

Control de un modelo de horno lineal: control centralizado $\mathcal{H}_2$ y $\mathcal{H}_\infty$, comparativa con PI por desacoplamiento SVD

Presentaciones en vídeo:

<http://personales.upv.es/asala/YT/V/forn6pg.html> ,

<http://personales.upv.es/asala/YT/V/forn7h2.html> ,

<http://personales.upv.es/asala/YT/V/forn8hi.html> .

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

Este código ejecutó correctamente en Matlab R2021a

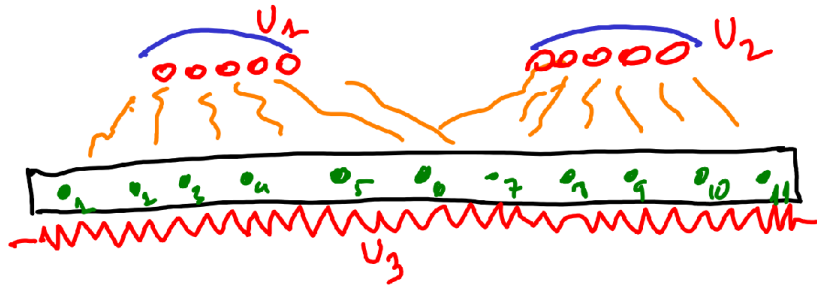
Objetivos: Diseñar controladores centralizados $\mathcal{H}_2$ y $\mathcal{H}_\infty$ en un modelo de horno y simular su comportamiento en tiempo y frecuencia. Comparar con PID por desacoplamiento SVD.

Tabla de Contenidos

Modelo.....	1
Análisis de controlabilidad Entrada-Salida.....	2
Cambio de variable, desacoplamiento SVD + PIDs.....	5
Sintonía de PIDs.....	7
Propiedades en bucle cerrado (tiempo, frecuencia) de los componentes principales.....	9
Control centralizado H2 o H-infinito.....	11
Planta Generalizada no ponderada.....	11
Diseño H-2 (LQR+Kalman).....	12
Diseño H-infinito (sensibilidad mixta).....	14
Reducción de orden de reguladores.....	16
Robustez de los diseños de control.....	21
Conclusiones.....	22

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

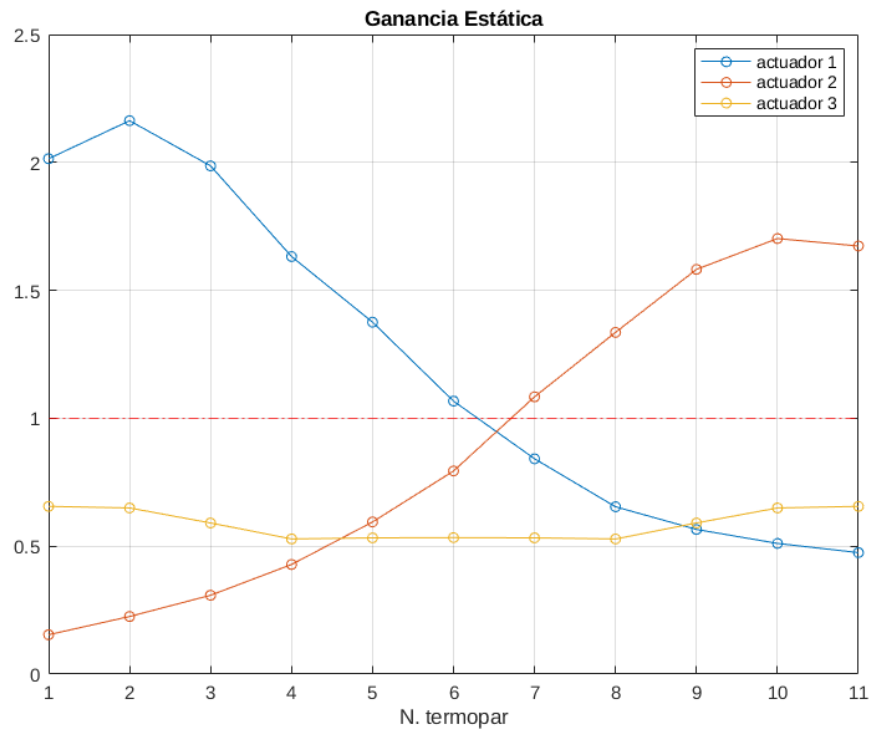
```
Ey=diag([100 120 140 160 160 160 160 160 140 120 100]);
```

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

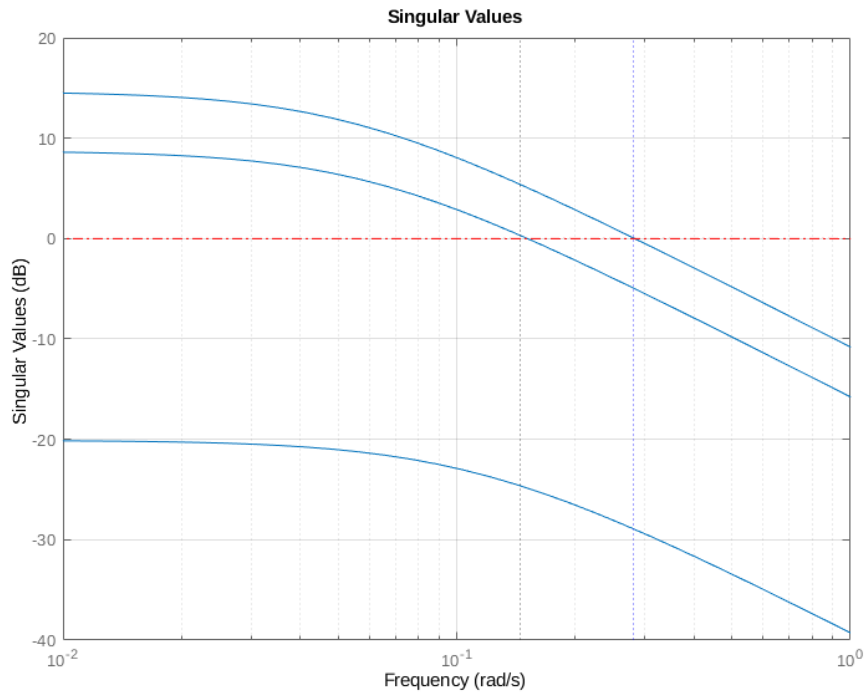


```
[Nout,Nin]=size(Gan_esc)
```

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

Los valores singulares en función de la frecuencia son:

```
sigma(forn_esc,tf(1),'-.r'), grid on,xline(0.28,':b'), xline(0.145,':')
```



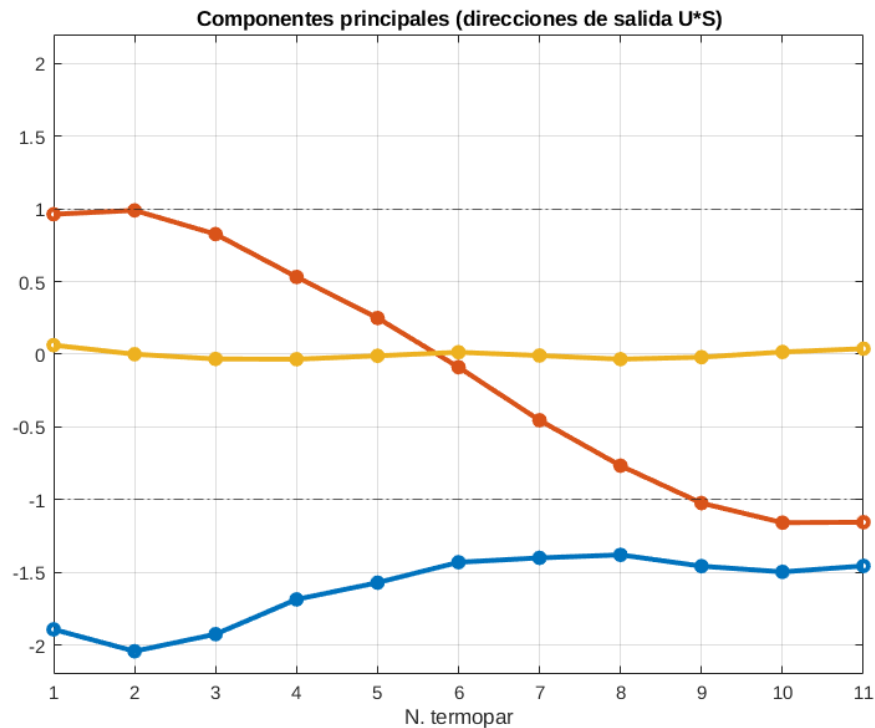
Los cruces con 0dB de cada maniobra son a 0.16 y a 0.28 rad/s.

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 = 3x3
    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

$V = 3 \times 3$

-0.7837	0.5683	-0.2507
-0.5071	-0.8185	-0.2700
-0.3586	-0.0845	0.9296

Cambio de variable, desacoplamiento SVD + PIDs

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 $u = V\tilde{u}$, $\tilde{y} = U^T y$, tenemos

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

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

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

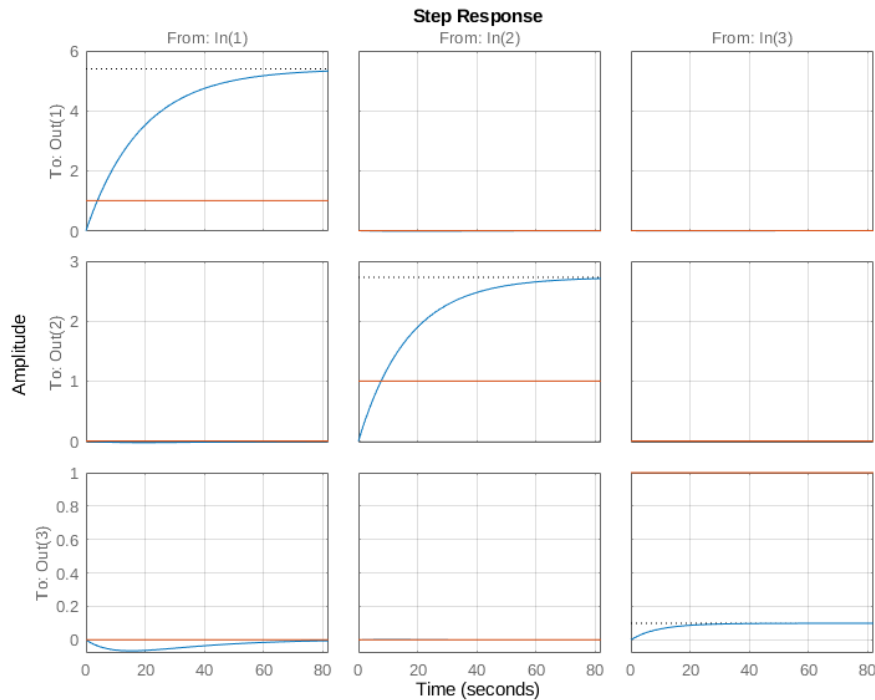
$\tilde{G} = 3 \times 3$

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

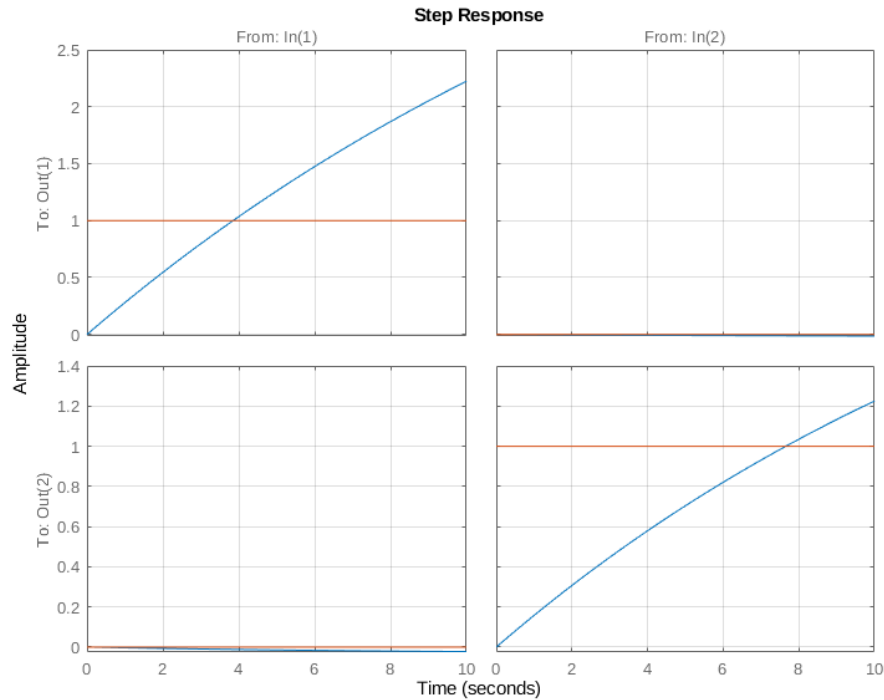
estática--), que en vez de controlar a $\tilde{G}$, implementaríamos con $K = V_{(:,1:2)} * K_{desac} * U_{(:,1:2)}^T$.

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 sature 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
```

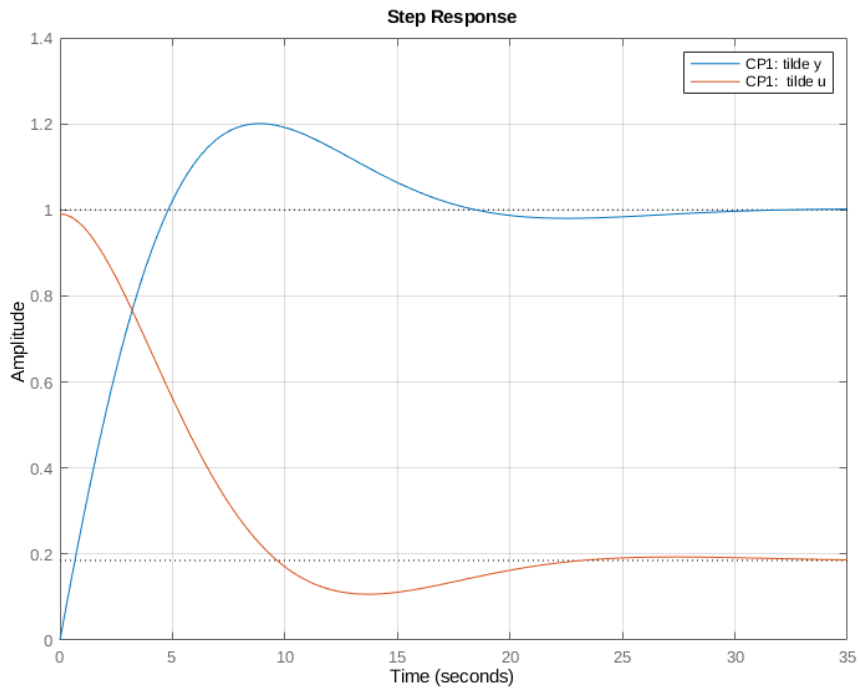
K1 =

$$K_p + K_i * \frac{1}{s}$$

with $K_p = 0.991$, $K_i = 0.284$

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
```

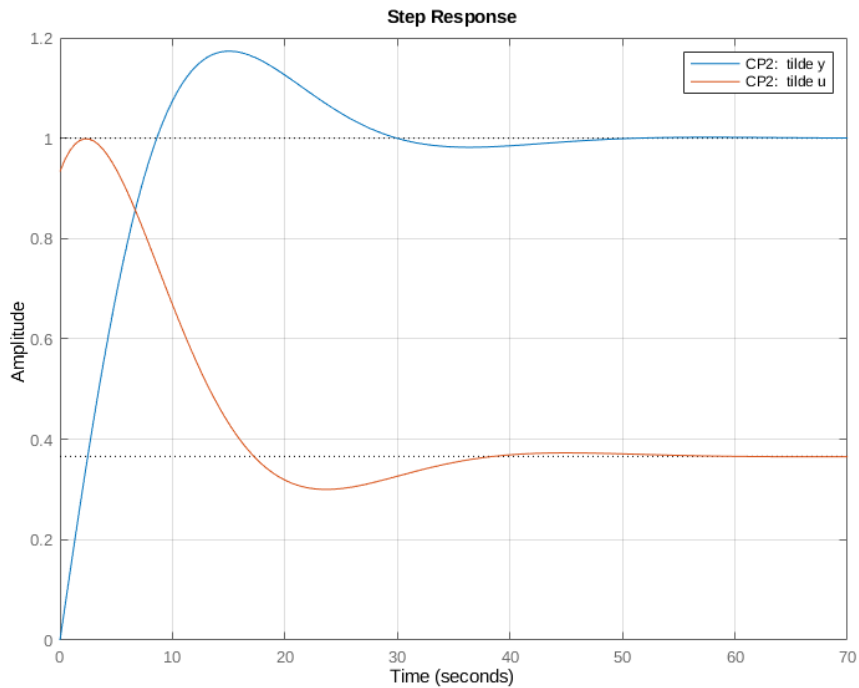
K2 =

$$K_p + K_i * \frac{1}{s}$$

with $K_p = 0.934$, $K_i = 0.202$

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')
```

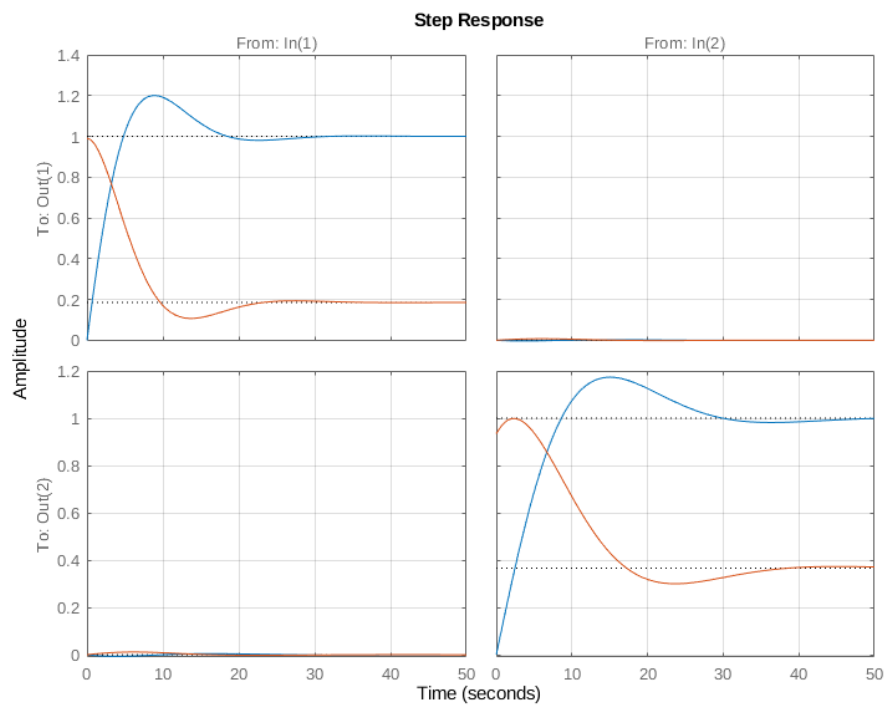
Con los anchos de banda objetivos del sigma del proceso, conseguimos tiempos de subida de 6 y 12 segundos, establecimiento de 20 y 35 segundos.

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

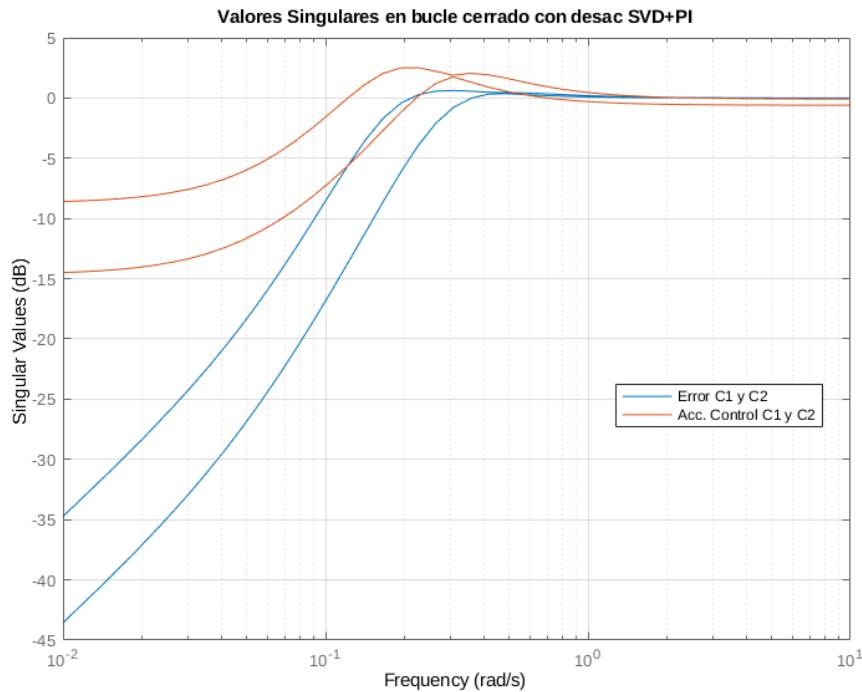
Simulemos todo en bucle cerrado, no sólo los elementos diagonales:

```
K2PItot=minreal(ss(blkdiag(K1,K2)));%no hagamos operaciones multivariabiles en MdT, numé
CLTot=feedback(tildeGdinamico*K2PItot,eye(2));
CLTotu=feedback(K2PItot,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',Location='best');
title("Valores Singulares en bucle cerrado con desac SVD+PI")
```

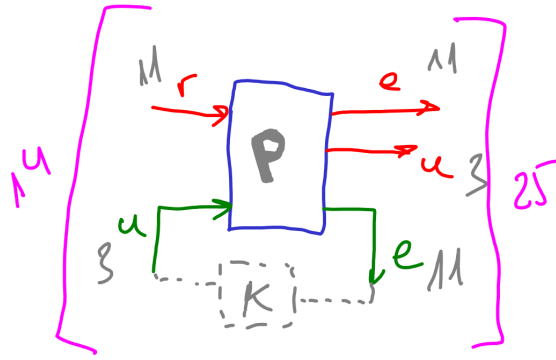


Si ponemos ancho de banda hasta saturar a 1 ante escalón, la acción de control "supera 1" en amplitudes de referencia o perturbación a ciertas frecuencias... Podemos hacer el regulador más corto o darlo por bueno hasta comprobar el funcionamiento experimental (apenas son un par de decibelios y tenemos 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.

Control centralizado H2 o H-infinito

Planta Generalizada no ponderada



Planta generalizada en representación interna:

```
orden=size(forn_esc.A,1);
nrefs=11;
Bgen=[zeros(orden,nrefs) forn_esc.B]; %entradas referencia y control
%salidas planta generalizada: error, acc. control, error_a_contr
Cgen=[-forn_esc.C;zeros(3,orden);-forn_esc.C];
Dgen=[eye(nrefs) -forn_esc.D;zeros(3,nrefs) eye(3);eye(nrefs) -forn_esc.D];
PlantaGeneralizada=ss(forn_esc.A,Bgen,Cgen,Dgen);
size(PlantaGeneralizada)
```

State-space model with 25 outputs, 14 inputs, and 11 states.

Alternativa: Planta generalizada "por bloques"

```
%PlantaGenPrueba2=minreal([eye(11) -forn_esc; ...
%                               zeros(3,11) eye(3); ...
%                               eye(11) -forn_esc]); %sale duplicada la planta
PlantaGenPrueba2=minreal( ...
[ [eye(11) ;zeros(3,11); eye(11) ] -[eye(11); zeros(3,11); eye(11)]*forn_esc+[zeros(11,3);zeros(3,11);zeros(11,3)]
```

Ambas alternativas son la misma:

```
norm(PlantaGenPrueba2-PlantaGeneralizada,'inf')
```

```
ans = 4.5589e-16
```

Diseño H-2 (LQR+Kalman)

Si queremos minimizar la integral de la respuesta "determinista" ante escalón de referencia

$1/s$... bueno $\int_0^\infty (\|err(t)\|^2 + \|u(t)\|^2) dt$ dado que todo está escalado para que las unidades sean comparables, tendremos:

```
s=tf('s');
Win=blkdiag(1/(s+.001)*eye(11),tf(eye(3)));

PlantaPonderadaH2=minreal(PlantaGeneralizada*Win);
PlantaPonderadaH2= ...
```

```

ss(blkdiag(eye(11),2.5*blkdiag((s+0.001)/(1*s+1)*eye(2),1)*V',eye(11))*PlantaGeneraliz
%PlantaPonderadaH2= ...
%ss(blkdiag(blkdiag((s+1)/(s+0.001)*eye(2),1)*U',eye(3)*5,eye(11))*PlantaGeneralizada*V

[K_h2,~,GAM2,~]=h2syn(PlantaPonderadaH2,11,3); GAM2 %no tiene nada que ver con "1"

GAM2 = 67.1361

```

```
size(K_h2)
```

State-space model with 3 outputs, 11 inputs, and 24 states.

```

CL2=lft(PlantaGeneralizada,K_h2);
dcgain(U(:,1:3)'*CL2(1:11,:)*U(:,1:3))

```

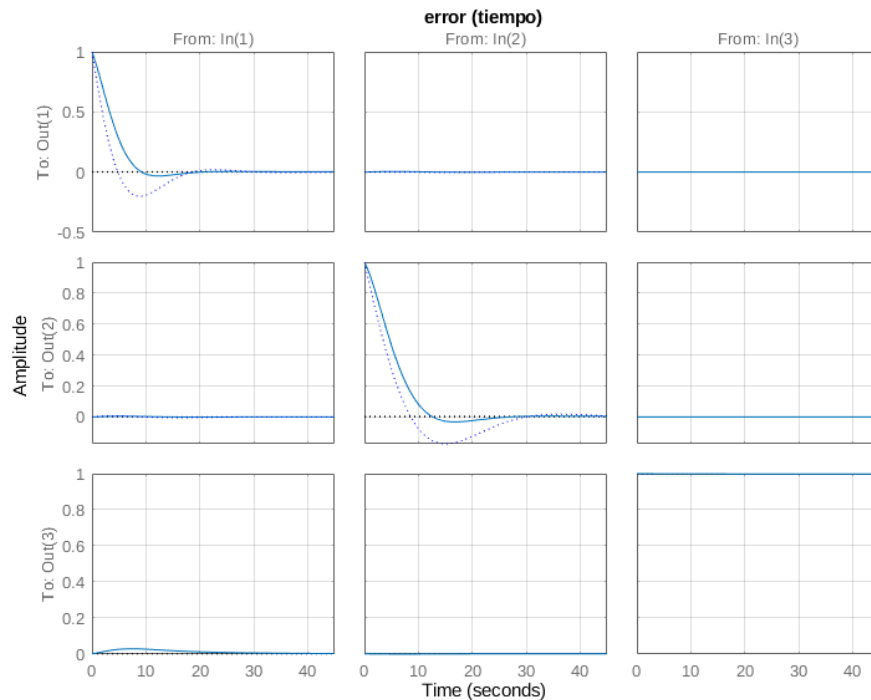
```

ans = 3×3
    0.0033    0.0000    0.0005
    0.0000    0.0047   -0.0000
    0.0000   -0.0000    0.9985

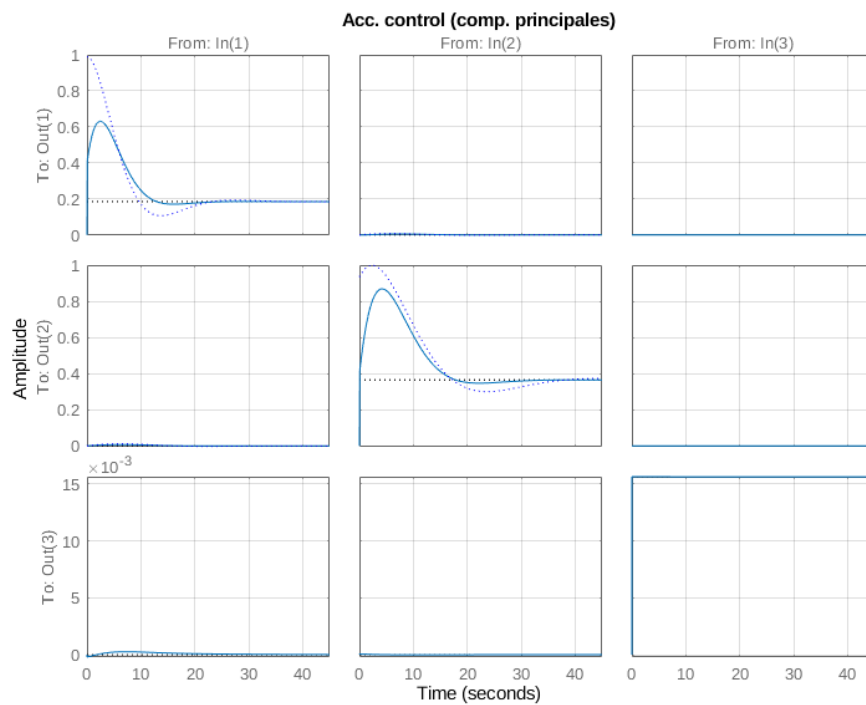
```

Comparando la respuesta temporal, se parece mucho al desacoplamiento SVD:

```
step(U(:,1:3)'*CL2(1:11,:)*U(:,1:3),ref_a_error,':',45), grid on, title("error (tiempo)
```



```
step(V'*CL2(12:14,:)*U(:,1:3),CLTotu,':',45), grid on, title("Acc. control (comp. princ
```

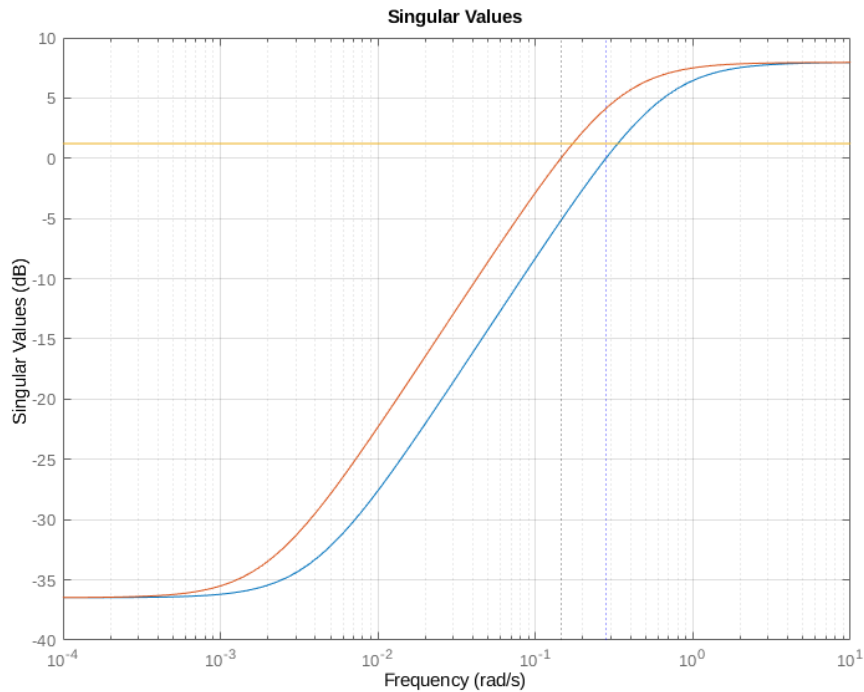


Diseño H-infinito ($\approx$ sensibilidad mixta)

```
anchobandamaniobras=[.28 .145]; %queremos acercar a ancho banda del sigma.
plantillaErr{1}=makeweight(0.015,anchobandamaniobras(1),2.5);
plantillaErr{2}=makeweight(0.015,anchobandamaniobras(2),2.5);
plantillaErr{3}=tf(1.15);%que no tenga "picos" nos vale
Werr= ...
    minreal(blkdiag(1/plantillaErr{1},1/plantillaErr{2},1/plantillaErr{3})*U');
```

2 states removed.

```
sigma(plantillaErr{:}), grid on, xline(anchobandamaniobras(1),':b'),xline(anchobandamar
```



```
Wu=tf(eye(3));
Wout=blkdiag(Werr,Wu,eye(11));
```

Planta generalizada ponderada:

```
PlantaGenPonderada=minreal(Wout*PlantaGeneralizada);
```

Controlador óptimo $\mathcal{H}_\infty$:

```
[K,~,GAM,~]=hinfsyn(PlantaGenPonderada,11,3);GAM
```

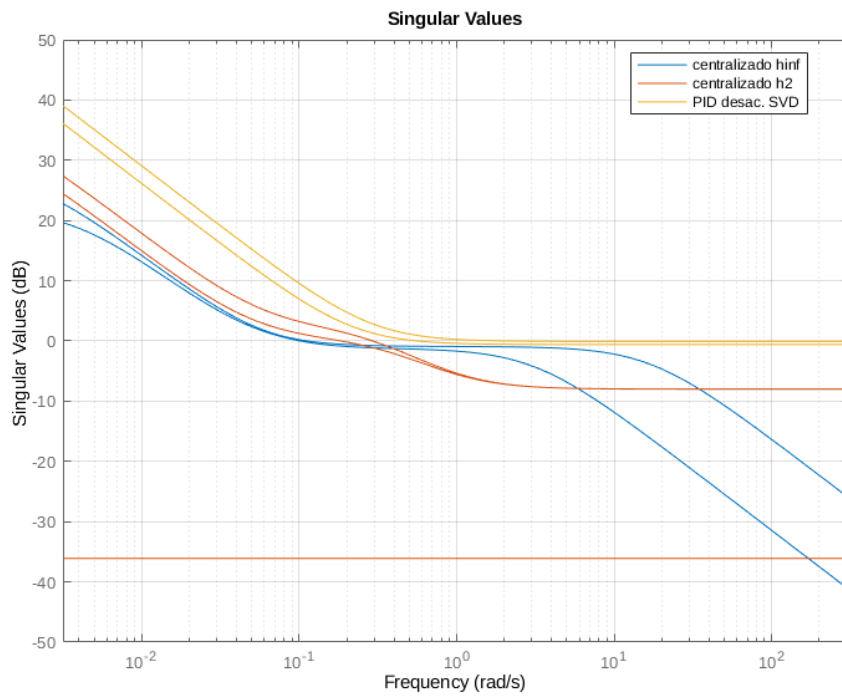
```
GAM = 0.9995
```

```
size(K)
```

```
State-space model with 3 outputs, 11 inputs, and 13 states.
```

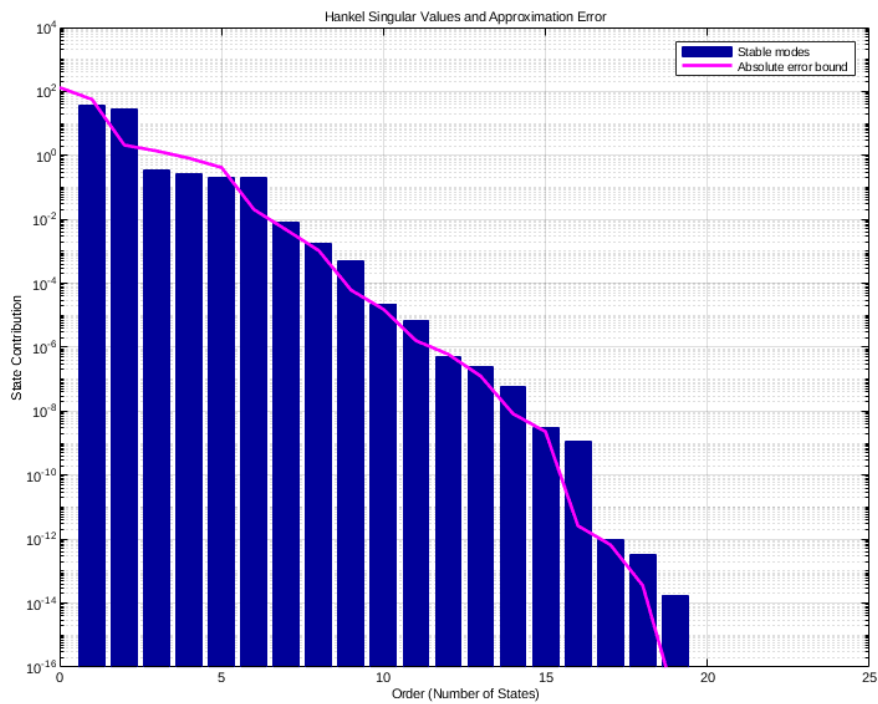
Comparamos las ganancias de los reguladores:

```
sigma(K,K_h2,K2Pitot,logspace(-2.5,2.5,200)), ylim([-50,50]), grid on,
legend("centralizado hinf","centralizado h2","PID desac. SVD",Location="best")
```



Reducción de orden de reguladores

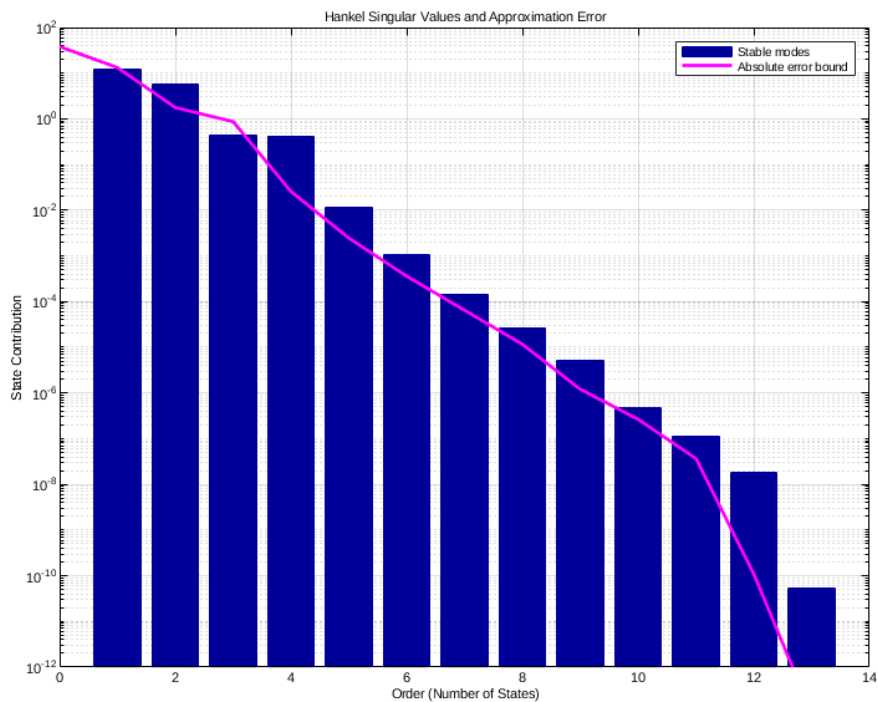
```
balred(K_h2)
```

```
K_h2reducido=balred(K_h2,3);
CLpond=lft(PlantaPonderadaH2,K_h2reducido); %apenas perdemos prestaciones por reducir a
format long
[norm(CLpond,2) GAM2]
```

```
ans = 1×2
    67.141796159497233    67.136087707925782
```

```
balred(K)
```



```
Kreducido=balred(K,2);
CLpond=lft(PlantaGenPonderada,Kreducido); %apenas perdemos prestaciones por reducir a c
[norm(CLpond,'inf') GAM]
```

```
ans = 1x2
    1.003233054578371    0.999501393396391
```

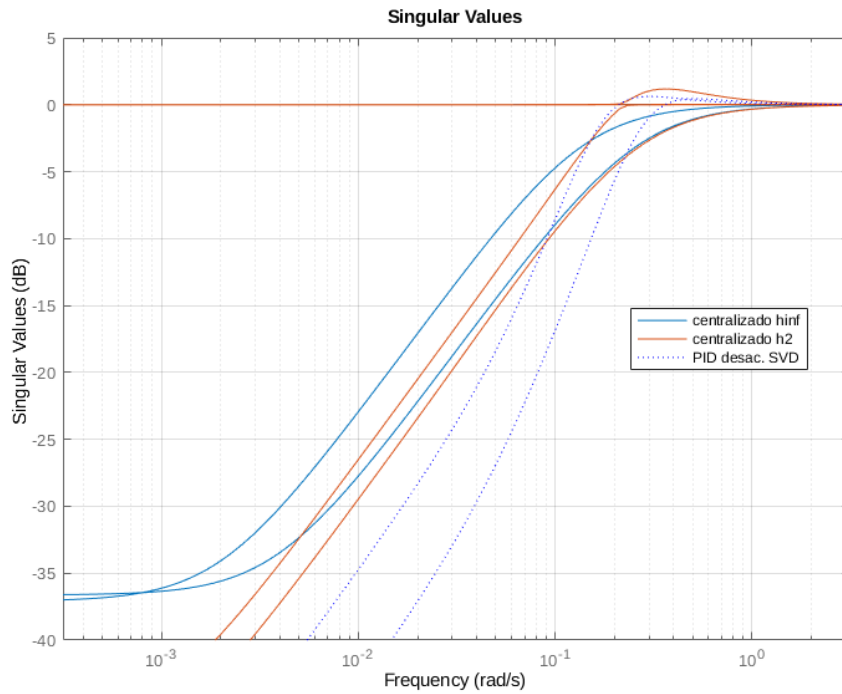
```
format short
```

Análisis de propiedades de control resultantes en tiempo y frecuencia (unidades originales):

```
CLinf=lft(PlantaGeneralizada,Kreducido);
CL2=lft(PlantaGeneralizada,K_h2reducido);
size(CLinf)
```

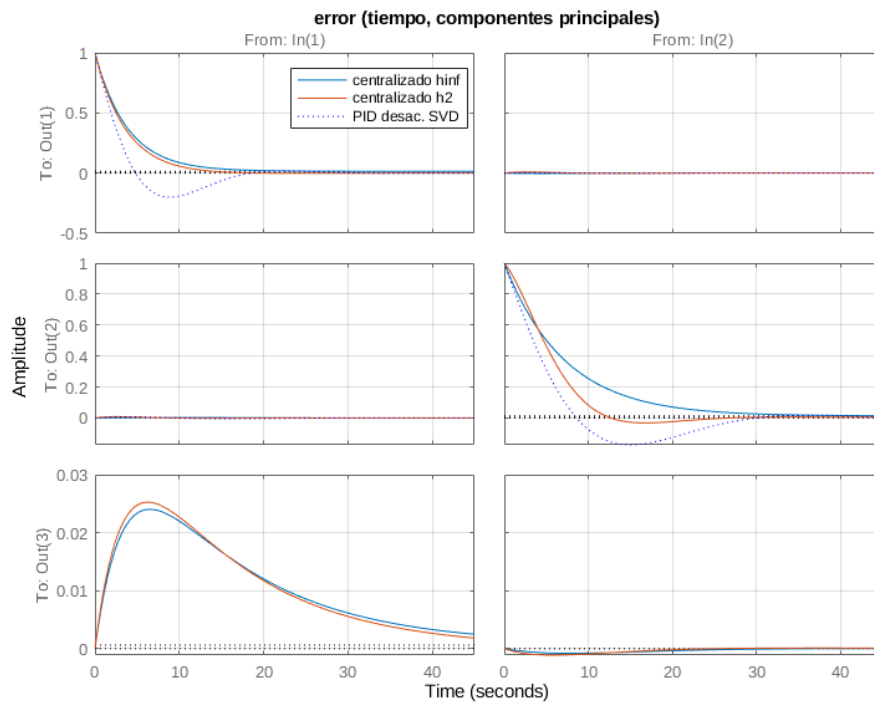
```
State-space model with 14 outputs, 11 inputs, and 13 states.
```

```
sigma(CLinf(1:11,:),CL2(1:11,:),ref_a_error,':',logspace(-3.5,0.5,100)), grid on, ylim
```

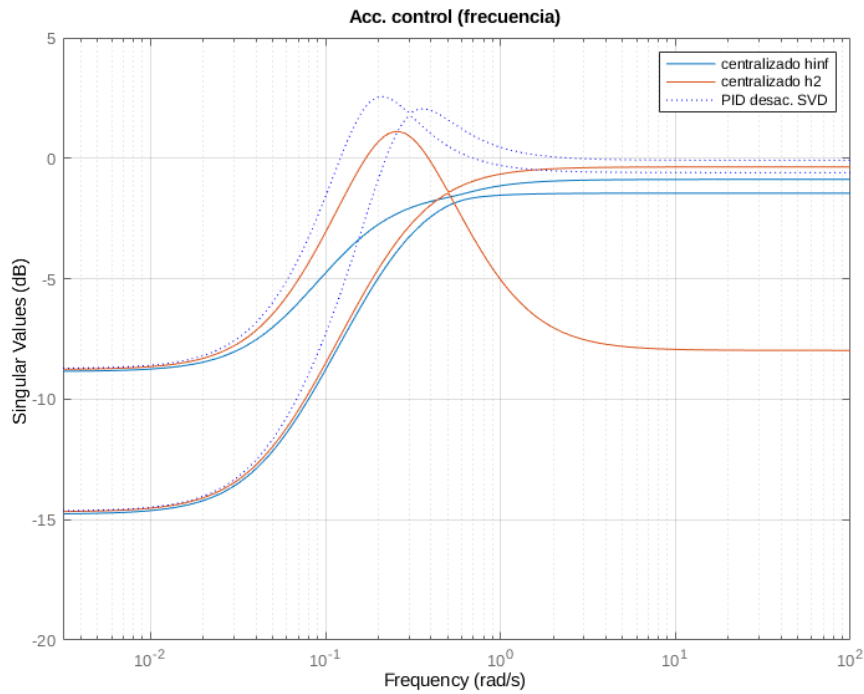


Comparando la respuesta temporal

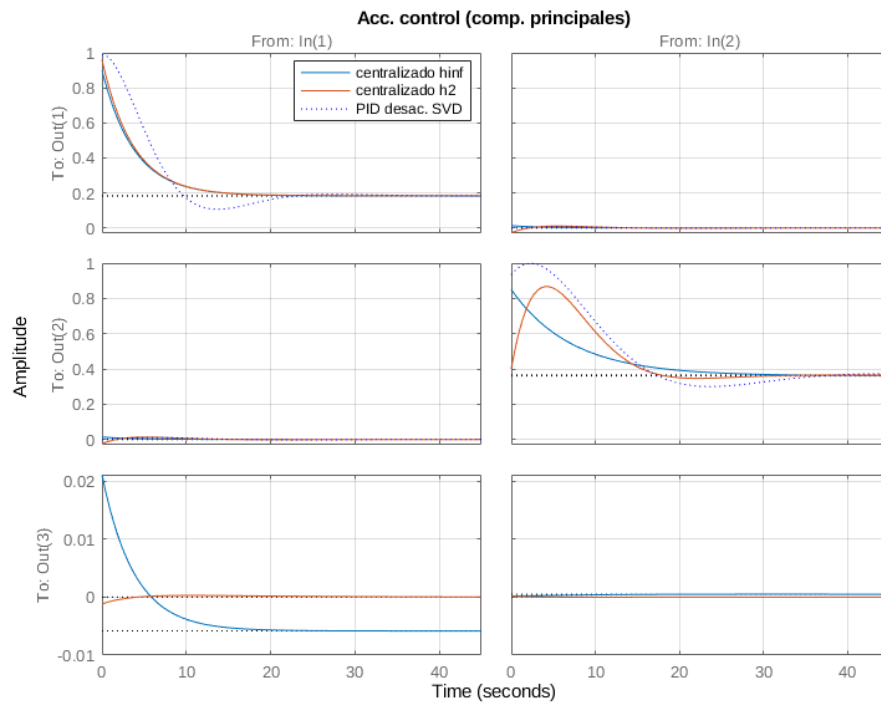
```
step(U(:,1:3)'*CLinf(1:11,:) *U(:,1:2),U(:,1:3)'*CL2(1:11,:) *U(:,1:2),ref_a_error,':',45)
```



```
sigma(CLinf(12:14,:),CL2(12:14,:),CLTotu,':',logspace(-2.5,2,250)), ylim([-20 5]), grid
```

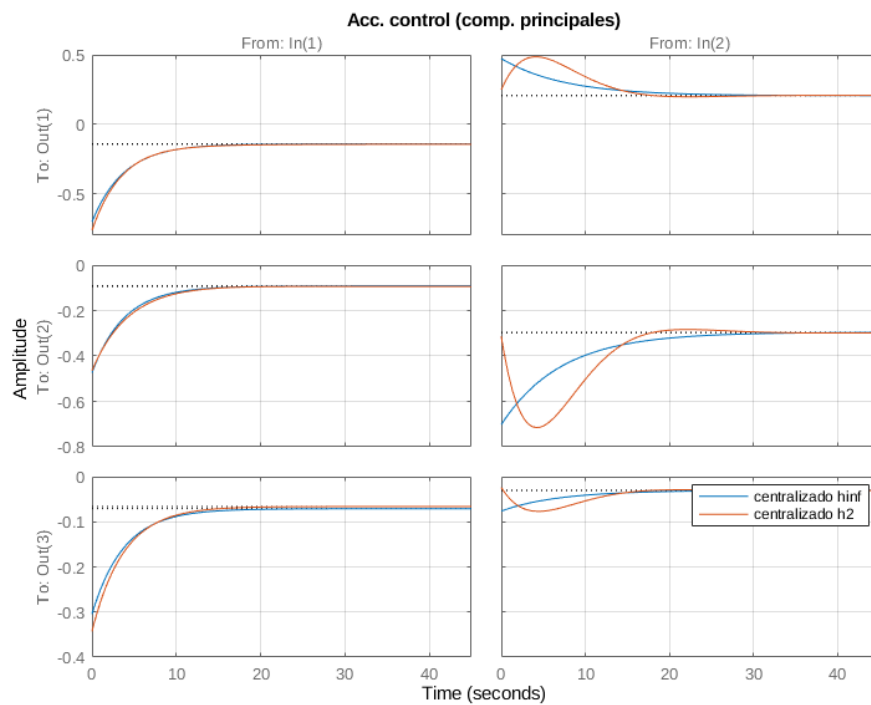


```
step(V'*CLinf(12:14,:)*U(:,1:2),V'*CL2(12:14,:)*U(:,1:2),CLTotu,':',45), grid on, title
```



```
step(CLinf(12:14,:)*U(:,1:2),CL2(12:14,:)*U(:,1:2),45), grid on, title("Acc. control (comp. principales)")
```

Warning: Ignoring extra legend entries.



Robustez de los diseños de control

Aunque depende de "escalados" y de "distribución en frecuencia de la incertidumbre", una forma rápida de evaluar (aparte de, claro, probar experimentalmente si la "simulación" se parece a la "práctica" que es "lo único que realmente importa") todos los picos de respuesta en frecuencia entre pert. entrada/salida y acc.control/vbles controladas serán bajos si el margen "normalised coprime factor" es grande:

```
ncfmargin(forn_esc,K_h2) %h2syn
```

```
ans = 0.5430
```

```
ncfmargin(forn_esc,K_h2reducido)
```

```
ans = 0.5426
```

```
ncfmargin(forn_esc,K) %hinfsyn
```

```
ans = 0.6932
```

```
ncfmargin(forn_esc,Kreducido)
```

```
ans = 0.7044
```

```
ncfmargin(forn_esc,V(:,1:2)*K2PItot*U(:,1:2)') %el de PIDs, en unidades físicas
```

```
ans = 0.4758
```

Conclusiones

Hemos sintonizado:

- a)** 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.
- b)** reguladores óptimos $\mathcal{H}_2$, con distintas opciones de pesos.
- c)** Regulador $\mathcal{H}_\infty$, con anchos de banda diferentes para cada maniobra SVD (en peso de salida).

Para que el orden de los diseños $\mathcal{H}_2$ y $\mathcal{H}_\infty$ sea aceptable, éste debe ser reducido.

Los tres diseños dan resultados "parecidos", jugando adecuadamente con los pesos y las diferentes opciones de diseño, si no hay saturación de las entradas.