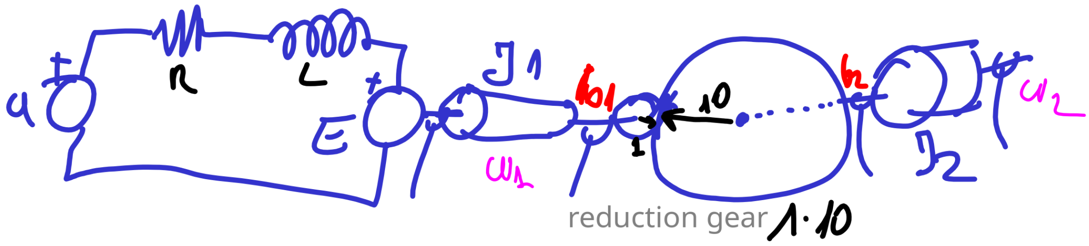
Control velocidad de motor DC via $\mathscr{H}_\infty$

4) Control bucle cerrado $\mathscr{H}_\infty$ (hinfsyn) [seguimiento de referencias]



Este código ejecutó en Matlab **`R2024b`**.

Presentación en vídeo:

https://personales.upv.es/asala/YT/V/mothin4.html

Tabla de Contenidos

1. Modelado

```
R=.3; L=20e-3; %electric circuit parameters
k=0.36; %Motor constant
J1=.45; b1=0.13; %motor (fast) side mechanical parameters
N=10; %gear ratio
J2=5; b2=0.6; %slow side mechanical parameters
```

El momento de inercia "equivalente" en el eje del motor es:

```
J=(J1+J2/N^2);
b=b1+b2/N^2; %Equivalent inertia and friction coefficients on motor
(fast) side
```

Física: $L\dfrac{dI}{dt} = U - R\cdot I - k\cdot\omega\,,\qquad J\dfrac{d\omega}{dt} = k\cdot I - b\cdot\omega + T_{carga}$

Ecuación de estado (matricial): $\dot{x} = Ax + Bu$

$$\frac{d}{dt}\begin{pmatrix} I \\ \omega \end{pmatrix} = \begin{pmatrix} \frac{-R}{L} & \frac{-K}{L} \\ \frac{k}{J} & \frac{-b}{J} \end{pmatrix}\begin{pmatrix} I \\ \omega \end{pmatrix} + \begin{pmatrix} \frac{1}{L} & 0 \\ 0 & \frac{1}{J} \end{pmatrix}\begin{pmatrix} U \\ T_{\text{carga}} \end{pmatrix}$$

```
A=[-R/L -k/L; +k/J -b/J];   B=[1/L 0; 0 1/J];
```

Ecuación de salida: $y = Cx + Du$; escogemos corriente y velocidad de giro.

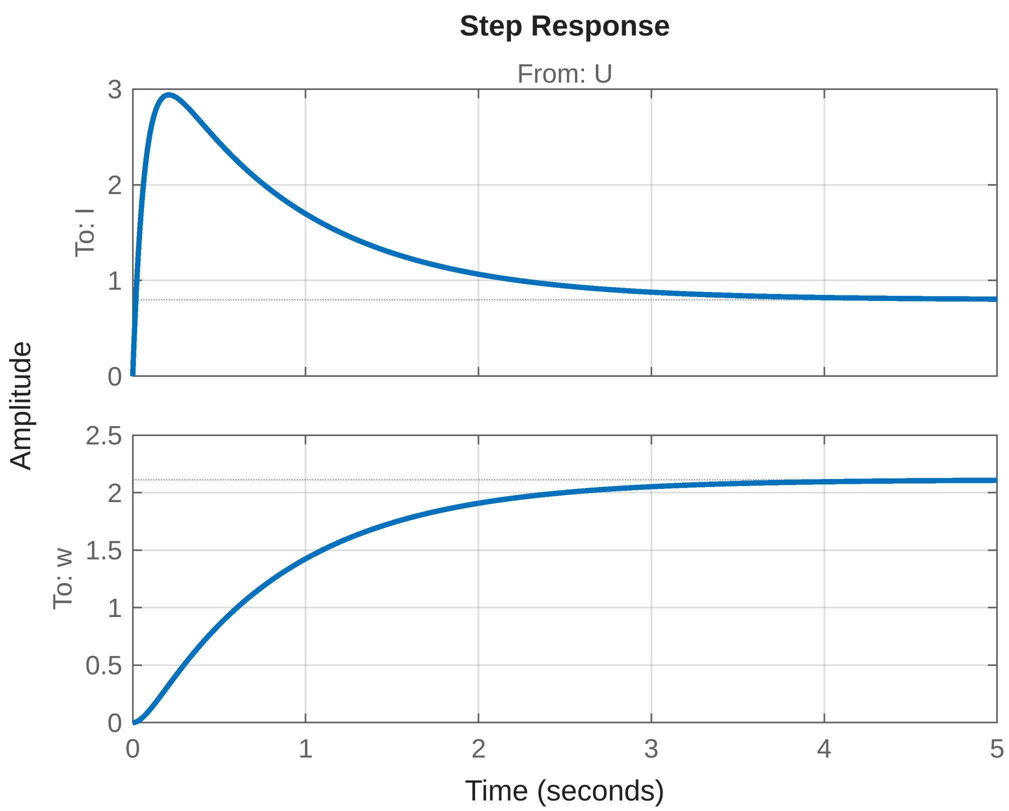
```
C=[1 0;  0 1]; D=zeros(2,2);
sysFull=ss(A,B,C,D); %state-space object
sysFull.OutputName={'I','w'};
sysFull.InputName={'U','Tload'};
```

*Por el momento, no consideramos perturbación de carga (sólo "seguimiento de referencia").

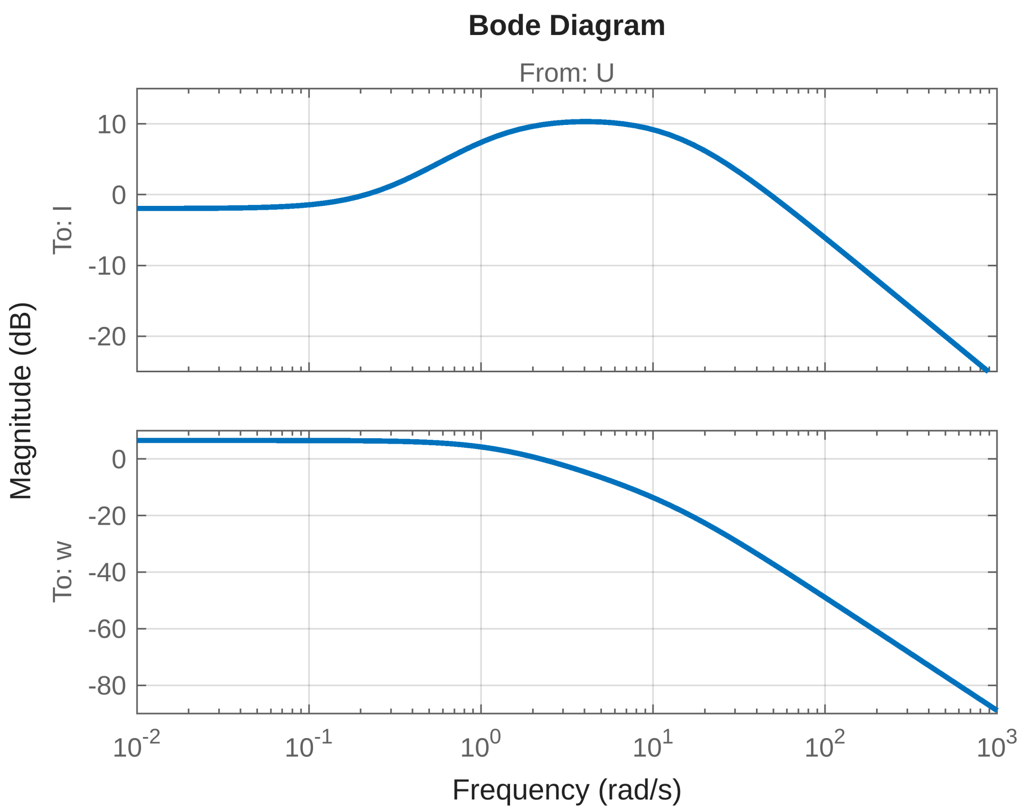
```
G=sysFull(:,1);
%zpk(G) %factorized fractions (zeros, poles, gains)
```

Respuesta temporal y en frecuencia en bucle abierto

```
step(G), grid on; h=gca; h.Responses(1).LineWidth=2;
```




```
bodemag(G), grid on; h=gca; h.Responses(1).LineWidth=2;
```



Análisis en equilibrio (CC)

```
DCG=dcgain(G);
gainUtoI=DCG(1) %Amperios por Voltio
```

```
gainUtoI =
0.7981
```

```
gainUtoW=DCG(2) %rad/s por Voltio
```

```
gainUtoW =
2.1127
```

2. Especificaciones

Límites de tensión y corriente

```
MaxI=75; %A
MaxU=120; %V
```

Amplitud de incrementos de referencia deseados

Mantenerse por debajo de MaxI, MaxU en el transitorio ante referencia máxima

```
Wref=90;
```

*Bueno, $\mathscr{H}_\infty$ se referirá a límites "en frecuencia", no en "tiempo", como discutiremos enseguida.

El control bucle abierto "naive" 1/ganancia, no funciona

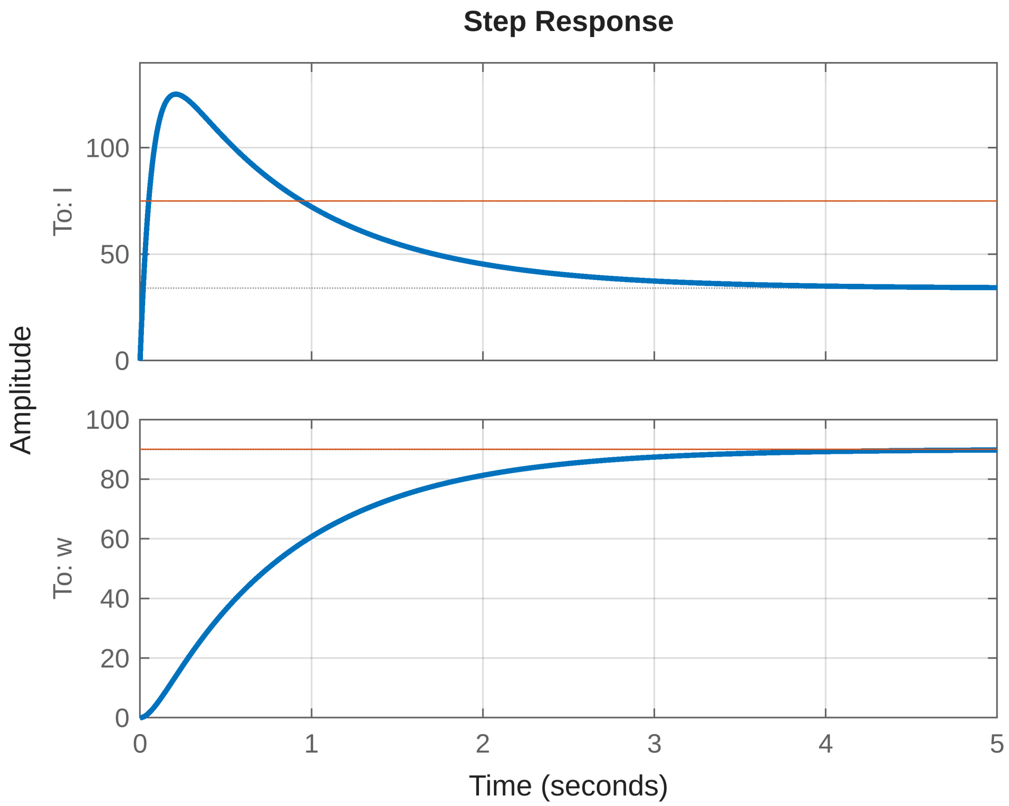
```
K_OL_gain=1/dcgain(G(2))
```

```
K_OL_gain =
0.4733
```

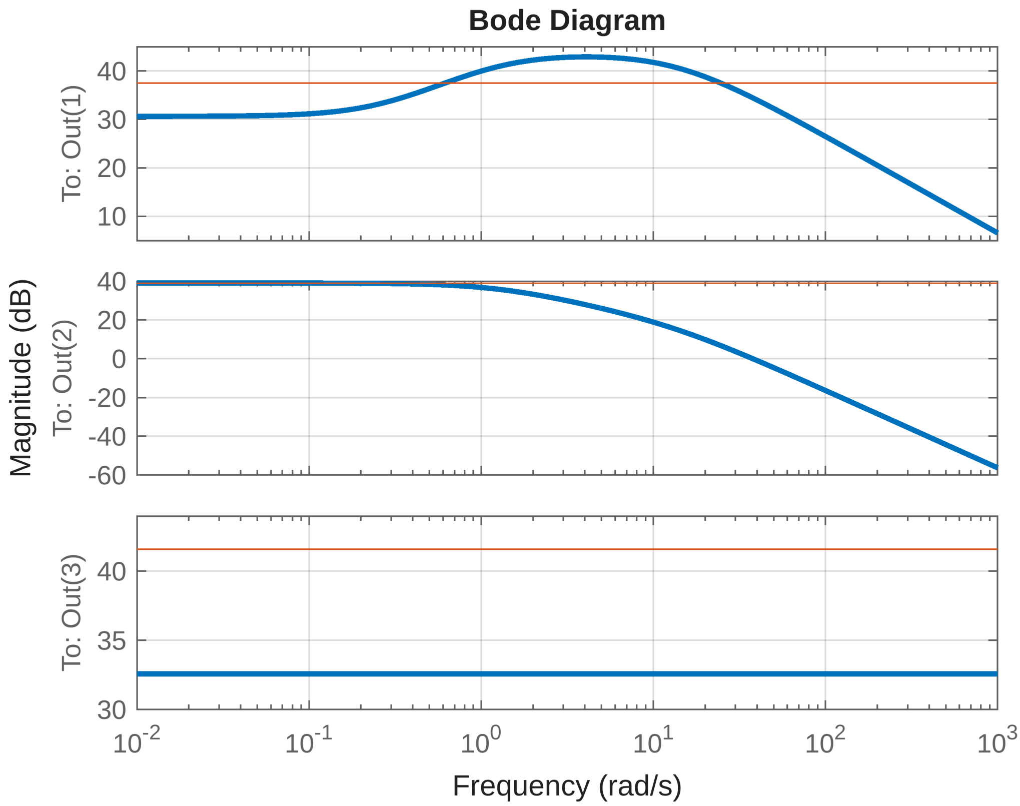
```
errScaled_naive=1-G(2)*K_OL_gain;
```

No funciona, pico de intensidad no admisible:

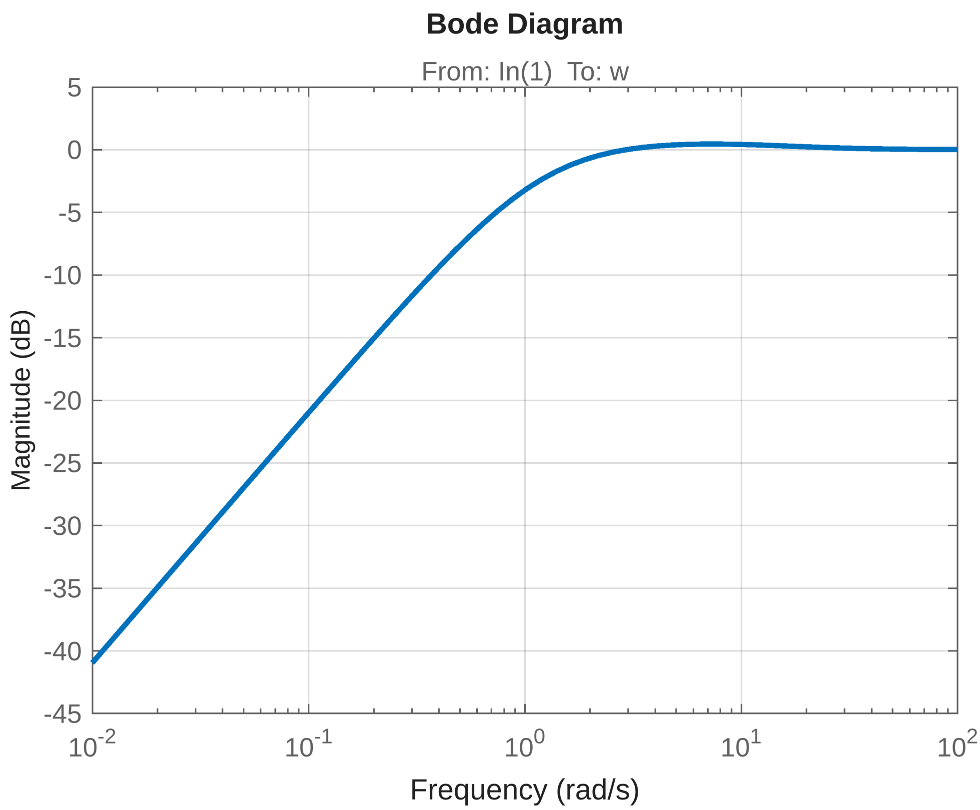
```
step(G*K_OL_gain*Wref,tf([MaxI;Wref])), grid on
h=gca;h.Responses(1).LineWidth=2;
```




```
bodemag([G;1]*K_OL_gain*Wref,tf([MaxI;Wref;MaxU])), grid on
h=gca;h.Responses(1).LineWidth=2;
```




```
bodemag(errScaled_naive), grid on
h=gca;h.Responses(1).LineWidth=2;
```



Control óptimo H-infinito

El escalado lo haremos directamente con pesos en planta generalizada ponderada.

```
%GP=[[0;-1] G;0 1;1 0]; %Open loop
%GP.InputName={'ref','U'};
%GP.OutputName={'I','err','U','r'};
GP=minreal(ss([[0;-1] G;0 1;1 -G(2)])); %closed loop
```

```
2 states removed.
```

```
GP.InputName={'ref','U'};
GP.OutputName={'I','err','U','err_copy'};
```

Nota: la línea 34 que define GP en bucle cerrado tiene "multi-incidencia" de G, porque sale dos veces; ello duplica estados y necesita "minreal"; en sistemas inestables o sobre todo inciertos (mu-síntesis) ello puede dar problemas numéricos, por ello se recomienda codificarlo SIN multiincidencia (no mencionado en el vídeo por brevedad y sencillez). Esto sería mejor:

```
GP2=[ [0;-1;0;1] [1 0;0 1;0 0;0 -1]*G+[0;0;1;0] ];
size(GP2)
```

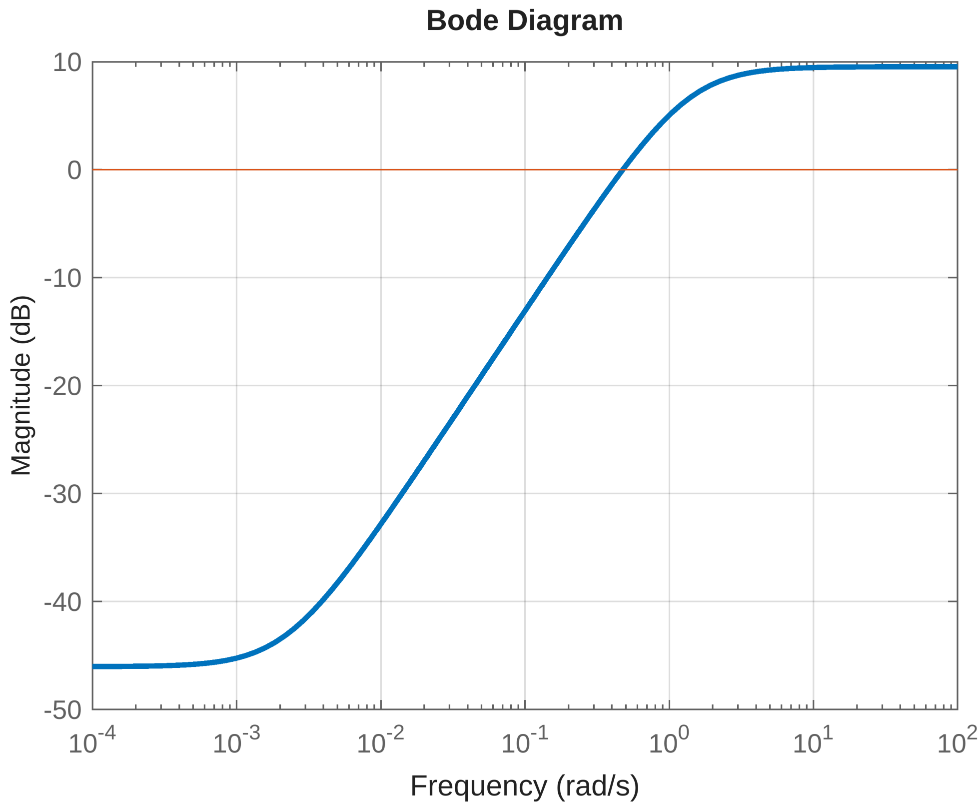
```
State-space model with 4 outputs, 2 inputs, and 2 states.
```

```
norm(GP2-GP) %son el mismo, pero sin necesitar hacer "minreal" en GP2.
```

```
ans =
5.1045e-15
```

```
Win1=Wref;

bw=0.476; %ancho de banda, bandwidth
PlantillaMaxErr=makeweight(0.005,bw,3); %Para ref UNITARIA
bodemag(PlantillaMaxErr,tf(1)), grid on
h=gca;h.Responses(1).LineWidth=2;
```




```
Wout1=minreal(blkdiag(1/MaxI, 1/(PlantillaMaxErr*Win1), 1/MaxU));
```

```
1 state removed.
```

```
GPWeigthed=minreal(blkdiag(Wout1,1)*GP*blkdiag(Win1,1));
[Khinf,CL,GAM]=hinfsyn(GPWeigthed,1,1); GAM
```

```
GAM =
0.9996
```

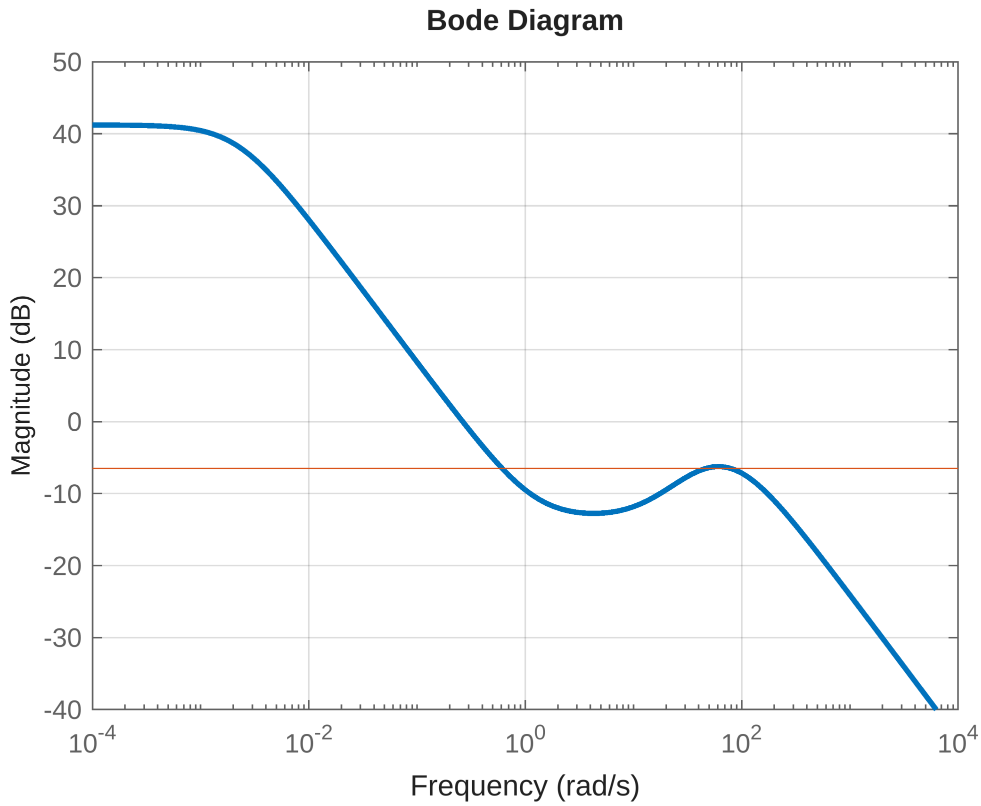
```
zpk(Khinf)
```

```
ans =

     62.774 (s+14.06) (s+1.212)
  --------------------------------
  (s+0.002244) (s+51.64) (s+80.18)

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

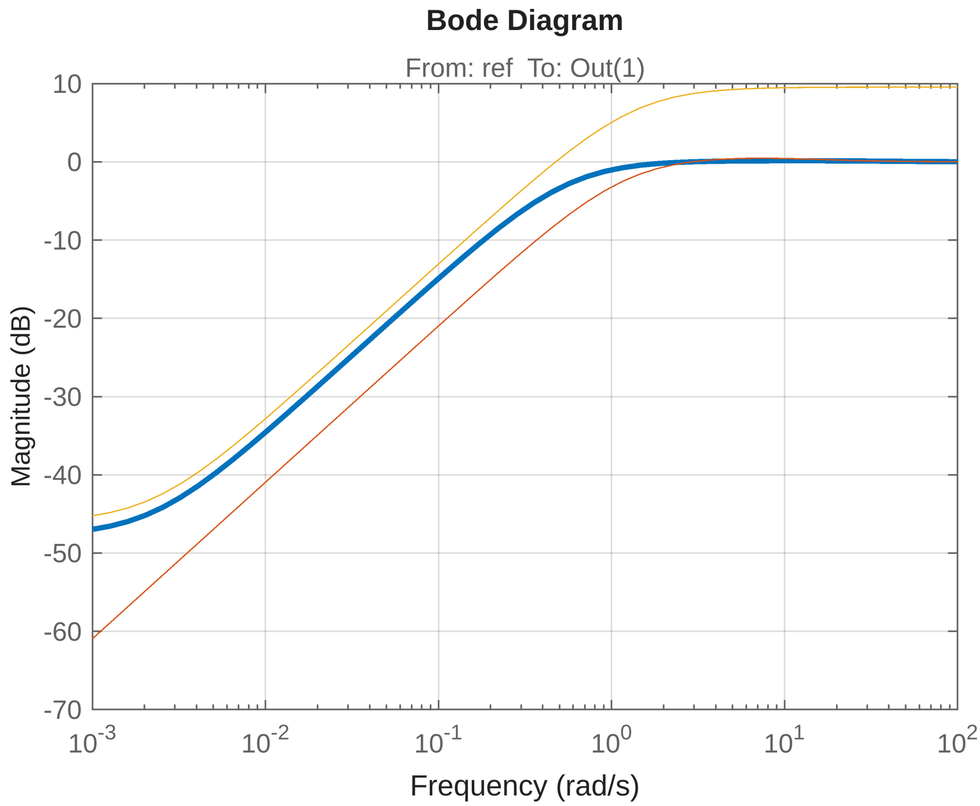
```
bodemag(Khinf,tf(K_OL_gain)), grid on
h=gca;h.Responses(1).LineWidth=2;
```



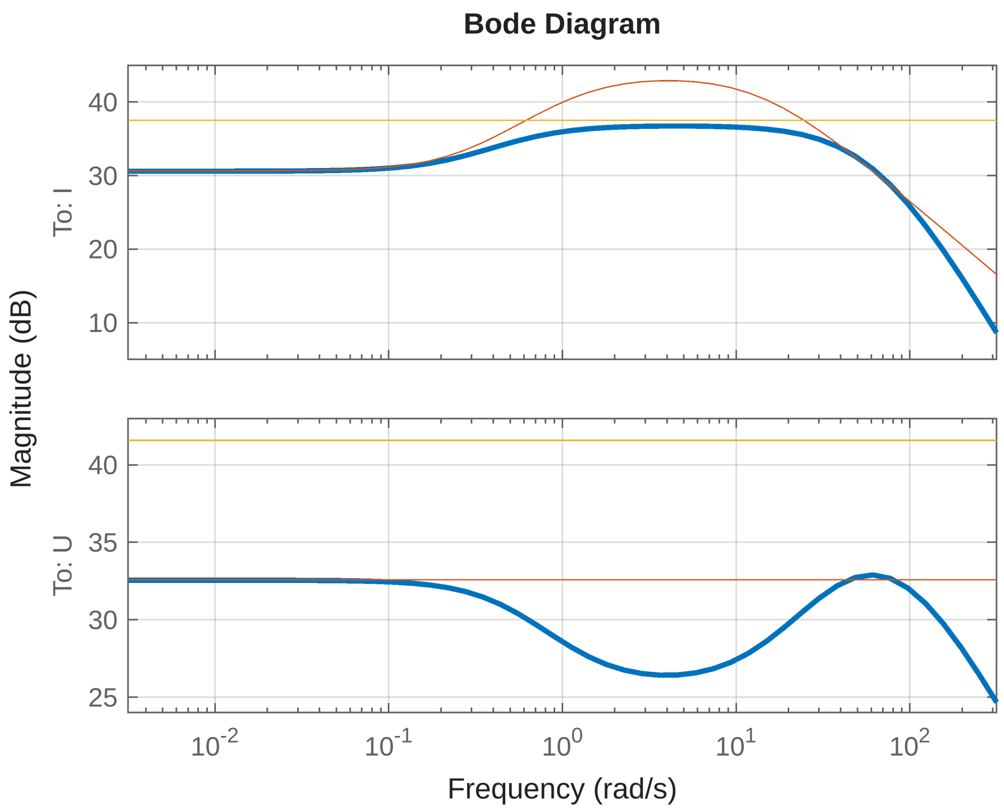

```
norm(lft(GPWeigthed,Khinf),inf) %GAM, within tolerances
```

```
ans =
0.9996
```

```
CLnotWeigthed=lft(GP,Khinf);
CLInputWeigthed=CLnotWeigthed*Win1;
bodemag(CLnotWeigthed(2),errScaled_naive,PlantillaMaxErr,logspace(-3,2)),
grid on
h=gca;h.Responses(1).LineWidth=2;
legend("Hinf","naive (no válido)","TemplateMaxErr",Location="best")
```




```
bodemag(CLInputWeigthed([1 3]),
[G(1);1]*K_OL_gain*Wref,tf([MaxI;MaxU]),logspace(-2.5,2.5)), grid on
legend("Hinf","naive (no válido)","bounds",Location="best")
h=gca;h.Responses(1).LineWidth=2;
```




```
step(CLInputWeigthed+[0;Wref;0],tf([MaxI;Wref;MaxU])), grid on
```

```
legend("Hinf","bounds")
h=gca;h.Responses(1).LineWidth=2;
```