sigue la referencia.
%y la acción de control es muy pequeña, porque aunque tratáramos de seguirla, saturaría
% con lo que no la seguimos.
step(Acccontrol*ref,30), grid on
```