Desacoplamiento SVD de un modelo de horno lineal: SIMULINK, diagrama de bloques, RESPUESTA TEMPORAL

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

Presentación en vídeo: http://personales.upv.es/asala/YT/V/forn3.html

Este código ejecutó correctamente en Matlab **R2021a**

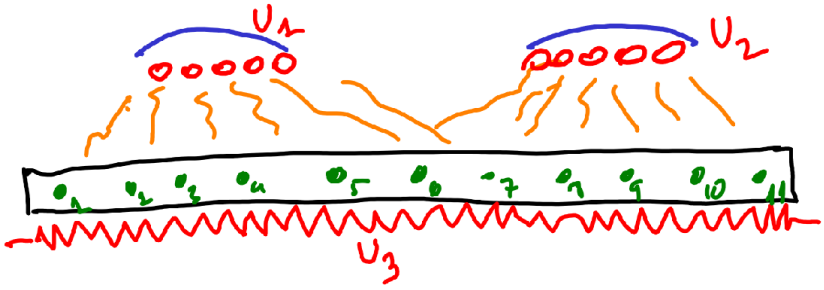
Objetivos: Diseñar controladores PID para combinaciones de salidas (desac. SVD) en un modelo de horno y simular su comportamiento en tiempo y frecuencia (en las coordenadas dadas por la transformación a componentes principales de la ganancia). Implementar en Simulink el controlador resultante.

Tabla de Contenidos

Modelo 1
Análisis de controlabilidad Entrada-Salida 2
Cambio de variable, desacoplamiento SVD 5
  Sintonía de PIDs 7
  Propiedades en bucle cerrado (tiempo, frecuencia) de los componentes principales 9
Conclusiones 11

Modelo

Supongamos un modelo de sistema térmico donde TRES elementos calefactores calientan determinada pieza y se desea controlar la temperatura en 11 puntos (informalmente, dado que matemáticamente no es posible con 3 actuadores):




```
MCe=10*diag(ones(1,11));
Acond=diag(-[1 2*ones(1,9) 1])+diag(ones(10,1),1)+diag(ones(10,1),-1);
Kconduc=0.5;Kamb=0.49;
A=MCe\(Acond*Kconduc-Kamb*diag([1.5 1 1 1 1 1 1 1 1 1 1.5]));
B=MCe\(20*[0.85  1.05 1.1  1 .8 .55 .45 .35 .25 .2 .2;
    .05 .1 .15 .3 .4 .5 .8 1.1 1.1 1. 0.95;
    .6*ones(1,11)]');
forn=ss(A,B,eye(11),0);
```

Análisis de controlabilidad Entrada-Salida

El escalado lo haremos con:

-- límite de cada actuador

```
Eu=diag([7; 5.5; 3.5]);
```

-- incremento (idealmente) deseado de cada termopar

```
Ey=diag([100 120 140 160 160 160 160 160 140 120 100]);
```

Con lo que el sistema dinámico escalado será:

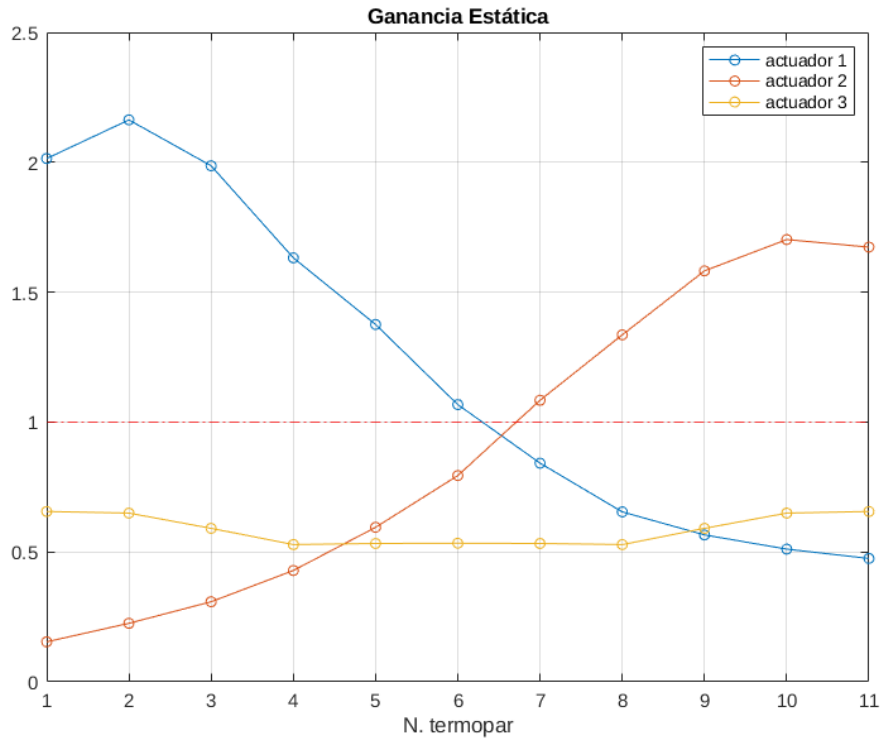
```
forn_esc=Ey\forn*Eu; %adimensional todo en [-1,+1] como referencia.
```

Resulta en esta matriz de ganancia estática escalada:

```
Gan_esc=dcgain(forn_esc)
```

```
Gan_esc = 11×3
    2.0145    0.1540    0.6556
    2.1631    0.2252    0.6495
    1.9864    0.3082    0.5908
    1.6321    0.4285    0.5283
    1.3756    0.5948    0.5325
    1.0671    0.7940    0.5336
    0.8418    1.0838    0.5325
    0.6541    1.3358    0.5283
    0.5656    1.5820    0.5908
    0.5108    1.7025    0.6495
     ⋮
```

```
plot(Gan_esc,Marker='o'), grid on, yline(1,'-.r'), title("Ganancia Estática")
legend("actuador 1","actuador 2","actuador 3"), xlabel("N. termopar")
```

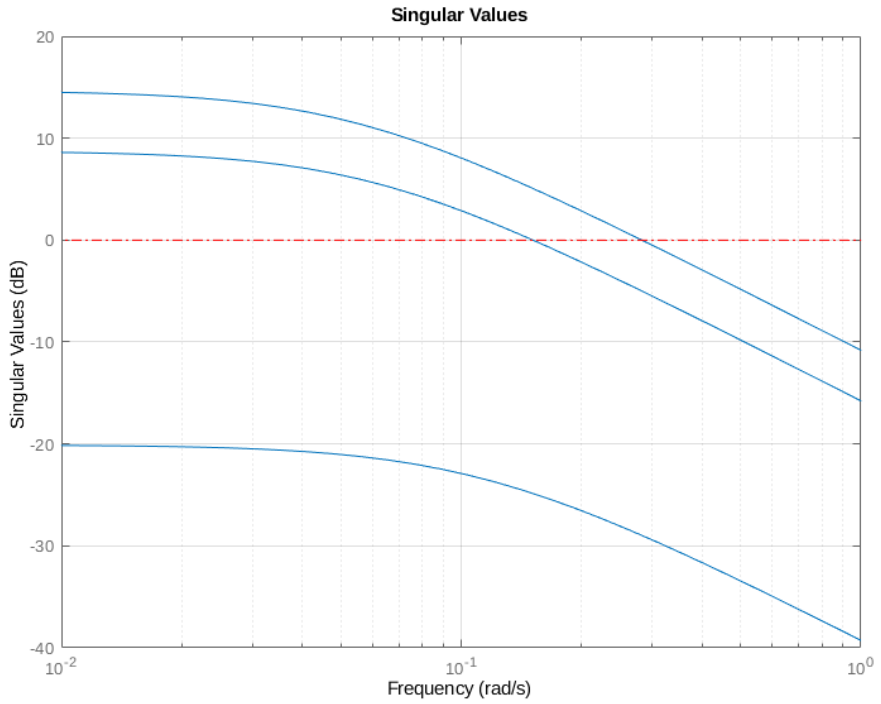



```
[Nout,Nin]=size(Gan_esc)
```

```
Nout = 11
Nin = 3
```

Los valores singulares en función de la frecuencia son:

```
sigma(forn_esc,tf(1),'-.r'), grid on
```



Los cruces con 0dB de cada maniobra son a 0.16 y a 0.28 rad/s. Su semiperíodo es una aproximación (quizás optimista) al tiempo de establecimiento alcanzable:

```
pi/0.15
```

```
ans = 20.9440
```

```
pi/0.29
```

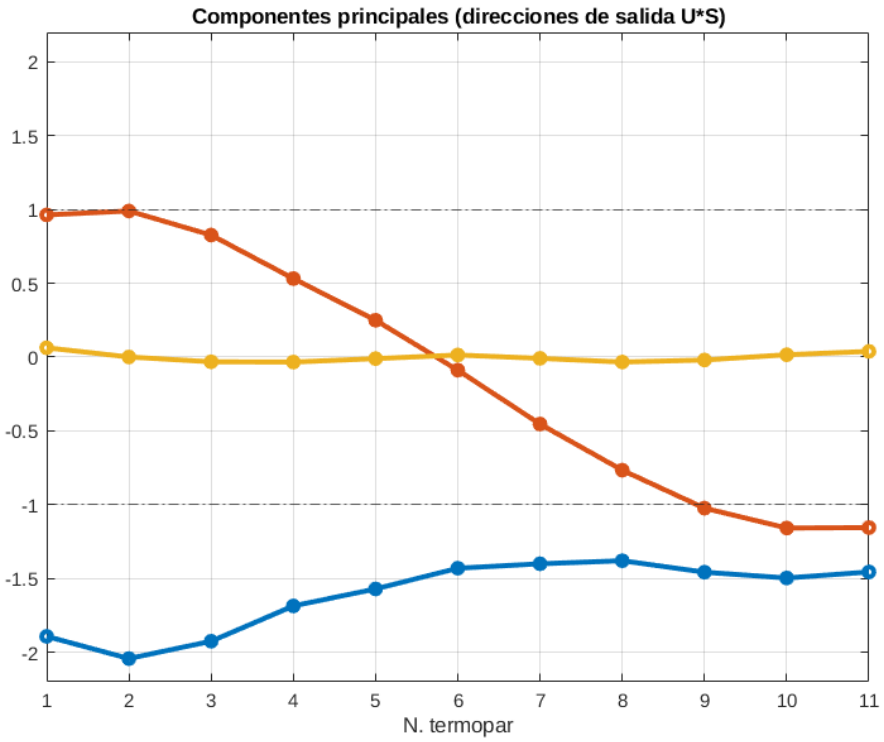
```
ans = 10.8331
```

Análisis de la ganancia: en concreto a frecuencia cero, las direcciones de entrada y salida son:

```
[U,S,V]=svd(Gan_esc,'econ');
S %ganancias principales (valores singulares)
```

```
S = 3×3
    5.3999         0         0
         0    2.7314         0
         0         0    0.0988
```

```
plot(U*S,LineWidth=3, Marker='o'), grid on, title("Componentes principales (direcciones
 yline(1,'-.'), yline(-1,'-.'), ylim([-2.2 2.2])
xlabel('N. termopar')
```



Las direcciones de entrada son:

```
V
```

```
V = 3×3
   -0.7837    0.5683   -0.2507
   -0.5071   -0.8185   -0.2700
   -0.3586   -0.0845    0.9296
```

Cambio de variable, desacoplamiento SVD

Desacoplamiento SVD se basa en que $G_{esc}(0) = USV^T$ implica que $S = U^T \cdot G_{esc}(0) \cdot V$ es diagonal (a frecuencia baja)... Entonces si $u = V\widetilde{u}$, $\widetilde{y} = U^T y$, tenemos

$$\widetilde{y} = U^T \cdot G \cdot u = \underbrace{U^T G V}_{\widetilde{G}} \cdot \widetilde{u}$$

con $\widetilde{G}$ diagonal a baja frecuencia.

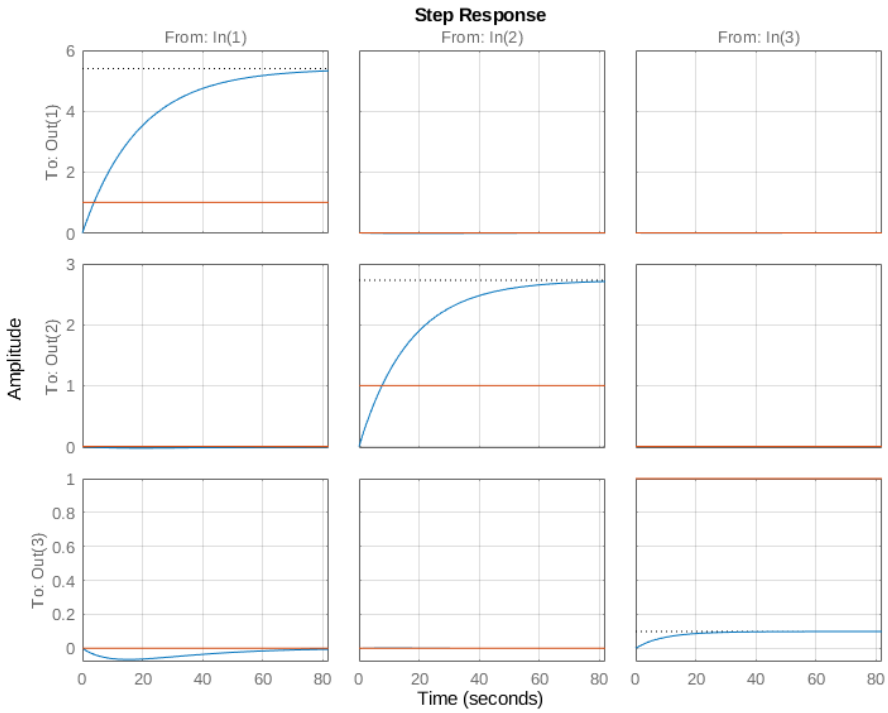
```
tildeGdinamico=U'*forn_esc*V;
tildeG=dcgain(tildeGdinamico) %coincide con S
```

```
tildeG = 3×3
    5.3999    0.0000    0.0000
```

```
   -0.0000    2.7314   -0.0000
    0.0000    0.0000    0.0988
```

Si vemos el transitorio en tiempo:

```
step(tildeGdinamico,tf(eye(3)),82), grid on
```



En efecto, está bastante desacoplado. Descartaremos controlar el tercer componente singular, porque su "ganancia" (valor singular) es muy pequeña. El condicionamiento del subsistema que vamos a controlar es bueno:

```
S(1,1)/S(2,2)
```

```
ans = 1.9770
```

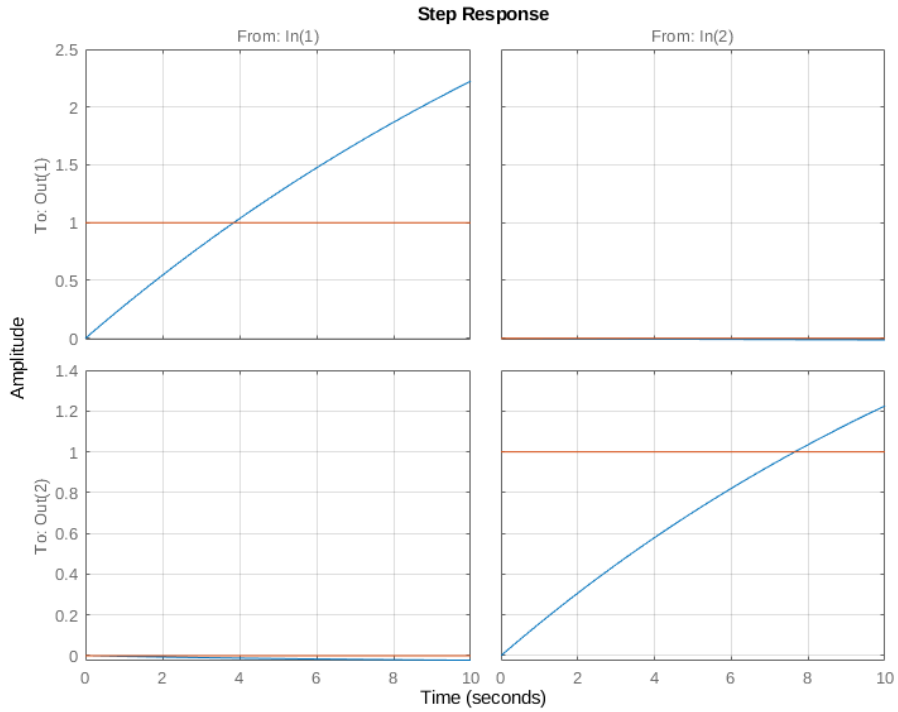
El control con "desacoplamiento SVD" sería un K_{desac} diagonal (comenzar con un PI, por ejemplo, dado que el desacoplamiento se origina por el SVD del proceso a frecuencias bajas --ganancia estática--), que en vez de controlar a $\widetilde{G}$, implementaríamos con $K = V_{(:,1:2)} * K_{desac} * U^T_{(:,1:2)}$.

Controlaríamos, por tanto, la proyección de los errores sobre las 2 primeras columnas de U(la otra tiene muy poca ganancia, y el bucle tendría muy mal condicionamiento). La acción integral haría cero esos dos componentes (en régimen permanente).

Si vemos las fases iniciales del transitorio:

```
tildeGdinamico=tildeGdinamico(1:2,1:2);
```

```
step(tildeGdinamico,tf(eye(2)),10), grid on
```



La primera maniobra tarda 4 segundos en llegar a 1, la segunda 8 segundos... son estimados optimistas de t_{est}, porque necesitamos "frenar" y "medir filtrando ruidos" y no "saturar durante todo el transitorio", son más estimados de tiempo de "subida" quizás.

Sintonía de PIDs

Diseñemos con pidtune, primero poniendo la frecuencia de corte con 0dB de las maniobras, luego subiendo el ancho de banda objetivo hasta que se satura a 1 en el transitorio la acción de control:

```
K1=pidtune(tildeGdinamico(1,1),'PI',0.36) %0.36 para saturar 1, 0.29 el ancho de banda
```

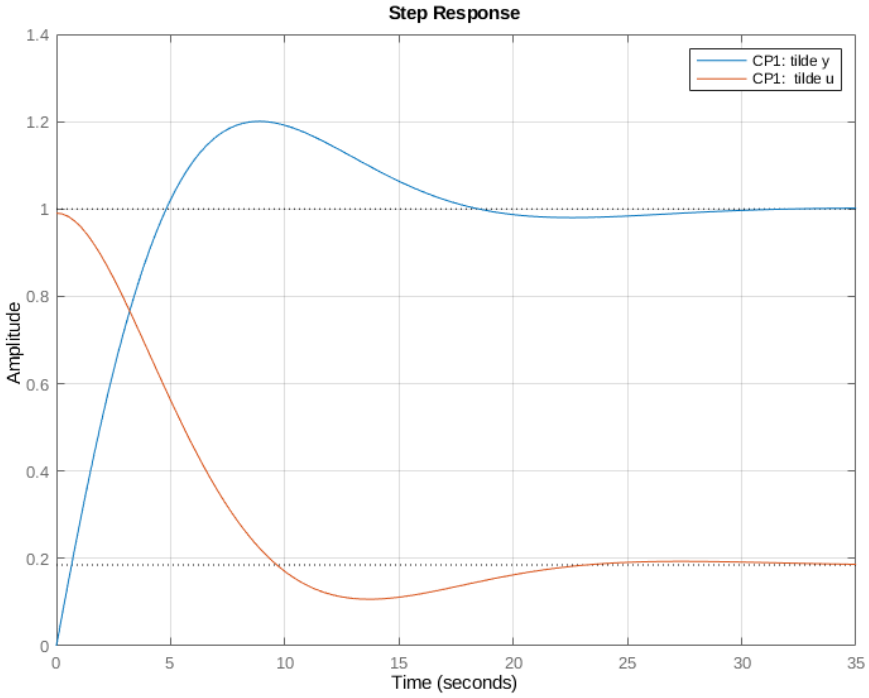
```
K1 =

             1
  Kp + Ki * ---
             s

  with Kp = 0.991, Ki = 0.284

Continuous-time PI controller in parallel form.
```

```
CL1=feedback(tildeGdinamico(1,1)*K1,1);
CL1u=feedback(K1,tildeGdinamico(1,1));
step(CL1,CL1u), grid on, legend('CP1: tilde y','CP1:  tilde u')
```




```
K2=pidtune(tildeGdinamico(2,2),'PI',0.21) %0.21 para saturar 1, 0.15 el ancho de banda
```

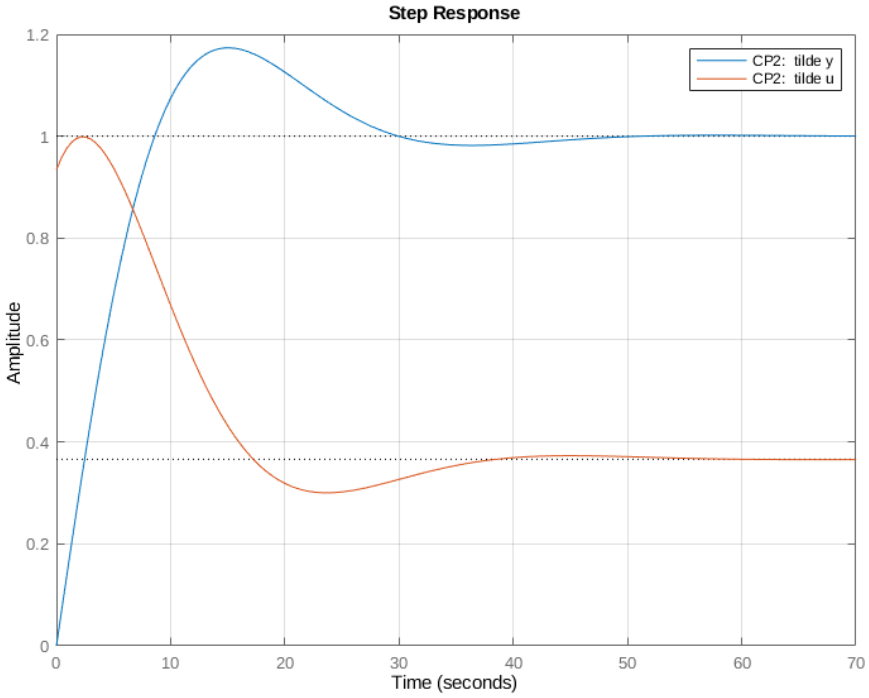
```
K2 =

             1
  Kp + Ki * ---
             s

  with Kp = 0.934, Ki = 0.202

Continuous-time PI controller in parallel form.
```

```
CL2=feedback(tildeGdinamico(2,2)*K2,1);
CL2u=feedback(K2,tildeGdinamico(2,2));
step(CL2,CL2u), grid on,, legend('CP2:  tilde y','CP2:  tilde u')
```



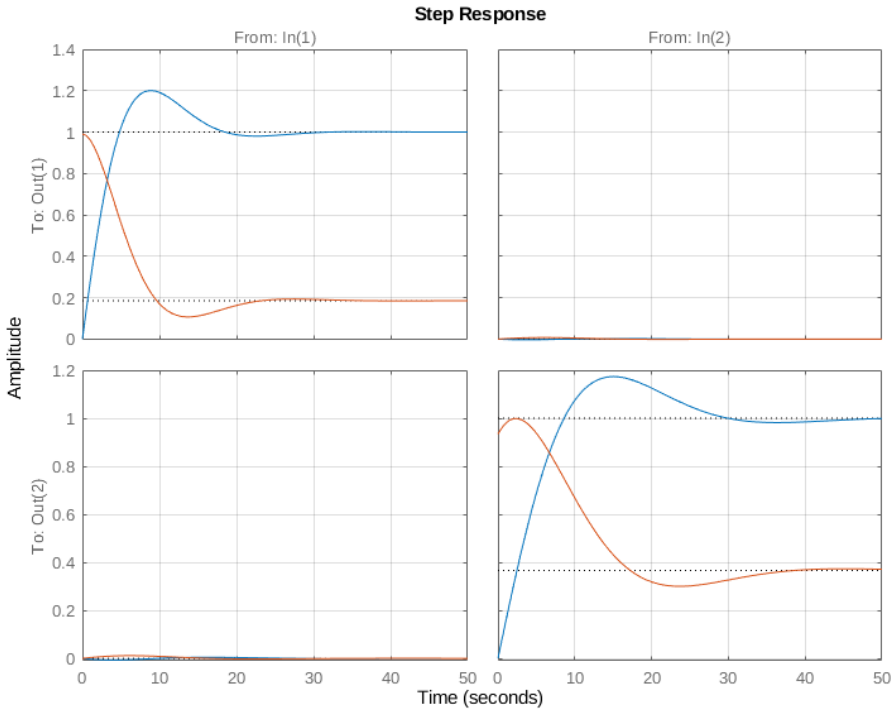
Con los anchos de banda objetivos del sigma del proceso, conseguimos tiempos de subida de 6 y 12 segundos, establecimiento de 20 y 35 segundos.

Saturando a 1 escalón conseguimos aproximadamente el "tiempo de subida" de 4 y 8 segundos... establecimiento, un poco mayor (15 y 30 segundos).

Propiedades en bucle cerrado (tiempo, frecuencia) de los componentes principales

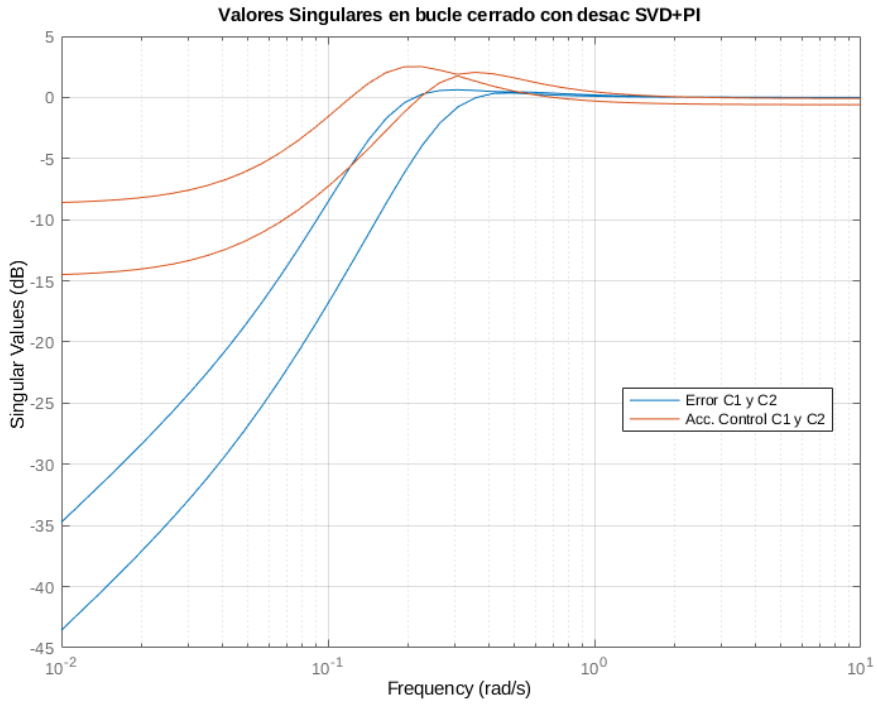
Simulemos todo en bucle cerrado, no sólo los elementos diagonales:

```
Ktot=minreal(ss(blkdiag(K1,K2)));%no hagamos operaciones multivariables en MdT, numéric
CLTot=feedback(tildeGdinamico*Ktot,eye(2));
CLTotu=feedback(Ktot,tildeGdinamico);
step(CLTot,CLTotu,50), grid on
```



En frecuencia (error, acc. control):

```
ref_a_error=eye(2)-CLTot;
sigma(ref_a_error,CLTotu), grid on, legend('Error C1 y C2','Acc. Control C1 y C2',Locat
title("Valores Singulares en bucle cerrado con desac SVD+PI")
```



Si ponemos ancho de banda hasta saturar a 1 ante escalón, la acción de control "supera 1" en amplitudes de referencia o perturbación a ciertas frecuencias... Podemos hacer el regulador más corto o darlo por bueno hasta comprobar el funcionamiento experimental (apenas son un par de decibelios y tenemos la incertidumbre esferas-cuadrados, y si saturamos "sólo esporádicamente" tampoco está tan mal)... aunque si se deseara que las maniobras fueran a "máximo 1" algún termopar en vez de "norma 1" podríamos saturar en exceso...

Importante: Aunque parezca 2x2 recordemos que estamos controlando 2 "combinaciones de sensores" con 2 "combinaciones de actuadores", con lo que realmente estamos gastando los **11 sensores y 3 actuadores** del proceso físico, en las direcciones de más ganancia.

IMPLEMENTACIÓN: el controlador en unidades físicas sería

```
Ksvd=Eu*V(:,1:2)*Ktot*U(:,1:2)'*inv(Ey);
```

La simulación del mismo ante referencias o perturbaciones, y la comprobación de si hay o no saturación, etc. no son, por brevedad, objetivo de este material.

Conclusiones

Hemos sintonizado **dos PIDs** (PI en concreto) para controlar un modelo de horno de **11 sensores** y **3 actuadores** mediante desacoplamiento SVD en las direcciones controlables de buena ganancia. Hemos evaluado la respuesta en tiempo y en frecuencia en los componentes principales.

Faltaría evaluar la respuesta en tiempo en unidades físicas, y el conservadurismo esferas/cuadrados de cara a la saturación ante las referencias que el usuario quisiese dar. También deberíamos evaluar si la saturación durante el transitorio causa problemas o no, porque el usuario realmente quiere ver a los actuadores saturar (y el predictivo así lo optimiza). Todo eso no es objetivo de este material.