Diseño de control (Glover-MCFarlane) para proceso de dos entradas y dos salidas (caso ponderado W1, W2)

Presentación en vídeo: http://personales.upv.es/asala/YT/V/gmf2w2.html

Este código funcionó correctamente con Matlab **R2020a**

Objetivo: comprender cómo usar Glover-McFarlane (**ncfsyn**) para diseñar un controlador con robustez garantizada para un sistema multivariable 2x2. Se comprobará el efecto de pesos.

Tabla de Contenidos

1. Modelado y estimación de error de modelado

Consideremos el proceso:

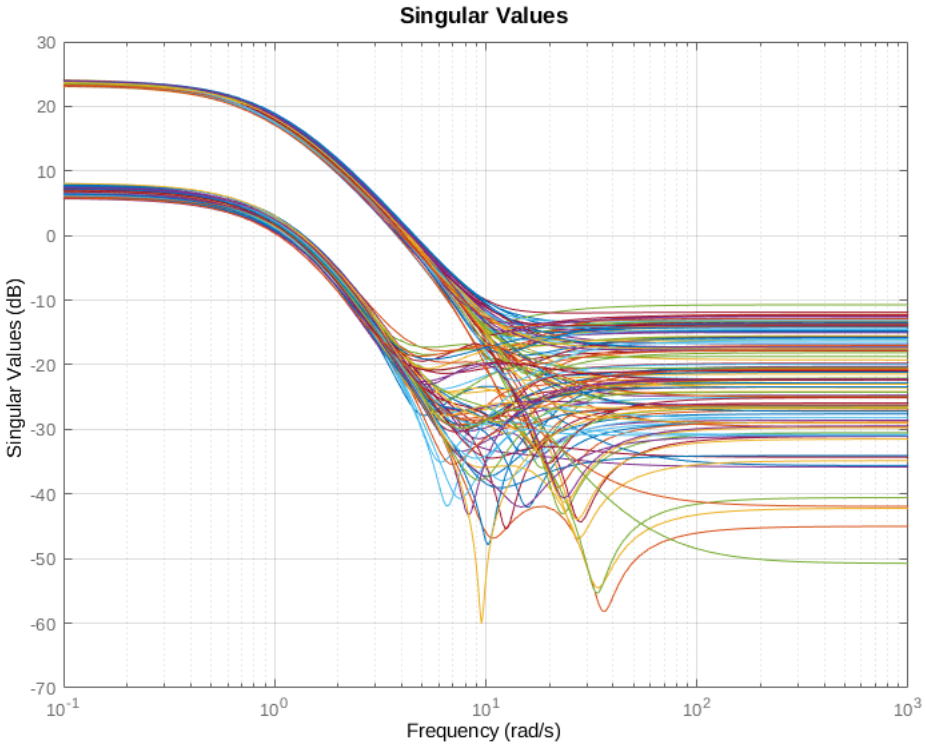
```
s=tf('s');
G=minreal(ss([6.7/(s+1)^2  4/(0.8*s+1)^2;
              -13/(s+1)^2 -2.7/(0.5*s+1)^2]));
size(G)
```

```
State-space model with 2 outputs, 2 inputs, and 6 states.
```

```
ninputs=size(G,2);noutputs=size(G,1);
```

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

```
rng(4343)
rndinte=@(low,up) low+rand()*(up-low);
Ntests=60;
for i=1:Ntests
   tau1=rndinte(0.9,1.1);
   Greal{i}=minreal(ss([rndinte(6.5,6.9)/(tau1*s+1)^2  rndinte(3.8,4.2)/(rndinte(0.75,0
         -rndinte(12,14)/(tau1*s+1)^2 -rndinte(2.5,2.9)/(rndinte(0.45,0.55)*s+1)^2] ...
         +randn(2)*.09*s/(s+10)),[],false);
      [~,nugapm{i}]=gapmetric(G,Greal{i}); %error ncf (sin ponderación)
end
sigma(Greal{:}), grid on
```




```
cotanugap=max([nugapm{:}]) %siempre estará entre 0 y 1 (máxima diferencia en la fact. r
```

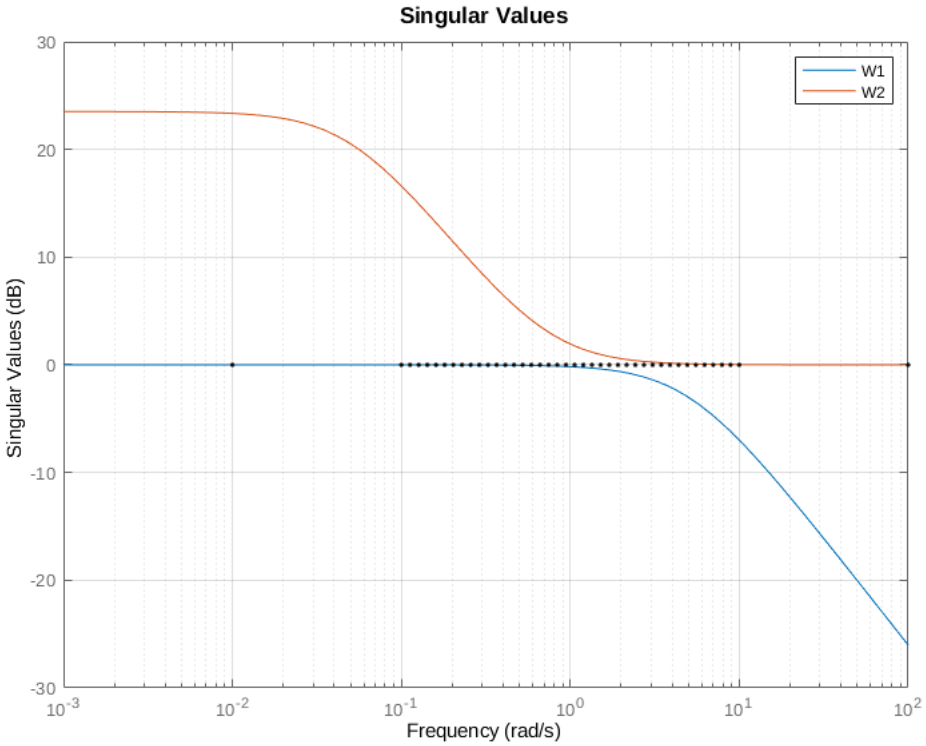
```
cotanugap = 0.2801
```

2. Diseño Glover-McFarlane Loop Shaping (ncfsyn)

2.1 Elección de pesos

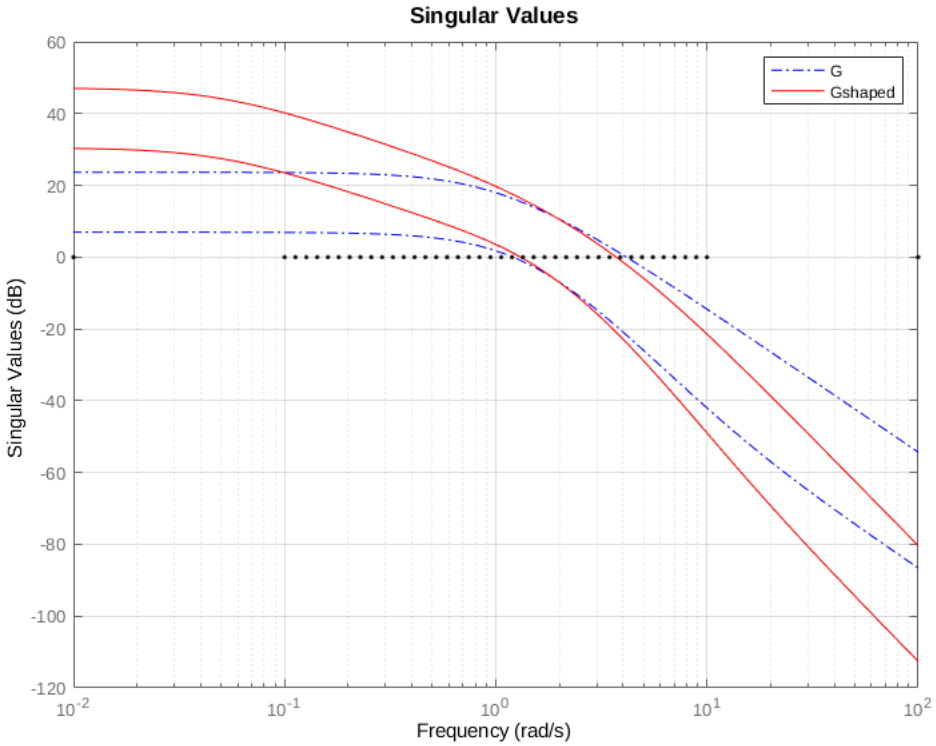
Seleccionemos primero los pesos de la planta ponderada:

```
W1=eye(2)*1/(0.2*s+1); %high freq cut, input weight
W2=diag([1 1])*(20*s+15)/(20*s+1); %low freq boost, output weight
sigma(W1,W2,tf(1),'.k'), grid on, legend('W1','W2')
```



Comparemos la planta original con la ponderada.

```
Gshaped=minreal(ss(W2*G*W1));
sigma(G,'-.',Gshaped,'r',tf(1),'.k'), grid on,legend('G','Gshaped')
```



Evaluemos una cota de incertidumbre entre las plantas reales/nominal ponderadas:

```
for i=1:Ntests
      [~,nugapmW{i}]=gapmetric(Gshaped,W2*Greal{i}*W1); %error ncf ponderado
end
cotanugapW=max([nugapmW{:}])
```

```
cotanugapW = 0.1778
```

2.2 Diseño del controlador ncfsyn y comprobación de márgenes

```
[K,~,GAM,Info]=ncfsyn(G,W1,W2);
```

Comprobemos estabilidad robusta:

Con plantas ponderadas

```
1/GAM
```

```
ans = 0.3681
```

```
ncfmargin(Gshaped,-inv(W1)*K*inv(W2)) %garantiza estabilidad robusta si mayor que cotar
```

```
ans = 0.3681
```

```
if(1/GAM>cotanugapW)
   disp('Estabilidad robusta <strong>garantizada</strong>')
else
```

```
   disp('Estabilidad robusta <strong>NO GARANTIZADA</strong>')
end
```

Estabilidad robusta **garantizada**

Con plantas sin ponderar:

```
vlm=ncfmargin(G,-K) %garantiza estabilidad robusta si mayor que cotanugap (sin ponderar
```

vlm = 0.3491

```
cotanugap %el peor nugap de la familia de plantas
```

cotanugap = 0.2801

```
if(vlm>cotanugap)
   disp('Estabilidad robusta <strong>garantizada</strong>')
else
   disp('Estabilidad robusta <strong>NO GARANTIZADA</strong>')
end
```

Estabilidad robusta **garantizada**

2.4 Comprobación de respuesta temporal y en frecuencia (nominal)

Cerremos el bucle para comprobar todo en tiempo y frecuencia (nominal).

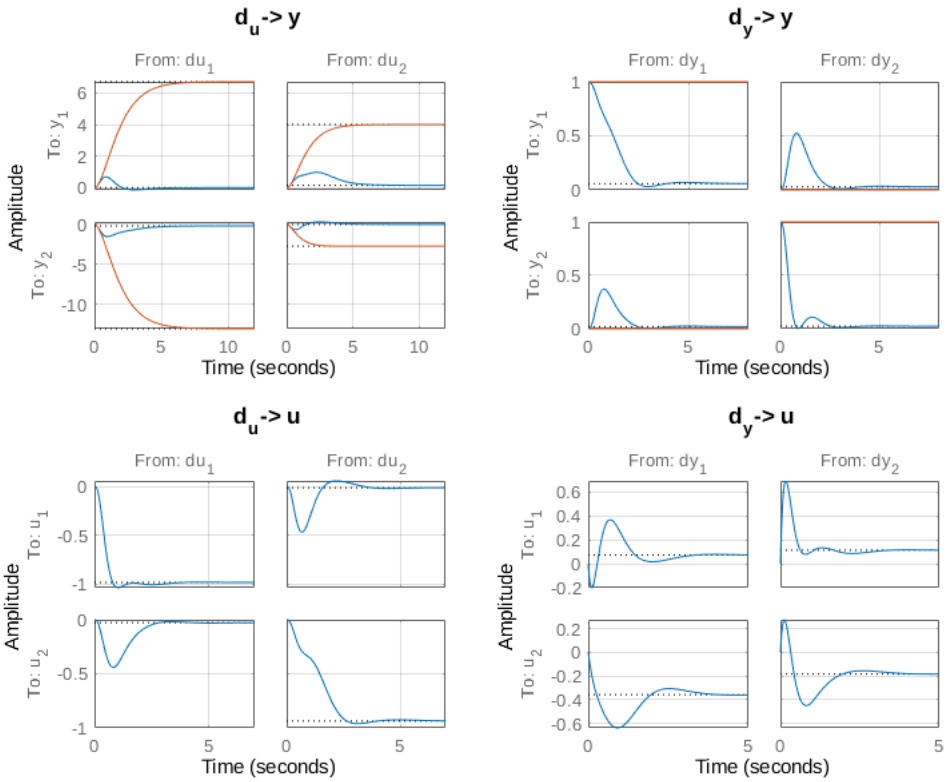
```
PlantaGen=@(G) minreal([G eye(noutputs) G; ...
               zeros(ninputs,2*noutputs) eye(ninputs);...
               G eye(noutputs) G],  [],   false);
PG=PlantaGen(G); size(PG)
```

State-space model with 6 outputs, 6 inputs, and 6 states.

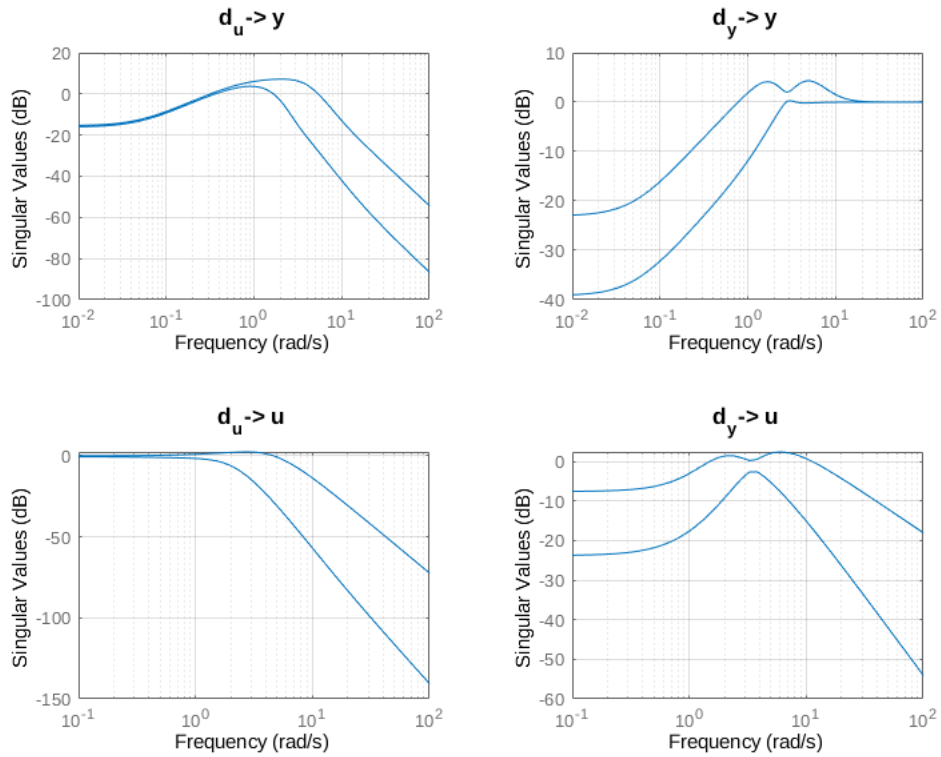
```
loscuatro=minreal(lft(PG,K));
size(loscuatro)
```

State-space model with 4 outputs, 4 inputs, and 19 states.

```
loscuatro.InputName={'du_1','du_2','dy_1','dy_2'};
loscuatro.OutputName={'y_1','y_2','u_1','u_2'};
trozoEdU=loscuatro(1:2,1:2);trozoEdy=loscuatro(1:2,3:4);
trozoUdU=loscuatro(3:4,1:2);trozoUdy=loscuatro(3:4,3:4);
subplot(2,2,1), step(trozoEdU,G), grid on, title('d_u-> y')
subplot(2,2,2), step(trozoEdy,tf(eye(2))), grid on, title('d_y-> y')
subplot(2,2,3), step(trozoUdU), grid on, title('d_u-> u')
subplot(2,2,4), step(trozoUdy), grid on, title('d_y-> u')
```

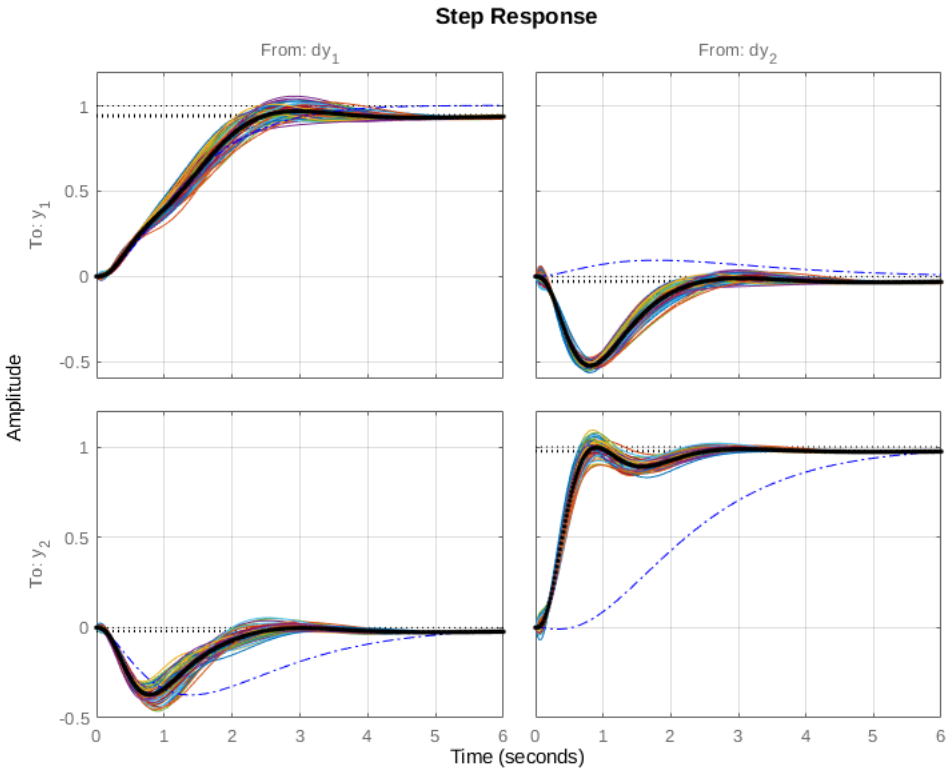



```
figure
subplot(2,2,1), sigma(trozoEdU), grid on, title('d_u-> y')
subplot(2,2,2), sigma(trozoEdy), grid on, title('d_y-> y')
subplot(2,2,3), sigma(trozoUdU), grid on, title('d_u-> u')
subplot(2,2,4), sigma(trozoUdy), grid on, title('d_y-> u')
```

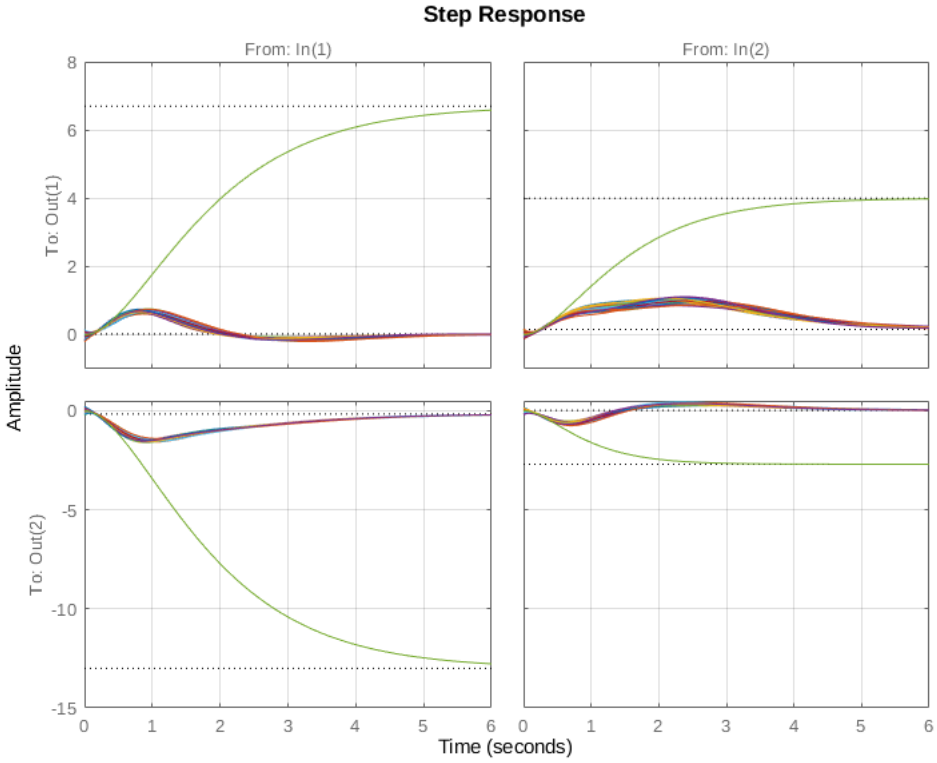


3. Comprobación de robustez: respuesta ante familia de plantas

```
for i=1:Ntests
   PGr{i}=PlantaGen(Greal{i});
   loscuatro_r{i}=minreal(lft(PGr{i},K),[],false);
   lasalidaanteref{i}=eye(2)-loscuatro_r{i}(1:2,3:4);
   lapertentrada{i}=loscuatro_r{i}(1:2,1:2);
end
figure
step(lasalidaanteref{:},G*inv(dcgain(G)),'-.',eye(2)-trozoEdy,'.k',6), grid on
```




```
step(lapertentrada{:},G,6), grid on
```



Conclusiones

Hemos comprobado el efecto de filtros en ncfsyn y el significado del nu-gap (ponderado y sin ponderar).