Análisis de controlabilidad, observabilidad y reducción de orden: ejemplo Matlab

Este código ejecutó sin errores en Matlab **R2022a**

Vídeo-presentación en: http://personales.upv.es/asala/YT/V/roml.html

Tabla de Contenidos

Modelo....1
Controlabilidad....3
Observabilidad....4
Reducción de modelos....4

Modelo

Dos masas de 1 kg y 1 kg, con muelles/amort por en medio...

Entrada: Fuerza sobre masa 2

Estados posición y velocidad masa 1, posición y velocidad masa 2.

```
k1=3;f1=.1;k12=400.1;f12=4.025; M2=1; M1=1;
A=[0 1 0 0; ...
    -(k1+k12)/M1 -(f1+f12)/M1 k12/M1 f12/M1; ...
    0 0 0 1;...
    k12/M2 f12/M2 -k12/M2 -f12/M2]
```

```
A = 4×4
         0    1.0000         0         0
 -403.1000   -4.1250  400.1000    4.0250
         0         0         0    1.0000
  400.1000    4.0250 -400.1000   -4.0250
```

```
eig(A)
```

```
ans = 4×1 complex
  -4.0501 +28.0232i
  -4.0501 -28.0232i
  -0.0249 + 1.2233i
  -0.0249 - 1.2233i
```

```
B=[0; 0; 0;1/M2]
```

```
B = 4×1
     0
     0
     0
     1
```

```
%B=[0 0;1 0;0 0;0 1];%2 fzas?
```

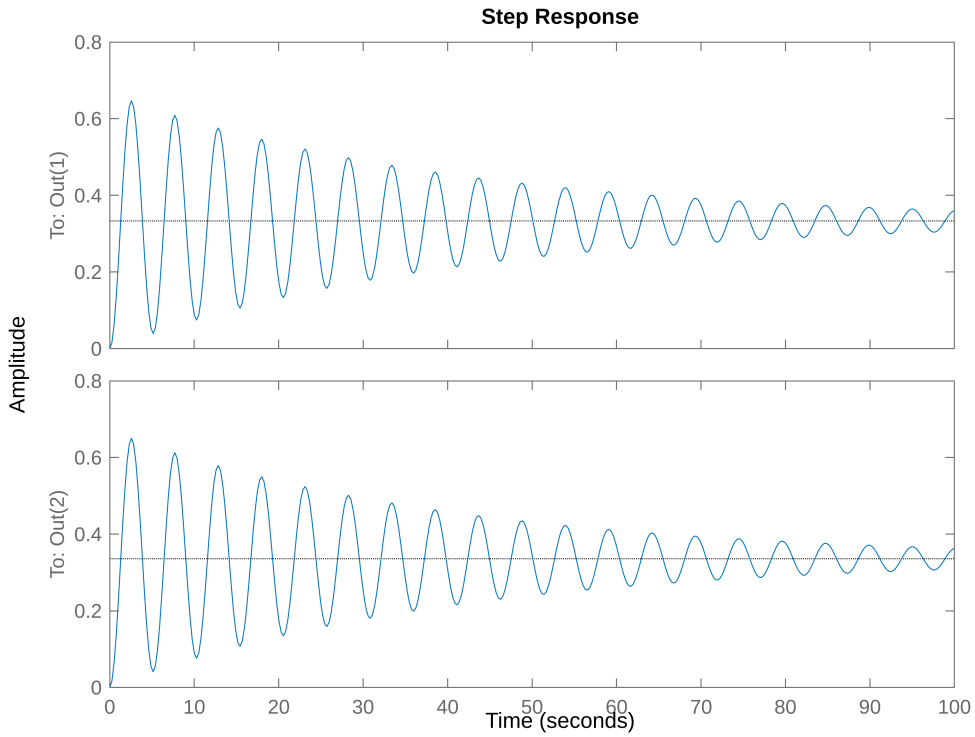
```
Csim=[1 0 0 0;0 0 1 0]
```

```
Csim = 2×4
     1     0     0     0
     0     0     1     0
```

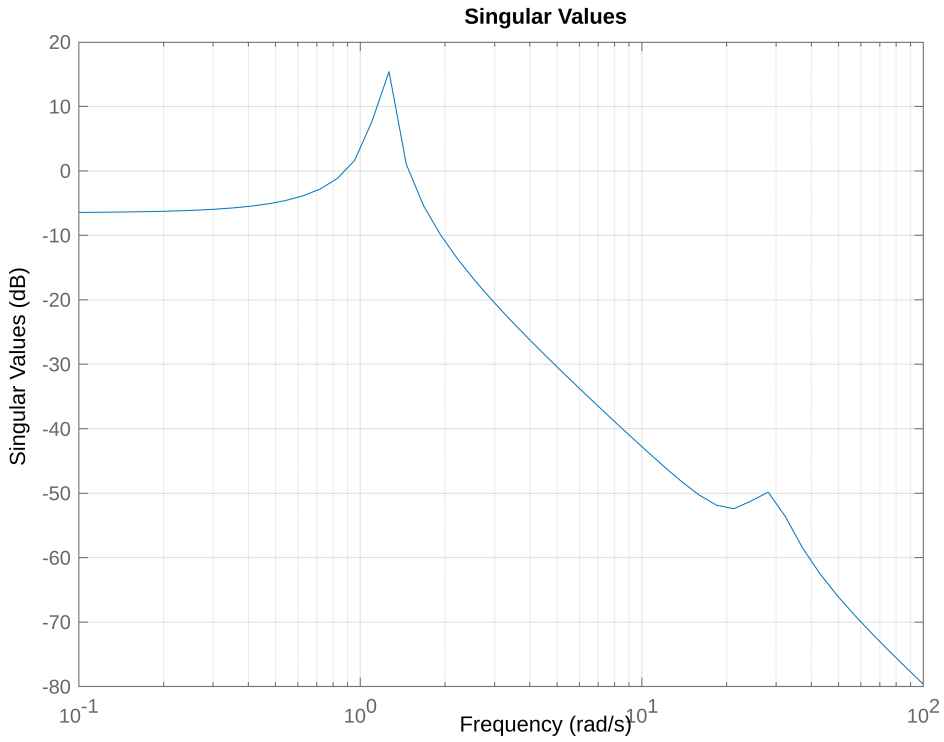
```
sys=ss(A,B,Csim,0);
tf(sys)
```

```
ans =

  From input to output...
                   4.025 s + 400.1
   1:  ---------------------------------------------
       s^4 + 8.15 s^3 + 803.6 s^2 + 52.09 s + 1200

                  s^2 + 4.125 s + 403.1
   2:  ---------------------------------------------
       s^4 + 8.15 s^3 + 803.6 s^2 + 52.09 s + 1200

Continuous-time transfer function.
```

```
step(sys,100)
```




```
sigma(sys,logspace(-1,2)), grid on
```



Controlabilidad

```
Contrm=ctrb(A,B)
```

```
Contrm = 4×4
10^3  ×
         0         0    0.0040    0.3673
         0    0.0040    0.3673   -6.2280
         0    0.0010   -0.0040   -0.3677
    0.0010   -0.0040   -0.3677    6.1792
```

```
Gramc=gram(sys,'c')
```

```
Gramc = 4×4
    1.6751    0.0000    1.6814   -0.0008
    0.0000    2.5233    0.0008    2.5020
    1.6814    0.0008    1.6877    0.0000
   -0.0008    2.5020    0.0000    2.5422
```

```
[V,Lam]=eig(Gramc); V
```

```
V = 4×4
    0.7084    0.0002    0.7058   -0.0001
    0.0001   -0.7084    0.0002    0.7058
   -0.7058    0.0002    0.7084    0.0001
    0.0001    0.7058   -0.0002    0.7084
```

```
sqrt(diag(Lam)')
```

```
ans = 1×4
    0.0062    0.1754    1.8338    2.2438
```

Probar distintos valores k12, f12.

Observabilidad

Vamos a discretizar, para también evaluar cómo influye la frecuencia con la que se toman las medidas. [Detalles teoría discretización en http://personales.upv.es/asala/videos/disc1.html y ejemplos de Matlab detallados en http://personales.upv.es/asala/videos/disc2.html]

```
Periodo=.3;
Csens=[1000 0 0 0];
sys2=ss(A,B,Csens,0);
sys2d=c2d(sys2,Periodo,'zoh');
Obsm=obsv(sys2d.A,sys2d.C)
```

```
Obsm = 4×4
10^3 ×
    1.0000         0         0         0
    0.4100    0.1495    0.5218    0.1411
    0.3549    0.2673    0.3886    0.2711
    0.2550    0.3550    0.2047    0.3563
```

```
Gramo=gram(sys2d,'o')
```

```
Gramo = 4×4
10^6 ×
    9.1418    0.4538    8.2072    0.4535
    0.4538    5.5426    0.0903    5.5628
    8.2072    0.0903    8.3149    0.0877
    0.4535    5.5628    0.0877    5.5831
```

```
[Vo,Lamo]=eig(Gramo);diag(Lamo)', Vo
```

```
ans = 1×4
10^7 ×
    0.0000    0.0499    1.1085    1.6998
Vo = 4×4
    0.0006    0.6895    0.0451   -0.7229
    0.7084   -0.0241   -0.7023   -0.0662
   -0.0009   -0.7235    0.0885   -0.6846
   -0.7058   -0.0227   -0.7049   -0.0662
```

Calidad de estimado de x0 en función de su efecto en medidas futuras:

```
sqrt(1./diag(Lamo)')
```

```
ans = 1×4
    0.1516    0.0014    0.0003    0.0002
```

Reducción de modelos

Escalado:

Queremos un modelo para predecir la posición de M1 y M2, con 10 veces más exactitud en M1 que en M2:

```
Cmr=[10 0 0 0;0 0 1 0];
sys3=ss(A,B,Cmr,0);
```

1. Reducción modal:

```
zpk(sys3)
```

```
ans =

  From input to output...
                    40.25 (s+99.4)
   1:  ---------------------------------------------
       (s^2 + 0.04984s + 1.497) (s^2 + 8.1s + 801.7)

                  (s^2 + 4.125s + 403.1)
   2:  ---------------------------------------------
       (s^2 + 0.04984s + 1.497) (s^2 + 8.1s + 801.7)

Continuous-time zero/pole/gain model.
```

```
dcgain(sys3)
```

```
ans = 2×1
    3.3333
    0.3358
```

Forma canónica modal:

```
syscanon=canon(sys3)
```

```
syscanon =

  A =
             x1        x2         x3        x4
   x1     -4.05     28.02          0         0
   x2    -28.02     -4.05          0         0
   x3         0         0  -0.02492     1.223
   x4         0         0    -1.223  -0.02492

  B =
             u1
   x1    0.7051
   x2   -0.1098
   x3  0.002012
   x4    -4.633

  C =
               x1          x2          x3          x4
   y1     0.03849       0.247     -0.8822  -0.0003828
   y2   -0.003893    -0.02459    -0.08855  -4.788e-05

  D =
       u1
   y1   0
   y2   0

Continuous-time state-space model.
```

Objetivo: modelo reducido que aproxime respuesta a frecuencias menores de 10 rad/s.

```
[g1,g2]=freqsep(syscanon,10);
```

```
g1
```

```
g1 =

  A =
             x1        x2
   x1  -0.02492     1.223
   x2    -1.223  -0.02492

  B =
              u1
   x1  -0.001006
   x2      2.316

  C =
              x1         x2
   y1      1.764  0.0007656
   y2     0.1771  9.576e-05

  D =
       u1
   y1   0
   y2   0

Continuous-time state-space model.
```

```
zpk(g1)
```

```
ans =

  From input to output...
       -1.7124e-06 (s-2.92e06)
   1:  -----------------------
       (s^2 + 0.04984s + 1.497)

        4.3633e-05 (s+1.15e04)
   2:  -----------------------
       (s^2 + 0.04984s + 1.497)

Continuous-time zero/pole/gain model.
```

```
dcgain(g1)
```

```
ans = 2×1
    3.3396
    0.3352
```

Veamos el resíduo (fracciones simples a altas frecuencias naturales)

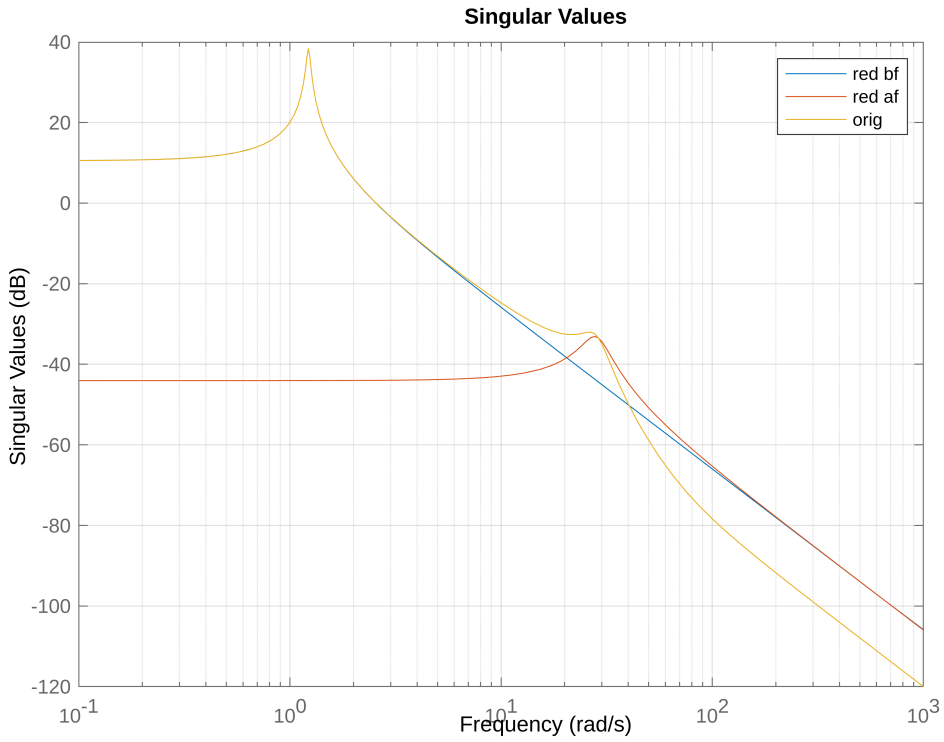
```
zpk(g2)
```

```
ans =

  From input to output...
       1.7124e-06 (s-2.92e06)
   1:  ----------------------
        (s^2 + 8.1s + 801.7)

       -4.3633e-05 (s-1.141e04)
   2:  -----------------------
         (s^2 + 8.1s + 801.7)

Continuous-time zero/pole/gain model.
```

```
dcgain(g2)
```

```
ans = 2×1
   -0.0062
    0.0006
```

```
sigma(g1,g2,sys3)
legend('red bf','red af','orig'), grid on
```



2. Reducción basada en equilibrado/valores singulares

Vamos a reducir la discretización a 80 milisegundos

```
sys3d=c2d(sys3,0.08,'zoh');
Gramo=gram(sys3d,'o'); Gramc=gram(sys3d,'c');
Qo=chol(Gramo);Qc=chol(Gramc);
[U,Sigma,V]=svd(Qo*Qc');
diag(Sigma)
```

```
ans = 4×1
   42.0715
   40.3934
    0.0121
    0.0075
```

Realización equilibrada (balanced)

```
T=sqrt(Sigma)*V'*inv(Qc'); %matriz de transformación
```

```
sysbal=ss2ss(sys3d,T); %cambio de coordenadas del estado xnew=T·xold
```

Los gramianos de controlabilidad y observabilidad son diagonales, iguales:

```
gram(sysbal,'o')
```

```
ans = 4×4
   42.0715   -0.0000   -0.0000   -0.0000
   -0.0000   40.3934   -0.0000   -0.0000
   -0.0000   -0.0000    0.0121   -0.0000
   -0.0000   -0.0000   -0.0000    0.0075
```

```
gram(sysbal,'c')
```

```
ans = 4×4
   42.0715    0.0000    0.0000    0.0000
    0.0000   40.3934   -0.0000   -0.0000
    0.0000   -0.0000    0.0121   -0.0000
    0.0000   -0.0000   -0.0000    0.0075
```

```
[sysbal2, Sigma2]=balreal(sys3d); %Matlab lo hace todo en una linea...
Sigma2'
```

```
ans = 1×4
   42.0715   40.3934    0.0121    0.0075
```

```
gram(sysbal2,'o')
```

```
ans = 4×4
   42.0715    0.0000    0.0000   -0.0000
    0.0000   40.3934   -0.0000    0.0000
    0.0000   -0.0000    0.0121    0.0000
   -0.0000    0.0000    0.0000    0.0075
```

```
gram(sysbal2,'c')
```

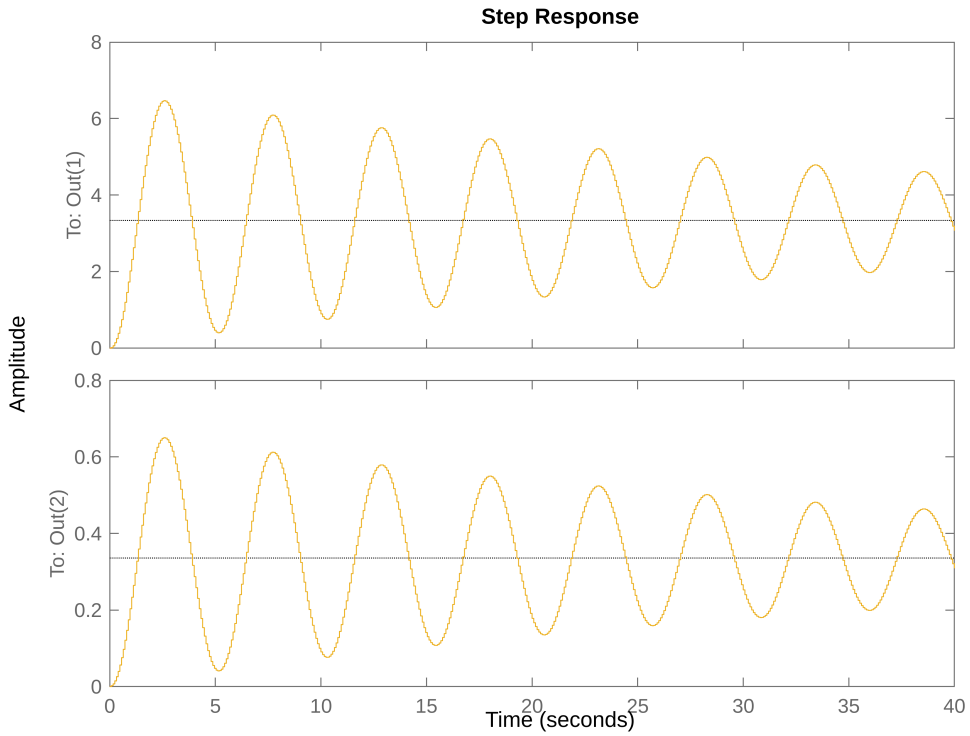
```
ans = 4×4
   42.0715   -0.0000   -0.0000    0.0000
   -0.0000   40.3934   -0.0000    0.0000
   -0.0000   -0.0000    0.0121    0.0000
    0.0000    0.0000    0.0000    0.0075
```

Dos de ellos son muy pequeños: reduzcamos el orden, eliminando los estados 3 y 4 de la realización, conservando ganancia:

```
sysred=modred(sysbal2,[3 4]);
```

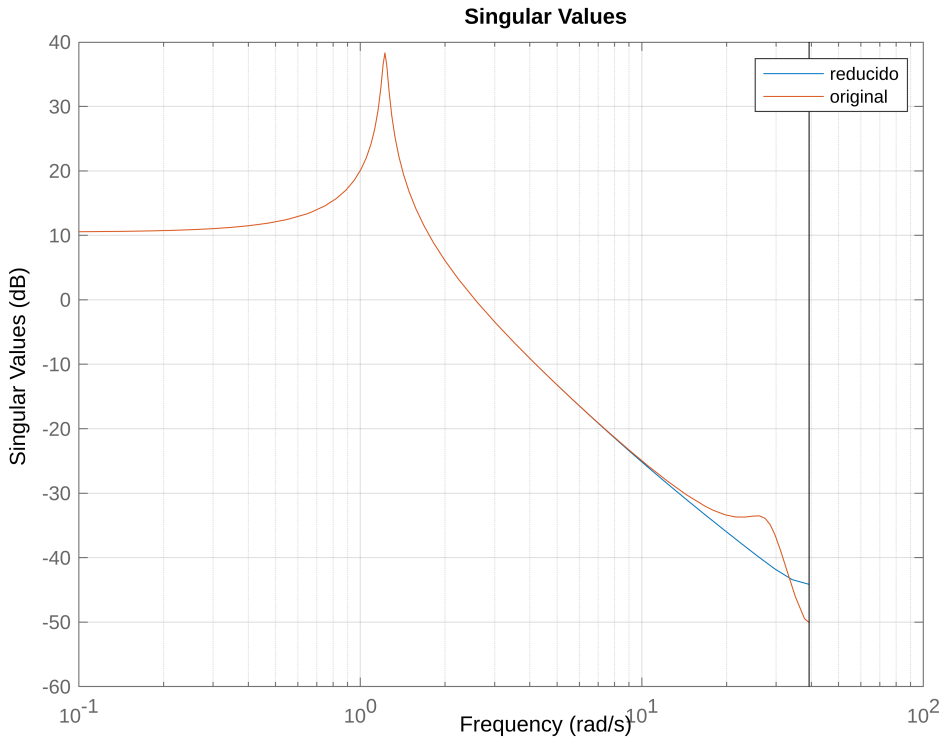
Las respuestas entrada-salida son todas prácticamente indisguibles:

```
step(sysbal2,sysred,sys3d,40)
```



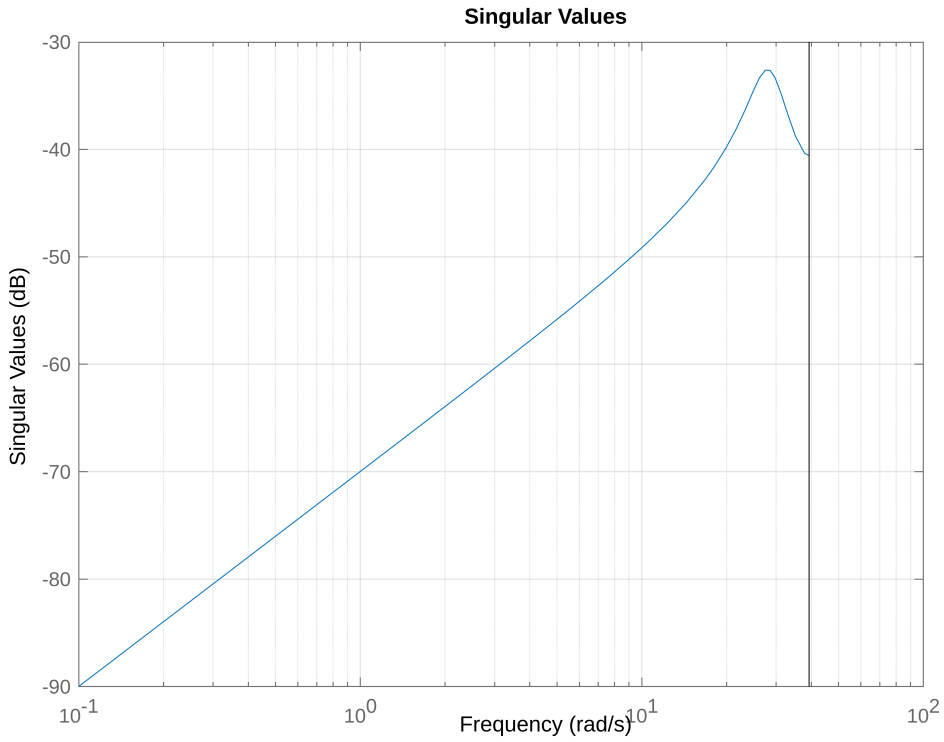
Comparemos el diagrama de amplitud (valores singulares) de respuesta en frecuencia

```
sigma(sysred,sys3d); grid on
legend('reducido','original')
```



Dibujemos el error de modelado,

```
sigma(sys3d-sysred); grid on
```



El pico de la respuesta en frecuencia del error es:

```
norm(sys3d-sysred,'inf')
```

```
ans = 0.0236
```

pasando a dB:

```
20*log10(ans)
```

```
ans = -32.5490
```

Matlab calcula la realización equilibrada y el modelo reducido en una sola linea:

```
sysred2=balred(sys3d,2);
```

Dan la misma tf reducida:

```
%original
zpk(sys3d)
```

```
ans =

  From input to output...
         0.0074349 (z+4.702) (z+0.6954) (z+0.07359)
   1:  ---------------------------------------------
       (z^2 - 1.986z + 0.996) (z^2 + 0.8994z + 0.5231)

       0.0024545 (z+0.819) (z^2 + 0.0007346z + 0.7405)
   2:  ---------------------------------------------
```

```
       (z^2 - 1.986z + 0.996) (z^2 + 0.8994z + 0.5231)

Sample time: 0.08 seconds
Discrete-time zero/pole/gain model.
```

```
%reducida hecho paso a paso por nosotros
zpk(sysred)
```

```
ans =

  From input to output...
       -0.0061746 (z-4.896) (z+0.324)
   1:  ------------------------------
           (z^2 - 1.986z + 0.996)

       0.00062084 (z^2 + 0.6004z + 3.568)
   2:  ----------------------------------
             (z^2 - 1.986z + 0.996)

Sample time: 0.08 seconds
Discrete-time zero/pole/gain model.
```

```
%reducida en 1 linea con balred de control syst. toolbox
zpk(sysred2)
```

```
ans =

  From input to output...
       -0.0061746 (z+0.324) (z-4.896)
   1:  ------------------------------
           (z^2 - 1.986z + 0.996)

       0.00062084 (z^2 + 0.6004z + 3.568)
   2:  ----------------------------------
             (z^2 - 1.986z + 0.996)

Sample time: 0.08 seconds
Discrete-time zero/pole/gain model.
```

Posición de los polos:

```
s=tf('s');
G=ss(1/(s+1)+(s+20)/(s^2+12*s+16^2));
Gbalred=balred(G,1);
eig(Gbalred)
```

```
ans = -1.0072
```

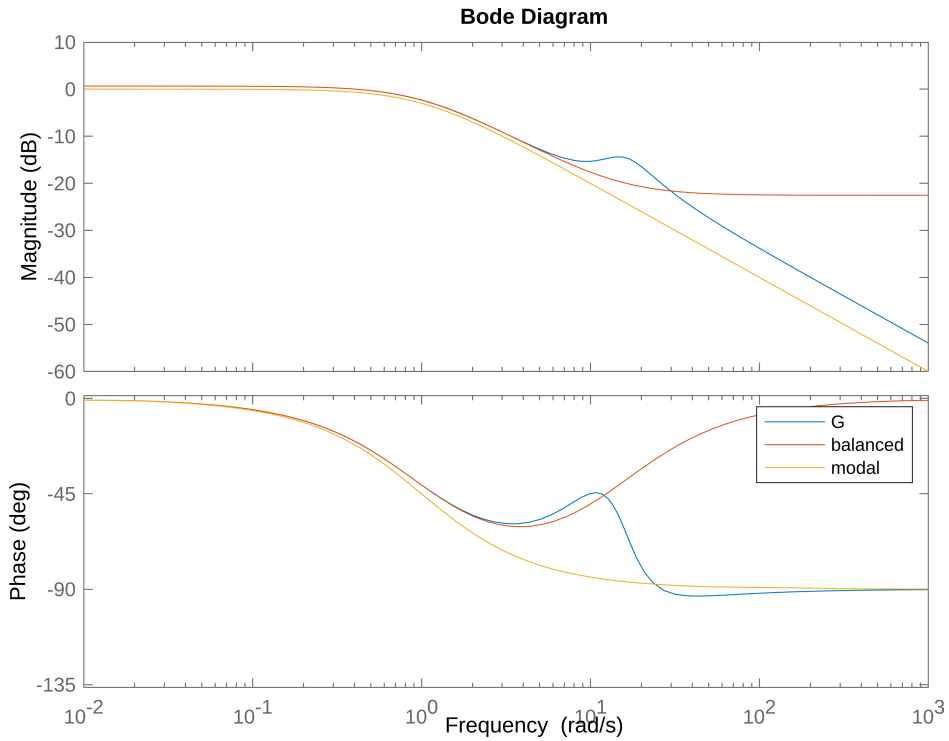
```
Gmodalr=freqsep(G,4); %4 rad/s
tf(Gmodalr)
```

```
ans =

    1
  -----
  s + 1

Continuous-time transfer function.
```

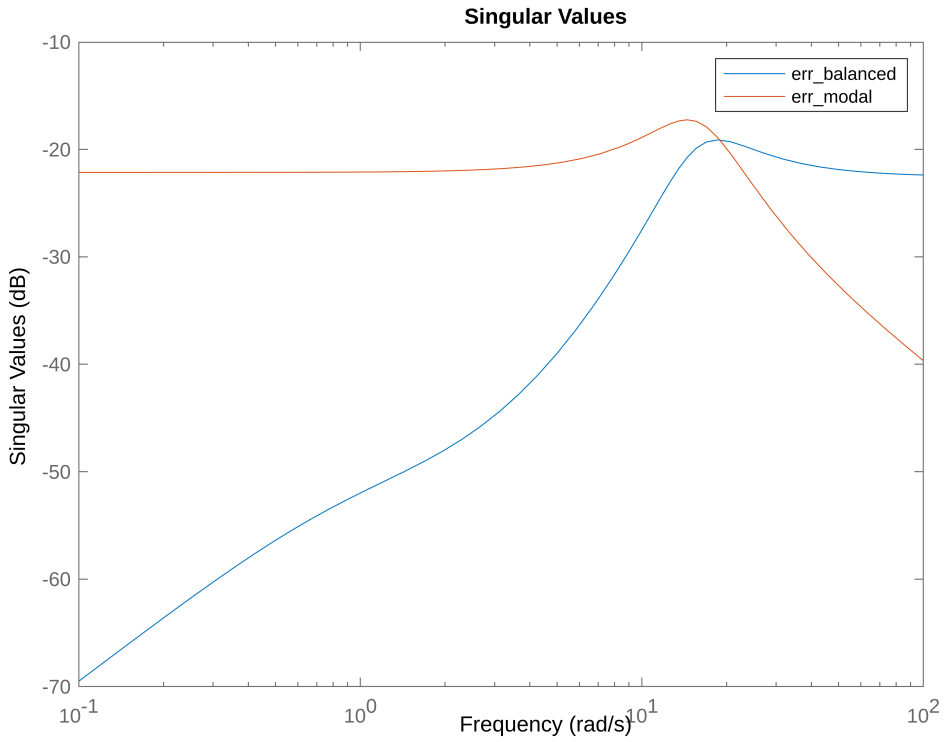
```
eig(Gmodalr)
```

```
ans = -1
```

```
bode(G,Gbalred,Gmodalr)
legend('G','balanced','modal')
```




```
sigma(G-Gbalred,G-Gmodalr)
legend('err_balanced','err_modal')
```


Singular Values
err_balanced
err_modal
Singular Values (dB)
-10
-20
-30
-40
-50
-60
-70
10^-1
10^0
10^1
10^2
Frequency (rad/s)