

Control con 2 grados de libertad (vs 1GL): enfoque $\mathcal{H}_\infty$

© 2019 Antonio Sala Piqueras, Universitat Politècnica de València. Todos los derechos reservados.

Este código ejecutó correctamente en Matlab R2018b

Vídeo-presentación en: <http://personales.upv.es/asala/YT/V/c2glhim.html>

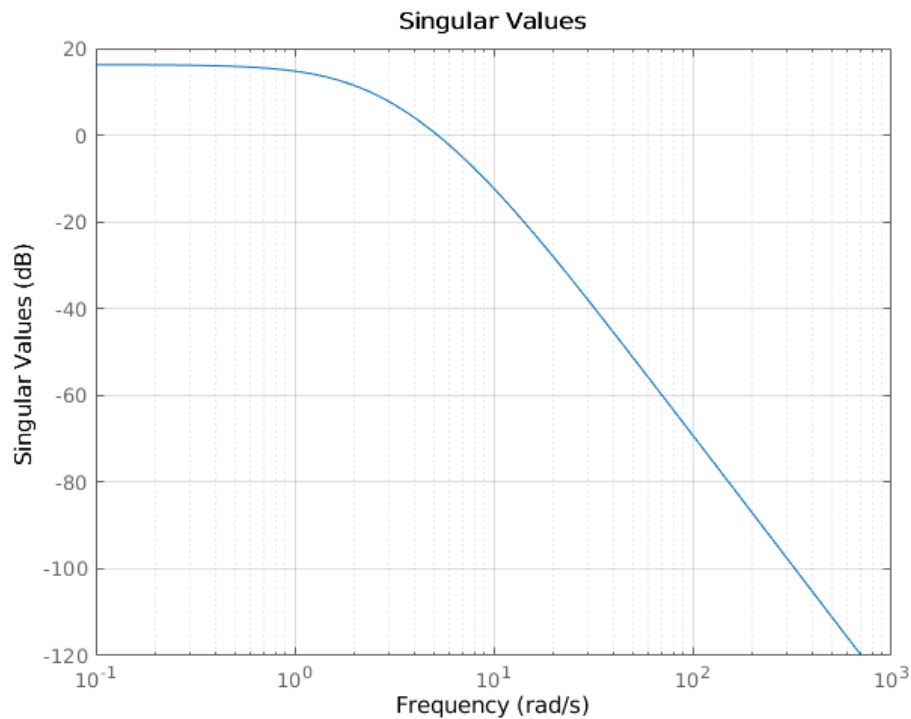
Objetivo: plantear una planta generalizada con control de dos grados de libertad para seguir referencias+perturbaciones, y comparar el resultado del control $\mathcal{H}_\infty$ obtenido con el problema de 1 grado de libertad con los mismos pesos.

Tabla de Contenidos

Modelo.....	1
Planta Generalizada No Ponderada.....	2
Planta Generalizada Ponderada y cálculo de control.....	2
Evaluación de la solución obtenida.....	5

Modelo

```
muchasfigs=1;
s=tf('s');
G=6.5*(2*3*9/(s-2)/(s+3)/(s+9));
if(muchasfigs)
    sigma(G), grid on
end
```



Planta Generalizada No Ponderada

```
PlantaGen=minreal(ss([-1 0 1 -G; ...
    0 0 0 1; ...
    -1 -1 1 -G; ...
    0 0 1 0])); %si es posible, evitar escribir G 2 veces (usar repr. interna)
PlantaGen.InputName={'d','v','r','u'};
PlantaGen.OutputName={'e','u_{copy}','err_{med}','r_{copy}'};
size(PlantaGen)
```

State-space model with 4 outputs, 4 inputs, and 3 states.

```
PlantaGen1GL=PlantaGen(1:3,:);
PlantaGen1GL.OutputName'
```

```
ans = 1x3 cell array
    {'e'}      {'u_{copy}'}      {'err_{med}'}
```

```
size(PlantaGen1GL)
```

State-space model with 3 outputs, 4 inputs, and 3 states.

Planta Generalizada Ponderada y cálculo de control

```
%parametros de diseño
numerosimulacion=3;
switch numerosimulacion
    case 1
        if(isstable(G)) %caso estable
```

```

    Amplitud_ref=1 %seguir referencias 1 lo más rápido que se pueda...
    amplitudpertbajafrec=1.1
    anchobandaerror=1.95 %para 2GL gam=1 2.08, para 1GL: 1.95
else %caso inestable
    Amplitud_ref=1 %seguir referencias 1 lo más rápido que se pueda...
    amplitudpertbajafrec=1.1
    anchobandaerror=1.15 %para 2GL gam=1 1.15, para 1GL: Imposible!
end
case 2
    Amplitud_ref=1 %seguir referencias 1 lo más rápido que se pueda...
    amplitudpertbajafrec=0.001
    anchobandaerror=2.5 %para 2GL gam=1 2.5, para 1GL: 0.11
case 3
    Amplitud_ref=0.001 %seguir referencias 1 lo más rápido que se pueda...
    amplitudpertbajafrec=0.8
    anchobandaerror=4.3 %los dos igual, obviamente.
end
end

```

```

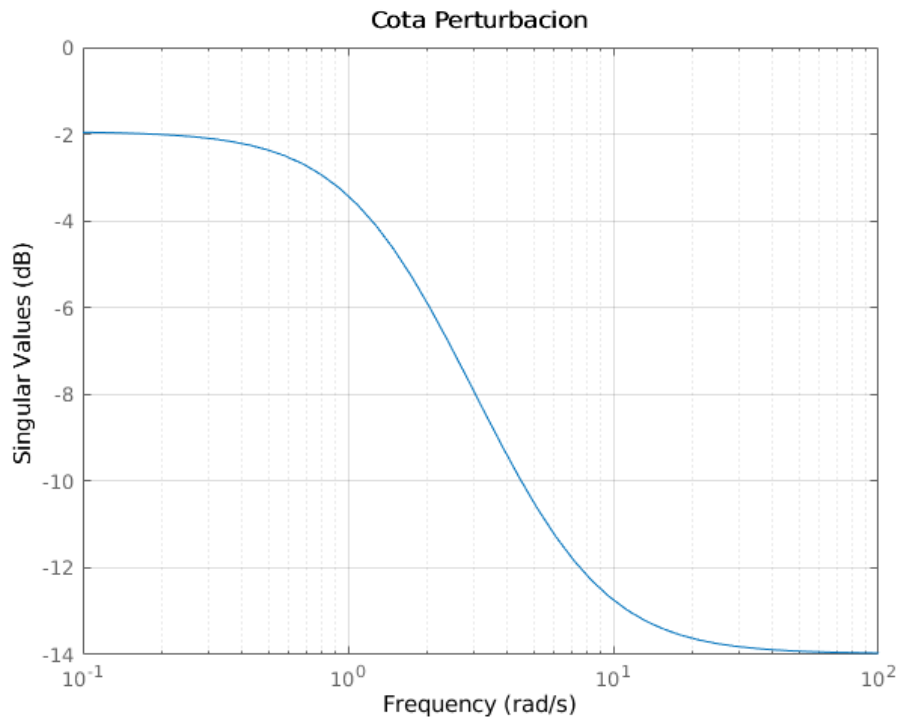
Amplitud_ref = 1.0000e-03
amplitudpertbajafrec = 0.8000
anchobandaerror = 4.3000

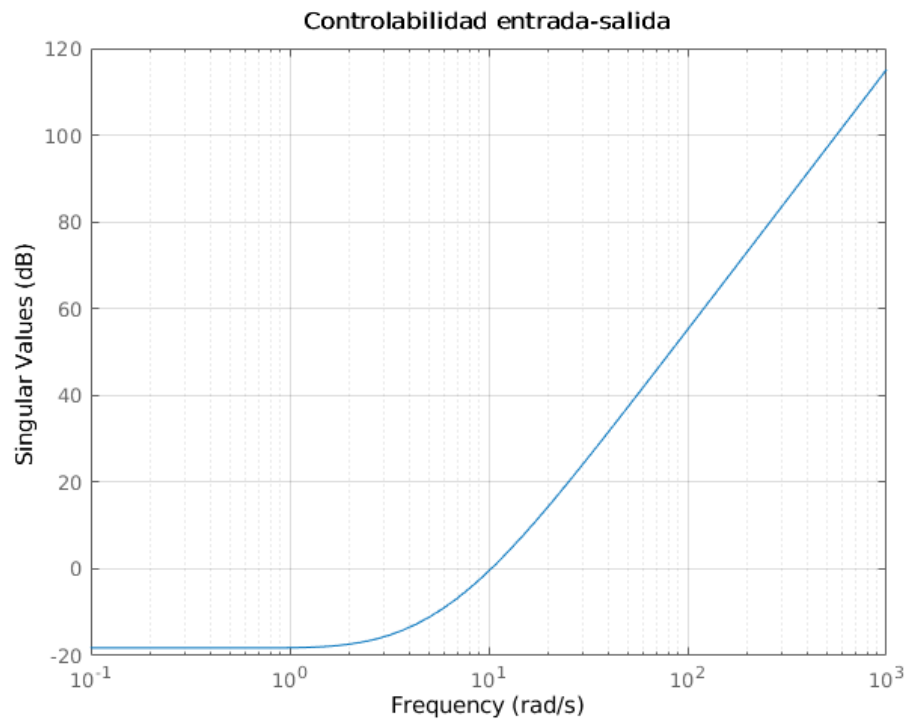
```

```

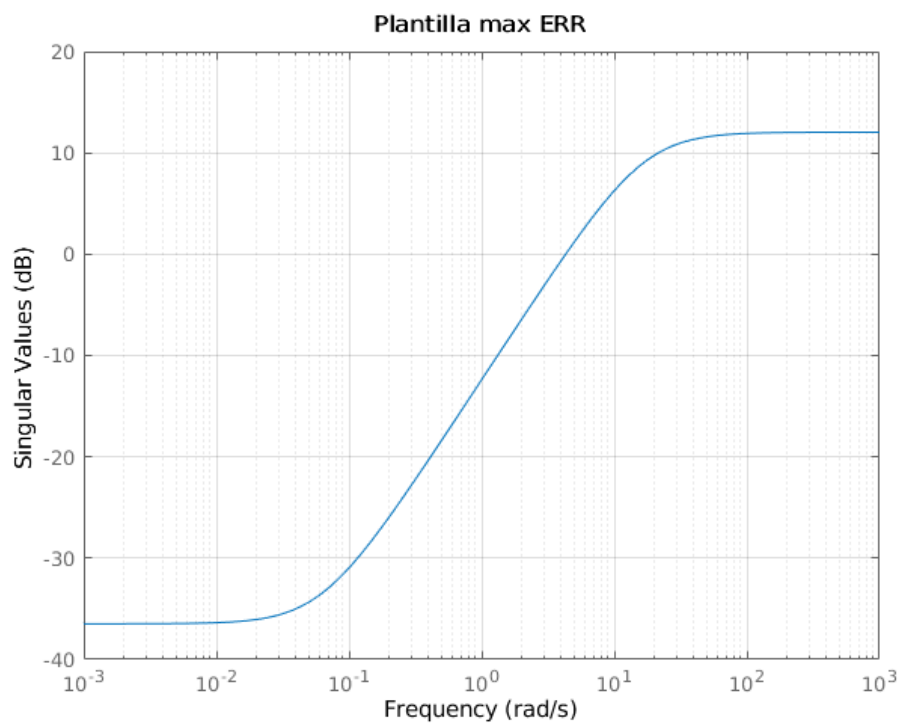
Wd=amplitudpertbajafrec*(1/(1.5*4)*s+1)/((1/1.5)*s+1); %filtro perturbaciones (entrada)
if(muchasfigs)
    sigma(Wd), grid on, title('Cota Perturbacion')
    figure
    sigma(inv(G)*[Wd Amplitud_ref]),grid on, title('Controlabilidad entrada-salida')
end

```





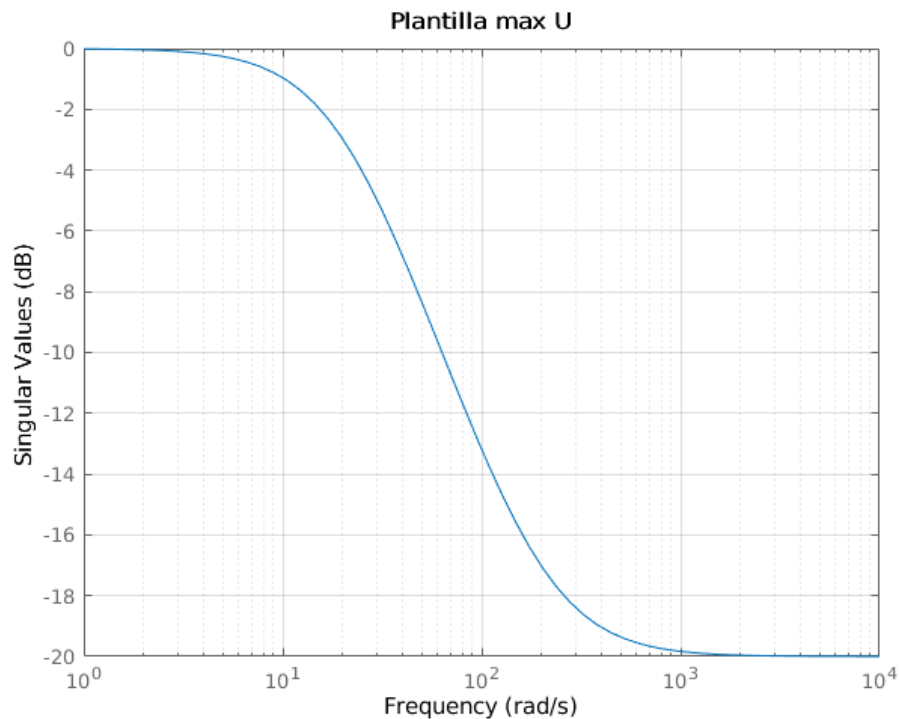
```
W_ruidomedida=0.01;
Plantilla_err=makeweight(0.015,anchobandaerror,4);
if(muchasfigs)
    sigma(Plantilla_err), grid on, title('Plantilla max ERR')
end
```



```

Werr=1/Plantilla_err;
anchobanda_act=20;
Plantilla_u=(1/(anchobanda_act*10)*s+1)/(1/anchobanda_act*s+1);
if(muchasfigs)
    sigma(Plantilla_u), grid on,title('Plantilla max U')
end

```



```

Wu=1/Plantilla_u;
W_ref=Amplitud_ref;%(10*s+1)
PesoEntrada=blkdiag(Wd,W_ruidomedida,W_ref,1);
PesoSalida=blkdiag(Werr,Wu,1,1);
PlantaGenPonderada=minreal(ss(PesoSalida*PlantaGen*PesoEntrada));
PlantaGenPonderada1GL=minreal(ss(PesoSalida(1:3,1:3)*PlantaGen1GL*PesoEntrada));
[K,CL,GAM,~]=hinfsvn(PlantaGenPonderada,2,1,hinfsvnOptions('RelTol',1e-3));
GAM

```

GAM = 0.9942

```

[K1GL,CL1GL,GAM1GL,~]=hinfsvn(PlantaGenPonderada1GL,1,1,hinfsvnOptions('RelTol',1e-3));
GAM1GL

```

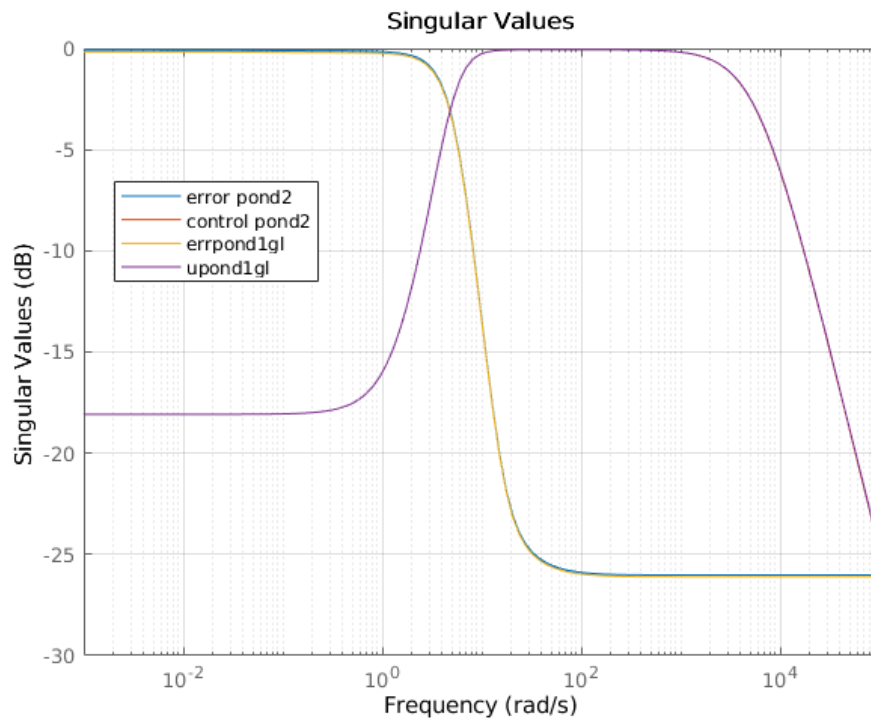
GAM1GL = 0.9942

Evaluación de la solución obtenida

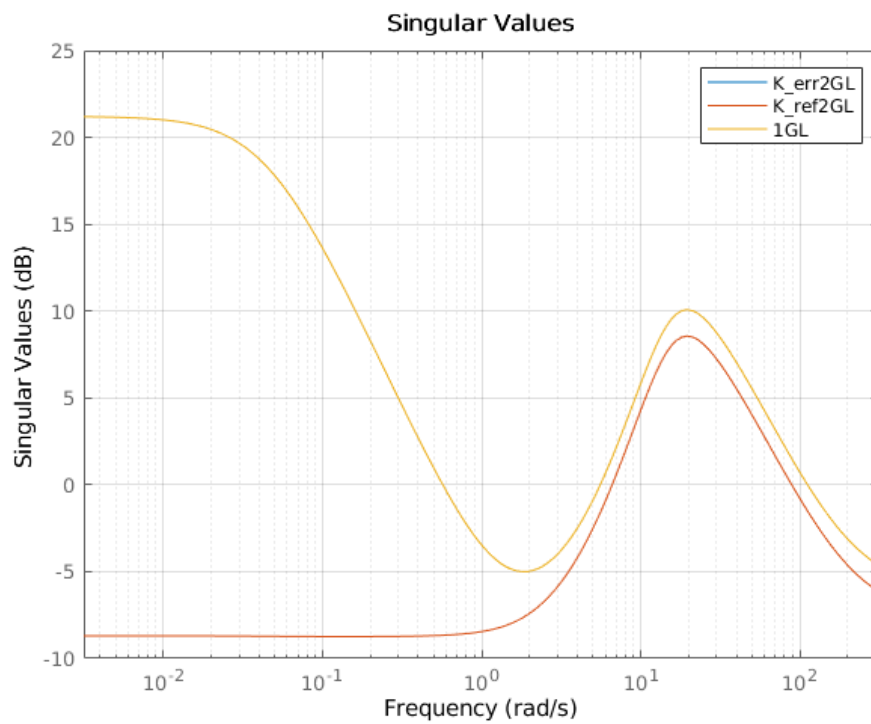
```

sigma(CL(1,:),CL(2,:),CL1GL(1,:)*0.99,CL1GL(2,:)), grid on, legend('error pond2','contr

```



```
sigma(K(:,1),K(:,2),K1GL,logspace(-2.5,2.5,150)), grid on, legend('K_err2GL','K_ref2GL')
```



```
CL_NoPesos=minreal(lft(PlantaGen,K));
```

2 states removed.

```
CL_NoPesos1GL=minreal(lft(PlantaGen1GL,K1GL));
```

2 states removed.

```
size(CL_NoPesos)
```

State-space model with 2 outputs, 3 inputs, and 7 states.

```
CL_NoPesos.InputName'
```

```
ans = 1x3 cell array
      {'d'}      {'v'}      {'r'}
```

```
CL_NoPesos.OutputName'
```

```
ans = 1x2 cell array
      {'e'}      {'u_{copy}'}
```

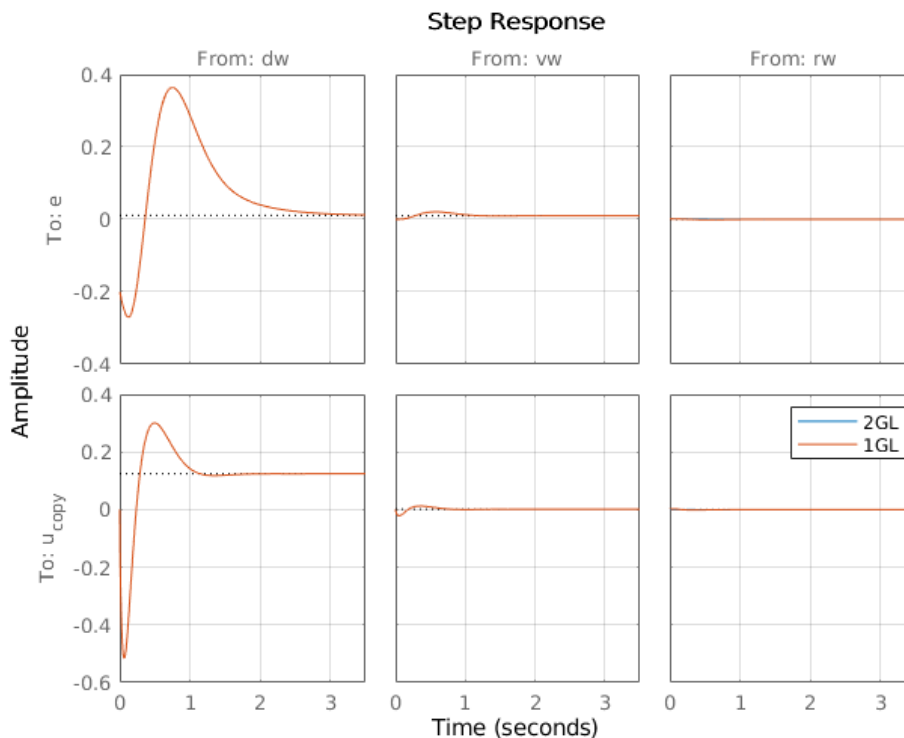
```
%step(CL_NoPesos,CL_NoPesos1GL), grid on,legend('2GL','1GL')
CL_SoloPesosEntrada=minreal(lft(PlantaGen*PesoEntrada,K));
```

2 states removed.

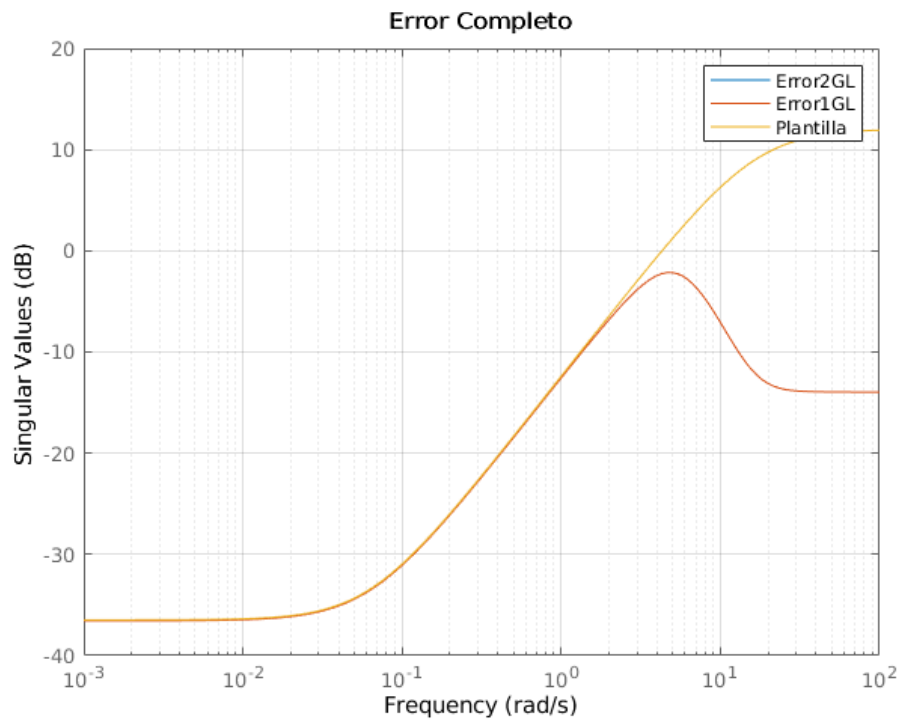
```
CL_SoloPesosEntrada1GL=minreal(lft(PlantaGen1GL*PesoEntrada,K1GL));
```

2 states removed.

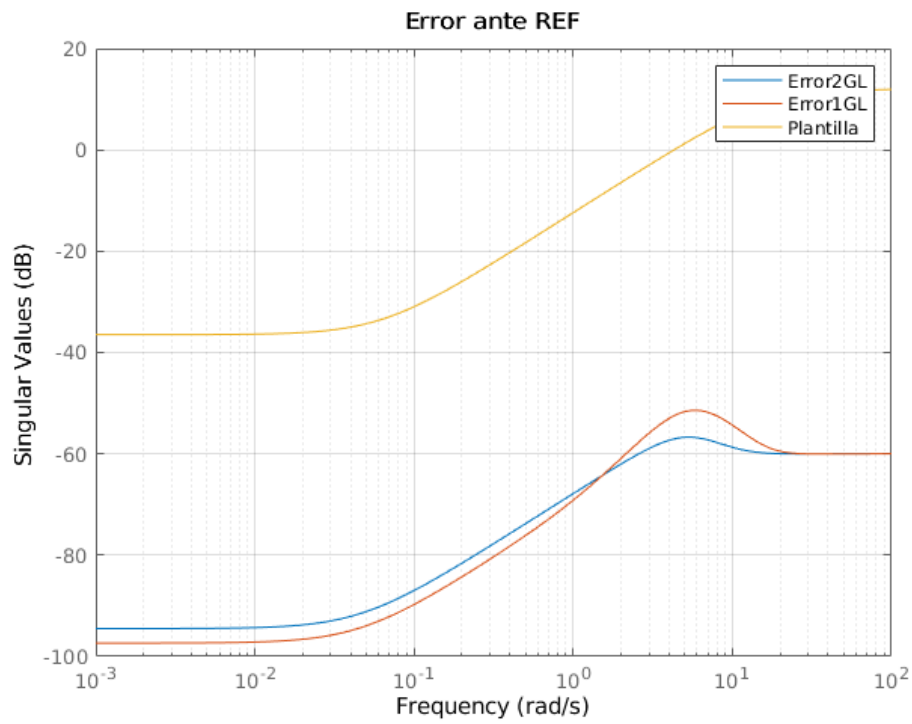
```
CL_SoloPesosEntrada1GL.InputName={'dw','vw','rw'};
step(CL_SoloPesosEntrada,CL_SoloPesosEntrada1GL), grid on, legend('2GL','1GL')
```



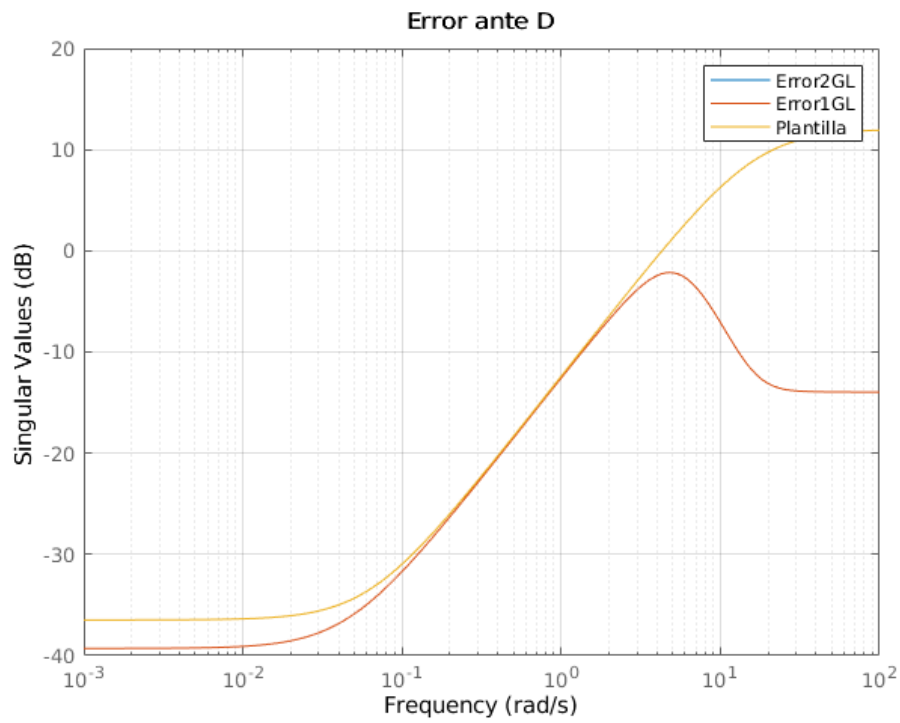
```
sigma(CL_SoloPesosEntrada(1,:),CL_SoloPesosEntrada1GL(1,:),Plantilla_err,logspace(-3,2,
```



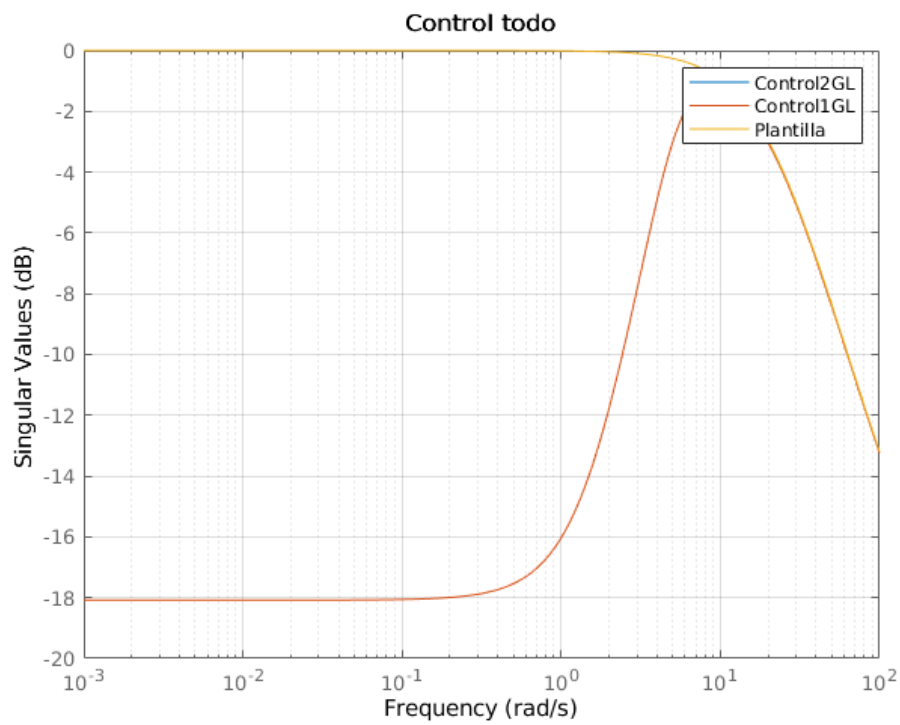
```
sigma(CL_SoloPesosEntrada(1,3),CL_SoloPesosEntrada1GL(1,3),Plantilla_err,logspace(-3,2,
```



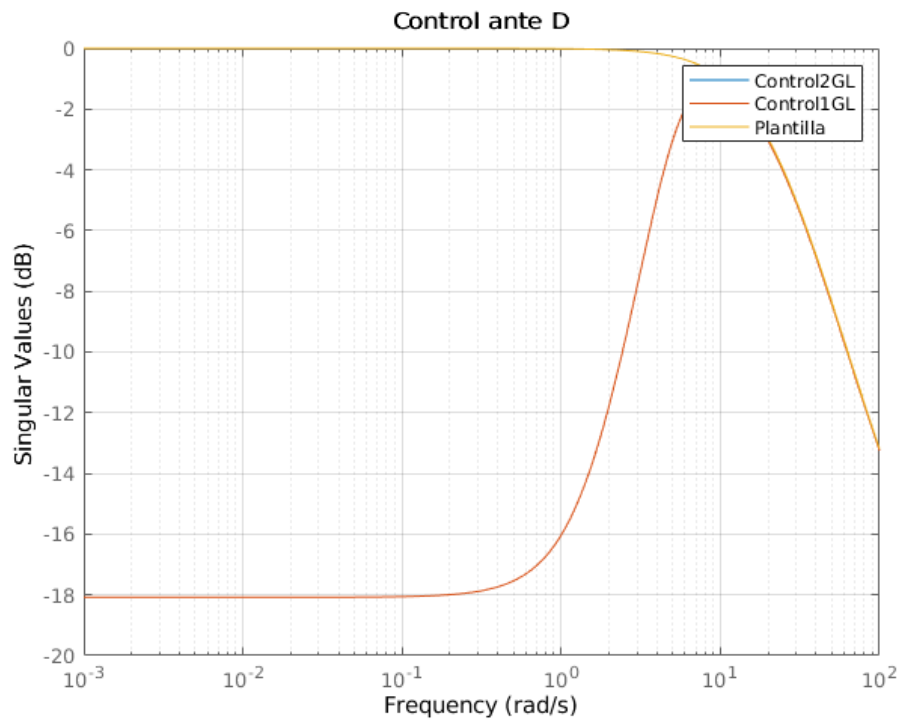
```
sigma(CL_SoloPesosEntrada(1,1),CL_SoloPesosEntrada1GL(1,1),Plantilla_err,logspace(-3,2,
```



```
sigma(CL_SoloPesosEntrada(2,:),CL_SoloPesosEntrada1GL(2,:),Plantilla_u,logspace(-3,2,15))
```



```
sigma(CL_SoloPesosEntrada(2,1),CL_SoloPesosEntrada1GL(2,1),Plantilla_u,logspace(-3,2,15))
```



```
sigma(CL_SoloPesosEntrada(2,3),CL_SoloPesosEntrada1GL(2,3),Plantilla_u,logspace(-3,2,15))
```

