

Diseño "mixed sensitivity" 2x2 ante incertidumbre aditiva: incorporación ucover con direccionalidad, desacoplamiento SVD

© 2020, Antonio Sala Piqueras, Universitat Politècnica de Valencia. Todos los derechos reservados.

Presentaciones en vídeo:

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

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

Este código funcionó correctamente con Matlab R2020a

Objetivo: comprender cómo usar sensibilidad mixta (mixsyn) para diseñar un controlador con robustez garantizada ante incertidumbre aditiva en un caso multivariable. Comprobar las diferentes opciones de "ucover" que pueden aplicarse si se usa mixsyn, así como la posibilidad de hacer más rápidas aquellas maniobras asociadas a direcciones de alta ganancia en las que el error de modelado es porcentualmente menos importante.

Tabla de Contenidos

1. Modelado y estimación de error de modelado (familia de plantas a estabilizar simultáneamente).....	1
Uso de UCOVER para obtener una cota del error de modelado.....	3
2. Diseño MIXED SENSITIVITY.....	5
3. Simulación del resultado (modelo nominal).....	6
4. Respuesta temporal final ante familia de plantas.....	11
Conclusiones.....	12

1. Modelado y estimación de error de modelado (familia de plantas a estabilizar simultáneamente)

Consideremos el proceso:

```
s=tf('s');  
G=minreal(ss([6.7/(s+1)^2 4/(0.8*s+1)^2;  
-13/(s+1)^2 -2.7/(0.5*s+1)^2]));
```

Para desacoplamiento SVD (estático) descompondremos la ganancia como USV':

```
[U,S,V]=svd(dcgain(G)); diag(S)'
```

```
ans = 1x2  
15.2390 2.2252
```

Haciendo adecuados cambios de variable en entradas y salidas $u = V_{USVD}$, $y = U_{USVD}$, el proceso se ve "diagonal" a baja frecuencia:

```
G_SVD=U'*G*V; dcgain(G_SVD)
```

```
ans = 2x2  
15.2390 -0.0000
```

-0.0000 2.2252

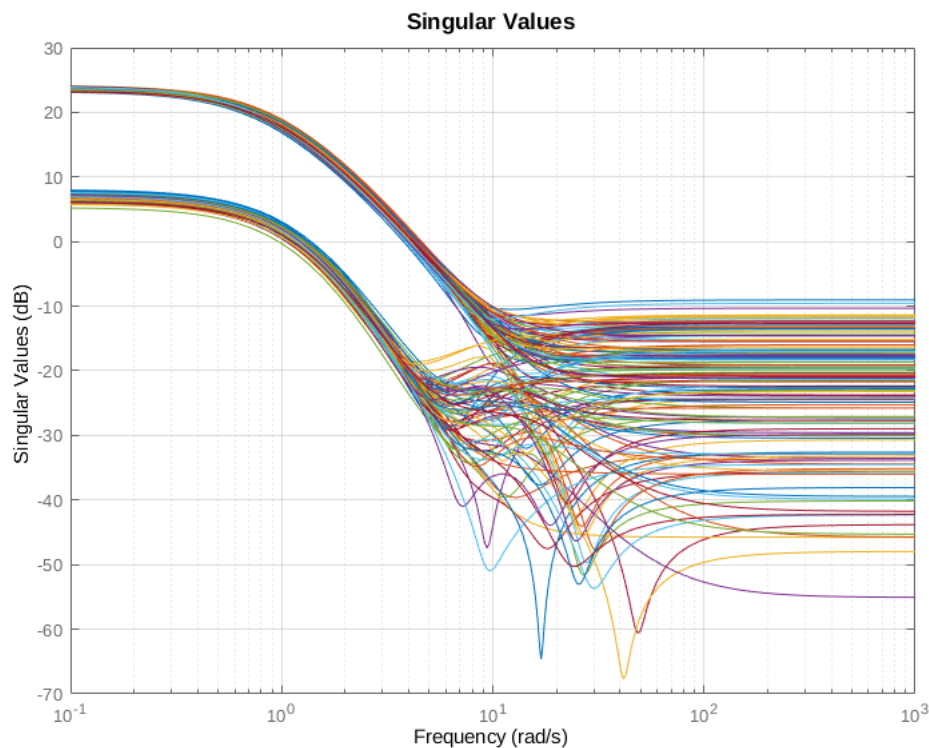
Estimación de cotas de error (familia de plantas)

Consideremos la posibilidad de que el proceso real esté en determinado intervalo de incertidumbre alrededor de él:

```
rng(42123) %para repetibilidad y comparaciones con otros métodos
randinter=@(low,up) low+rand()*(up-low);
Ntests=70;
for i=1:Ntests
    taul=randinter(0.9,1.1);
    Greal{i}=minreal(ss([randinter(6.5,6.9)/(taul*s+1)^2 randinter(3.8,4.2)/(randinter(0.45,0.55)*s+1)^2
        -randinter(12,14)/(taul*s+1)^2 -randinter(2.5,2.9)/(randinter(0.45,0.55)*s+1)^2
        +randn(2)*.09*s/(s+10)],[],false);
    errormod{i}=G-Greal{i};
    GrealSVD{i}=U'*Greal{i}*V;
    errormodSVD{i}=G_SVD-GrealSVD{i};
    normerr{i}=norm(errormod{i},inf);
    %estimamos el tamaño del error aditivo
end
cotaerrmod=max([normerr{:}]) %cota aditiva no estructurada constante...
```

cotaerrmod = 1.3317

sigma(Greal{:}), grid on %resp. en frecuencia de la familia de plantas

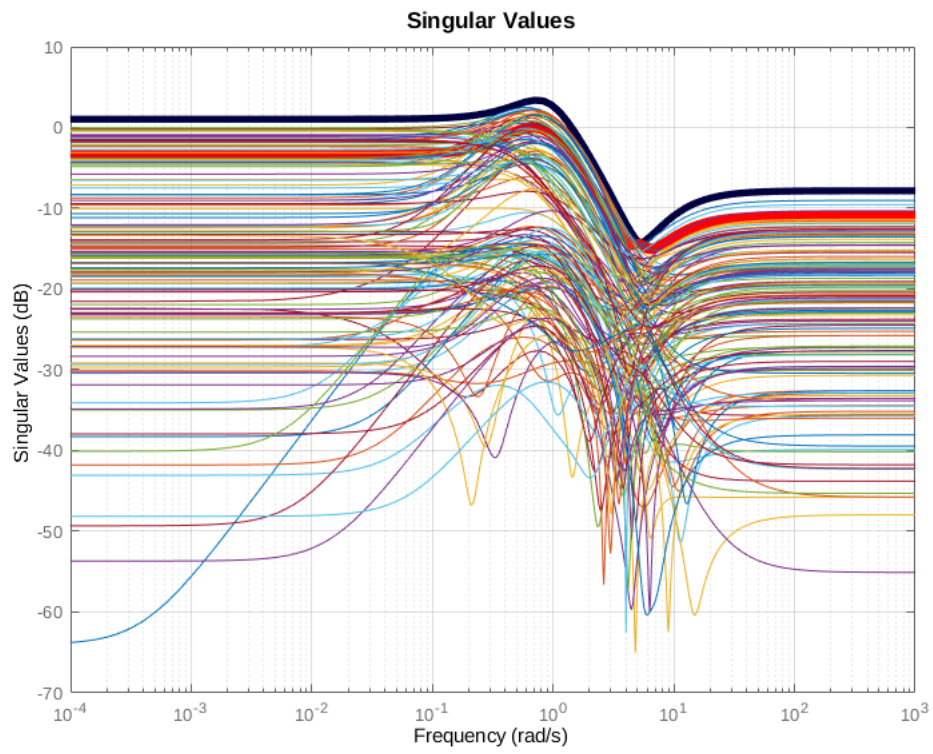


Uso de UCOVER para obtener una cota del error de modelado

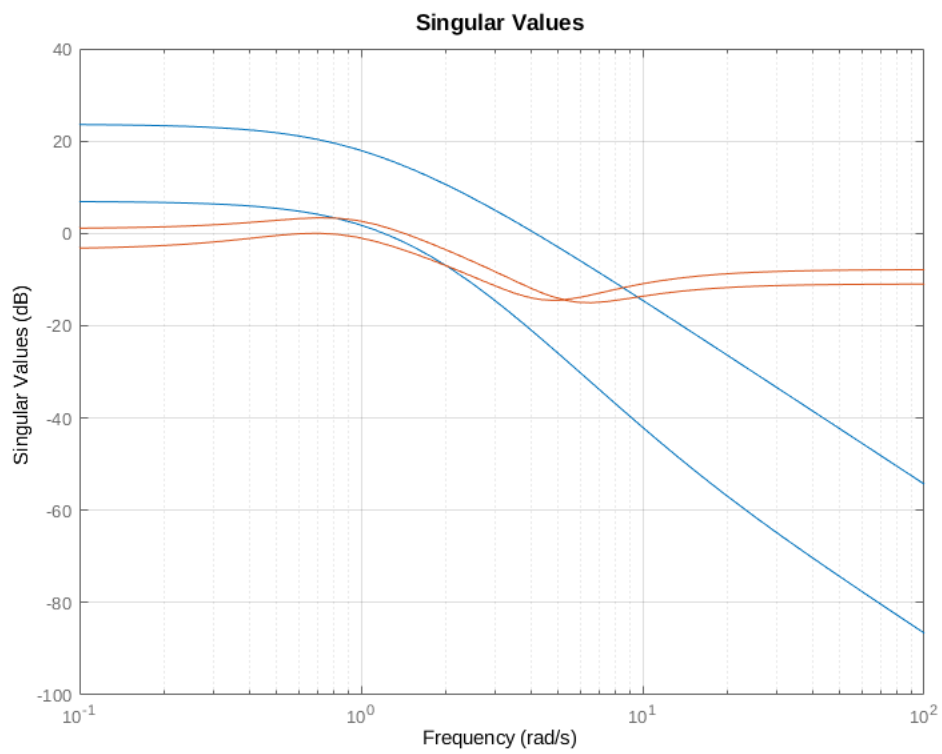
```
w=logspace(-1,2,80); %frecuencias de interés
caso="COVER SVD";
switch caso
    case "COVER ESCALAR"
        Parr=stack(1,Greal{:});
        mPf=frd(Parr,w); %frequency-response data model
        [Pr,Info]=ucover(mPf,G,[],3,'additive');
        tamanyoerrormodelado=minreal(Info.W1);
        %cubrimos Greal dentro de G+Delta*w(s) con w 1x1 orden 3

    case "COVER DIAGONAL"
        Parr=stack(1,Greal{:});
        mPf=frd(Parr,w); %frequency-response data model
        [Pr,Info]=ucover(mPf,G,[],[3;3],'additive');
        tamanyoerrormodelado=minreal(Info.W2);
        %cubrimos Greal dentro de G+Delta*W(s) con W 2x2 diagonal orden 3 cada elemento

    case "COVER SVD"
        Parr=stack(1,GrealSVD{:});
        mPf=frd(Parr,w); %frequency-response data model
        [Pr,Info]=ucover(mPf,G_SVD,[],[3;3],'additive');
        tamanyoerrormodelado= minreal(Info.W2)*V';
        %cubrimos GrealSVD=G_SVD+Delta*W(s) con W 2x2 diagonal orden 3 cada elemento
        %cubrimos U'*Greal*V=U'*G*V+Delta*W(s) con W 2x2 diag. orden 3 cada elemento
        %y deshacemos el cambio de variable: Greal=G+U*Delta*W(s)*V'
        %pero U*Delta tiene las mismas propiedades que Delta:
        % girar una esfera de incertidumbre resulta la misma esfera
        % sólo se necesita Greal=G+Delta*W(s)*V'
end
sigmaplot(tamanyoerrormodelado,'r'), grid on
lines = findobj(gcf,'Type','Line');
lines(2).LineWidth=6;
if(caso ~= "COVER ESCALAR")
    lines(3).LineWidth=4;
    lines(3).Color=[0 0 .25];
end
hold on
sigma(errormod{:})
hold off
```



```
sigma(G,tamanyoerrormodelado), grid on
```



2. Diseño MIXED SENSITIVITY

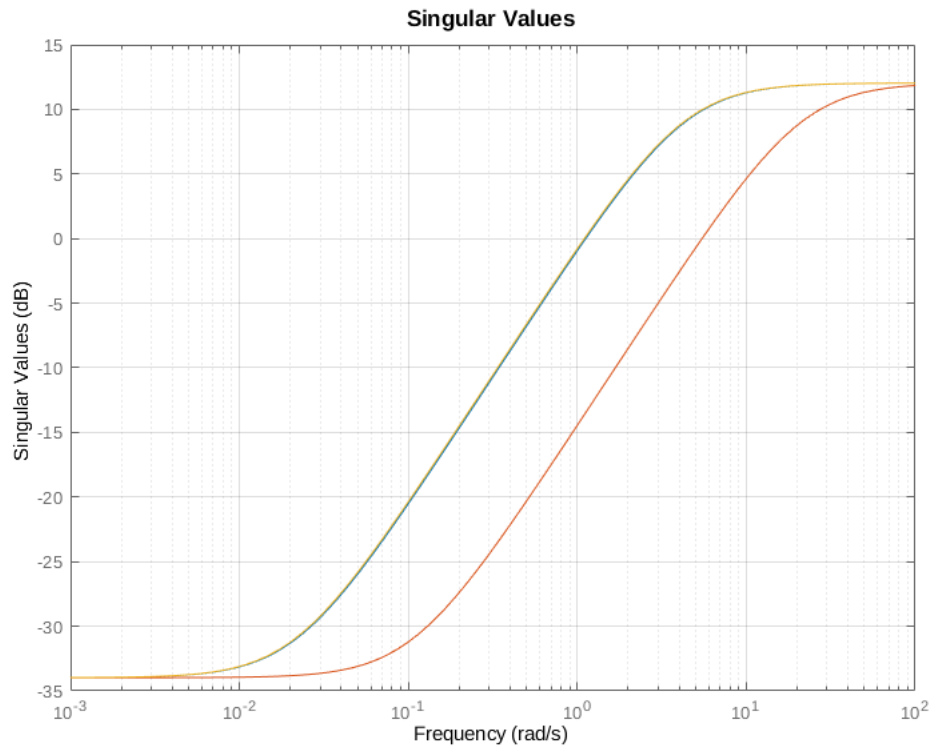
Para cada opción de "ucover" podremos conseguir unas prestaciones (nominales) diferentes.

```
switch caso
case "COVER ESCALAR"
    bw_comun=0.871; %unica
    bw1=5.02;%desac SVD (dirección alta ganancia)
    bw2=0.84;%desac SVD (dirección baja ganancia)
case "COVER DIAGONAL"
    bw_comun=1.14; %unica
    bw1=5.03;%desac SVD (dirección alta ganancia)
    bw2=1.1;%desac SVD (dirección baja ganancia)
case "COVER SVD"
    bw_comun=1.12; %unica
    bw1=5.47;%desac SVD (dirección alta ganancia)
    bw2=1.1;%desac SVD (dirección baja ganancia)

end
```

Construimos las plantillas común versus diferentes para cada maniobra del desac SVD:

```
plantillacomunerror=makeweight(0.02,bw_comun,4);
plantillaerror_svd1=makeweight(0.02,bw1,4); %maniobra "rápida" (mayor ganancia)
plantillaerror_svd2=makeweight(0.02,bw2,4); %maniobra "lenta" (menor ganancia)
sigma(plantillacomunerror,plantillaerror_svd1,plantillaerror_svd2), grid on
```



[A] El diseño con prestaciones de ancho de banda **único para todas las direcciones** sería:

```
Werr=1/plantillacomunerror*eye(2); tamanyoerrorrelativo=0.05; % 5 por ciento "extra" pi
[K,~,GAM,~]=mixsyn(G,Werr,tamanyoerrormodelado,tamanyoerrorrelativo); GAM

GAM = 0.9994
```

[B] El diseño usando la **direccionalidad del SVD** a baja frecuencia sería:

```
Werr=minreal(blkdiag(1/plantillaerror_svd1,1/plantillaerror_svd2))*U';
[Ksvd,~,GAM,~]=mixsyn(G,Werr,tamanyoerrormodelado,tamanyoerrorrelativo); GAM

GAM = 0.9977
```

3. Simulación del resultado (modelo nominal)

```
L=minreal(G*K);
```

6 states removed.

```
bcy=feedback(L,eye(2));
bce=feedback(eye(2),L);
Lsvd=minreal(G*Ksvd);
```

6 states removed.

```
bce_svd=feedback(eye(2),Lsvd);
bcy_svd=feedback(Lsvd,eye(2));
```

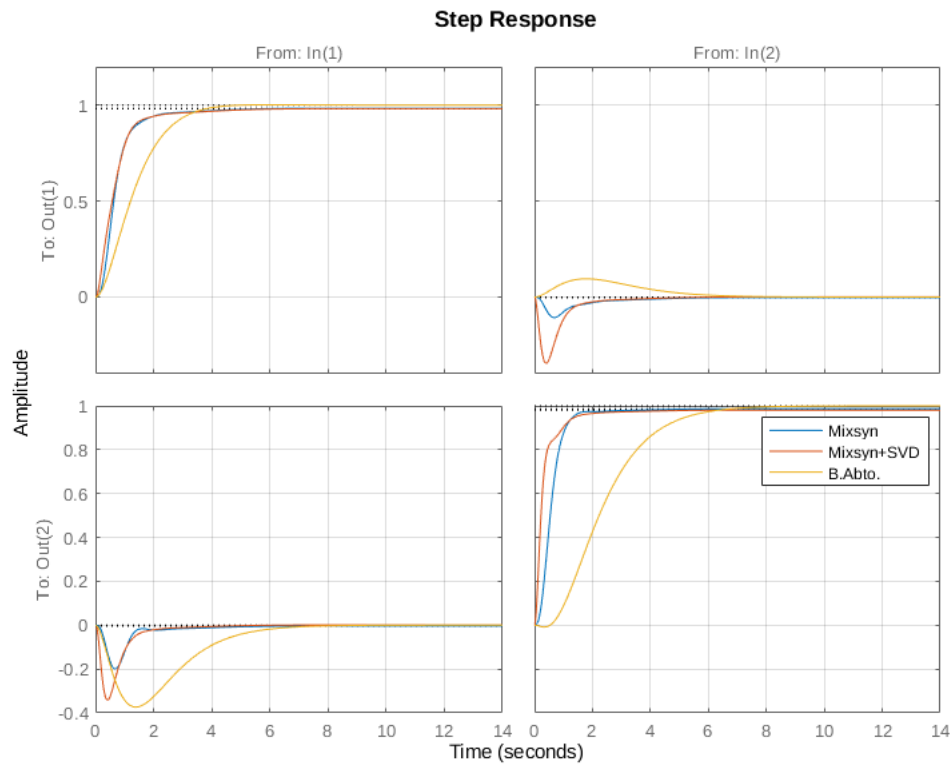
```
bcu=minreal(feedback(K,G));
```

6 states removed.

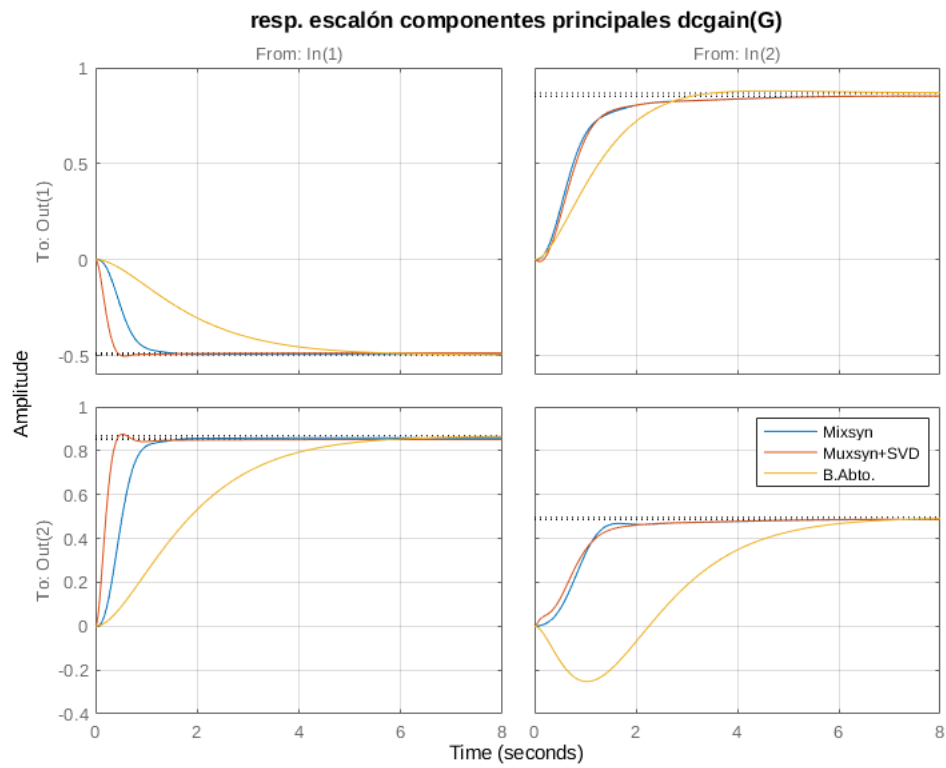
```
bcu_svd=minreal(feedback(Ksvd,G));
```

6 states removed.

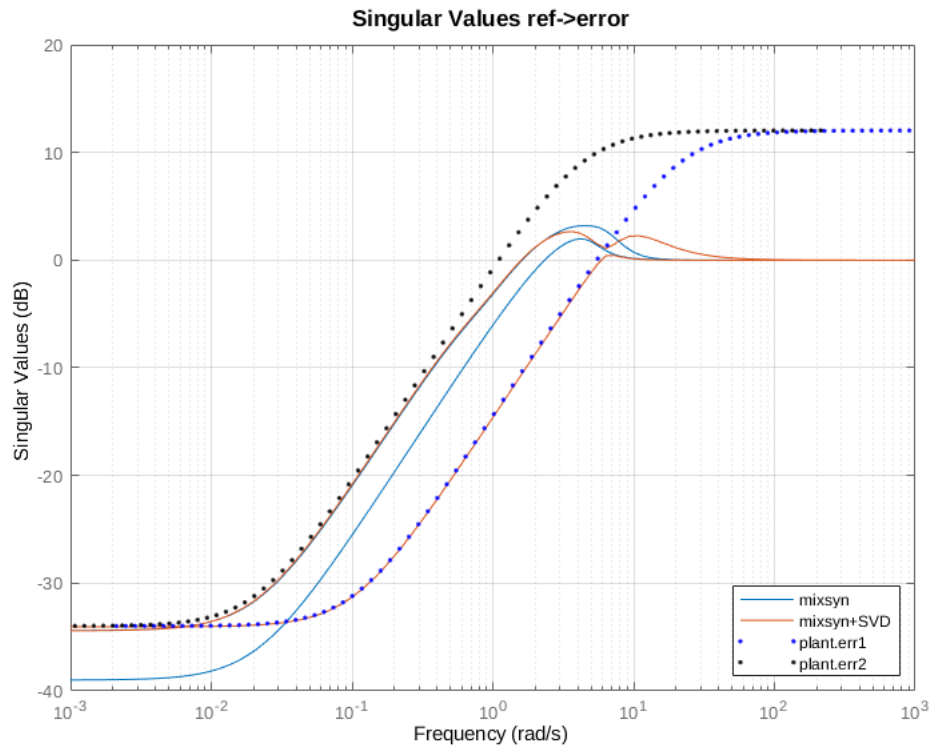
```
step(bcy,bcy_svd,G*inv(dcgain(G)), grid on  
legend('Mixsyn','Mixsyn+SVD','B.Abto.')
```



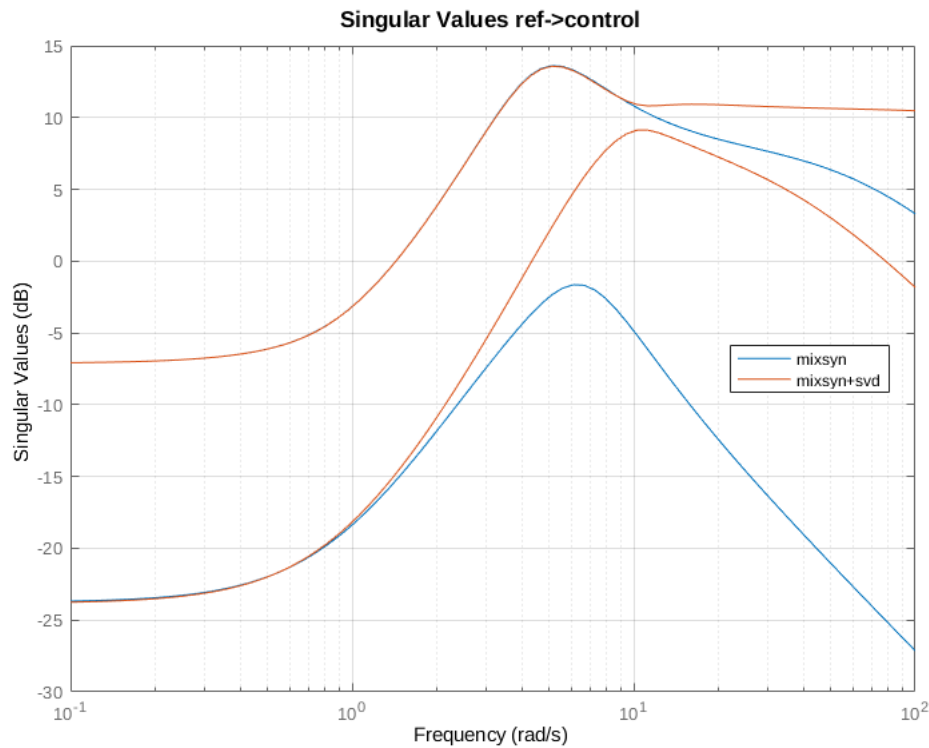
```
step(bcy*U,bcy_svd*U,G*inv(dcgain(G))*U,8), grid on %comparad con los PIDs.  
legend('Mixsyn','Muxsyn+SVD','B.Abto.')  
title('resp. escalón componentes principales dcgain(G)')
```



```
sigma(bce,bce_svd,plantillaerror_svd1,',' ,plantillaerror_svd2,',' ,k'),grid on
legend('mixsyn','mixsyn+SVD','plant.err1','plant.err2','Location','Southeast')
title("Singular Values ref->error")
```



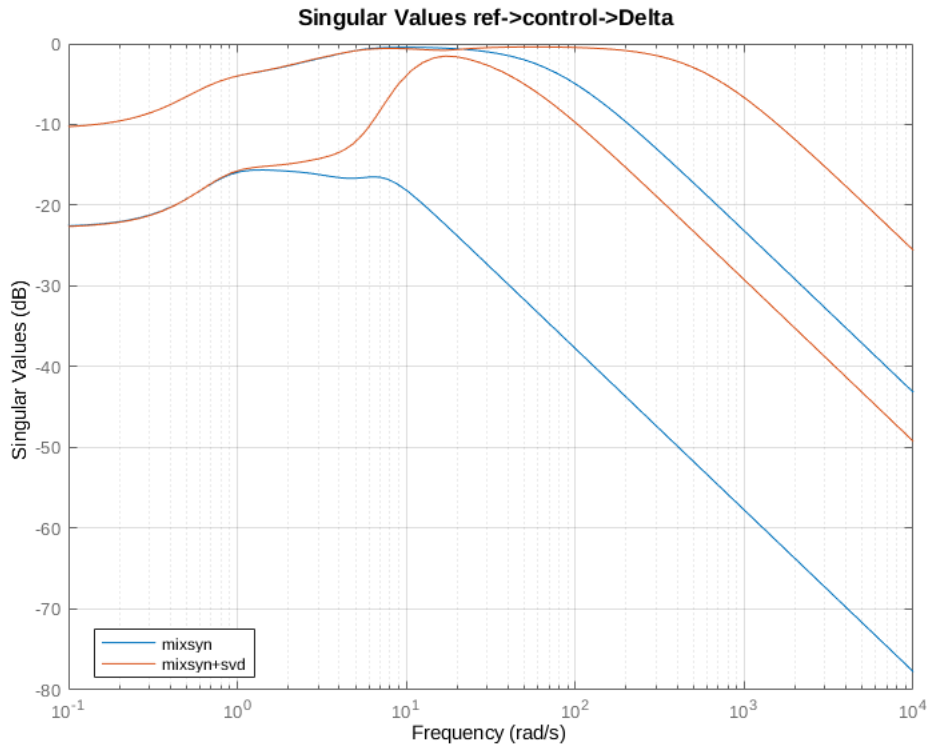
```
sigma(bcu,bcu_svd,w),grid on %supera 1, no estamos teniendo en cuenta "saturación / pla
legend('mixsyn','mixsyn+svd','Location','best'), title("Singular Values ref->control")
```



Podemos comprobar que el error de modelado (con su direccionalidad, de haberla), cumple el teorema de pequeña ganancia:

```
sigma(minreal(tamanyoerrormodelado*bcu),minreal(tamanyoerrormodelado*bcu_svd)),grid on
6 states removed.
6 states removed.
```

```
legend('mixsyn','mixsyn+svd','Location','best')
title("Singular Values ref->control->Delta ")
```



```
norm(tamanyoerrormodelado*bcu,inf)
```

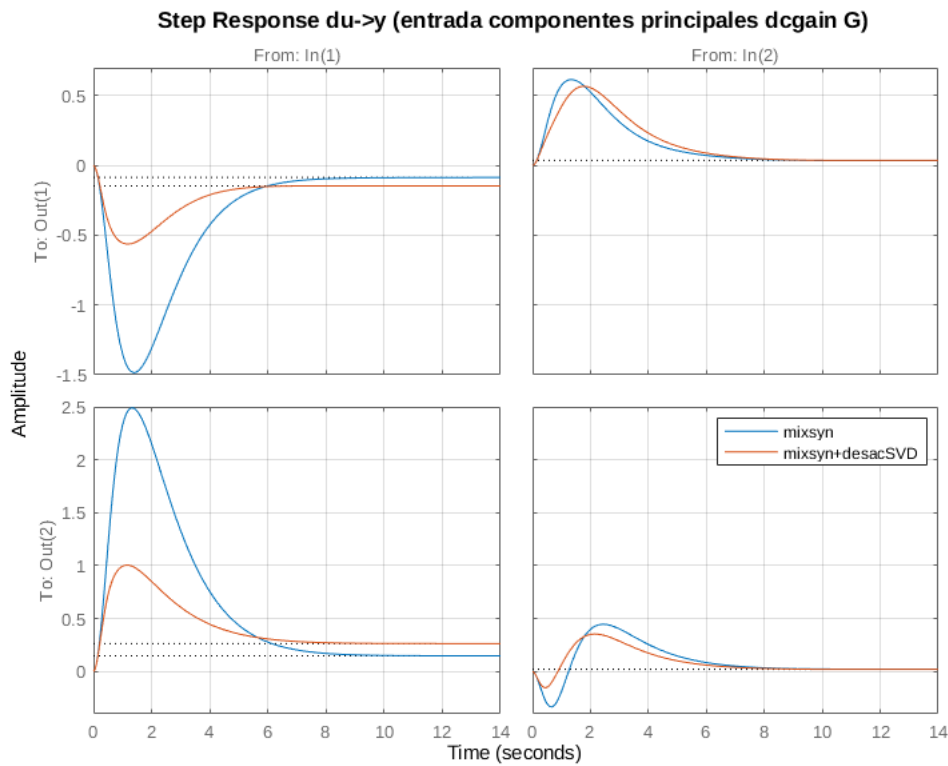
```
ans = 0.9494
```

```
norm(tamanyoerrormodelado*bcu_svd,inf) %los dos son robustos
```

```
ans = 0.9555
```

El mixsyn internamente es un regulador de cancelación (en el caso estable; en el inestable ya no cancela, obviamente). Para polos cercanos al eje imaginario, la respuesta ante perturbación de entrada podría no ser satisfactoria. La dibujamos.

```
bucle_du_to_y=minreal(feedback(G,K));
bucle_du_to_ysvd=minreal(feedback(G,Ksvd));
step(bucle_du_to_y*V,bucle_du_to_ysvd*V), grid on
legend('mixsyn','mixsyn+desacSVD')
title("Step Response du->y (entrada componentes principales dcgain G)")
```

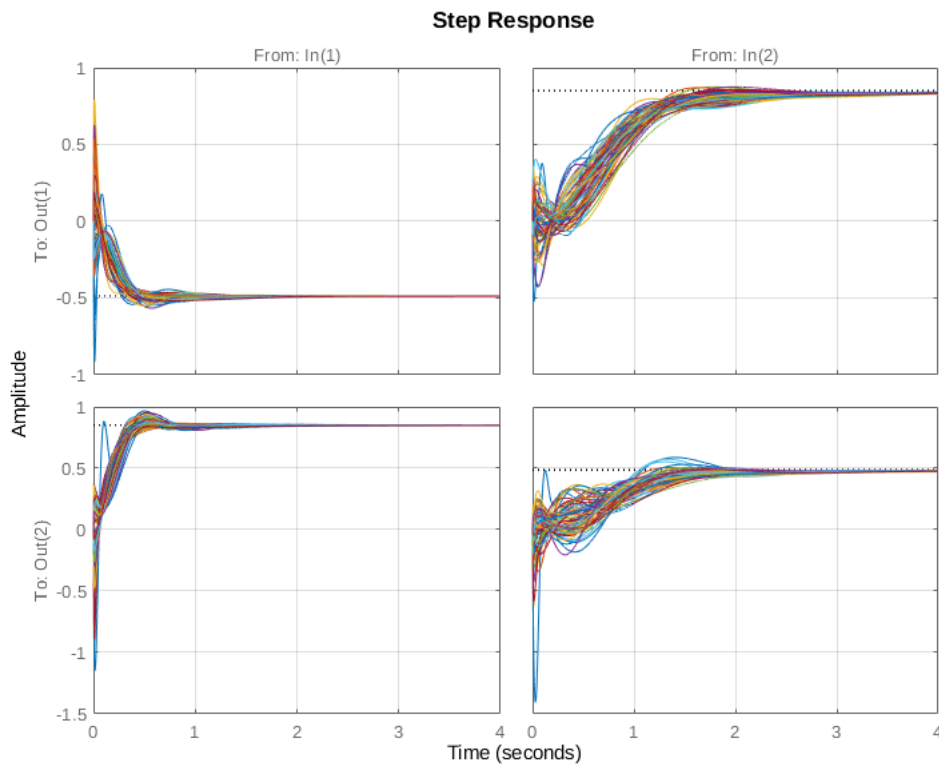


En este caso, es razonable... aunque si pert. entrada tuvieran amplitud grande, podría probarse a modificar la planta generalizada y usar el código del vídeo [mxscan2] para que el optimizador, además de tolerar los errores de modelado prefijados, intentara que la respuesta de perturbación a la entrada fuera mejor, o utilizar Glover-McFarlane (ncfsyn).

4. Respuesta temporal final ante familia de plantas

Como Ksvd parece que "aprovecha" mejor el compromiso especificaciones-robustez, dibujamos la respuesta de este regulador ante todas las plantas de la familia que habíamos cubierto.

```
for i=1:Ntests
    bcreal_y{i}=feedback(minreal(ss(Greal{i})*ss(Ksvd),[],false),eye(2))*U;
end
step(bcreal_y{:},4), grid on
```



Conclusiones

Hemos usado UCOVER para cubrir una familia de sistemas como una "bola" de incertidumbre. Con mixed sensitivity hemos diseñado el regulador que mayor ancho de banda conseguía tolerando la cota de error obtenida.

Hemos comprobado los diferentes anchos de banda que hemos podido conseguir garantizando estabilidad robusta según diferentes opciones de ucover (SISO, diagonal, diagonal+desacSVD) y mixsyn (ancho de banda único, ancho de banda diferente para diferentes maniobras).