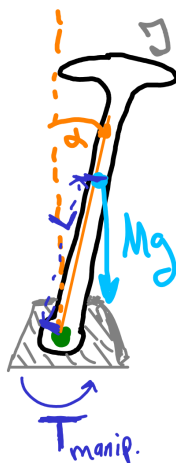


# Control PD péndulo en posición vertical inestable



© 2026, Antonio Sala, Universitat Politècnica de València. Todos los derechos reservados.

**Objetivos:** 1) diseñar un PD con controlabilidad total para un proceso de orden 2.

2) ilustrar con un ejemplo que **cancelar un polo inestable** del proceso mediante un cero del regulador (p.ej. en PD o PID) **no funcionará NUNCA**. La función de transferencia de referencia a salida controlada será "nominalmente" estable pero: **(a)** ante error de modelado infinitesimal no será estable en la práctica (no es "robustamente estable"), y **(b)** la función de transferencia de perturbación a la entrada de proceso a salida controlada es inestable, incluso sin error de modelado.

\*Se deja como ejercicio probar que cancelar un cero de fase no mínima con un polo del regulador tampoco funcionaría: **(a)** no sería "robustamente estable", y **(b)** la FdT de referencia a acción de control sería inestable incluso sin error de modelado.

Presentaciones en vídeo:

<https://personales.upv.es/asala/YT/V/pendi1.html> [modelado]

<https://personales.upv.es/asala/YT/V/pendipd.html> [PD sin cancelación]

<https://personales.upv.es/asala/YT/V/pendipdcanmal.html> [PD cancelando polo inestable, MAL]

<https://personales.upv.es/asala/YT/V/pendipdcanbe.html> [PD cancelando polo estable]

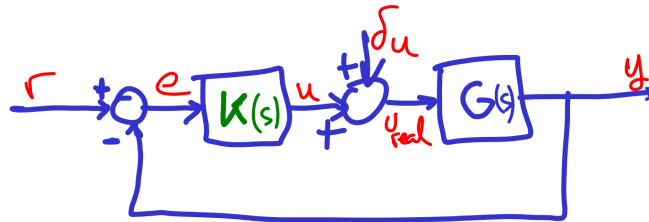
## Tabla de Contenidos

Preliminares.....	2
Ejemplo diseño PD sin usar cancelación (asignando 2 polos en lugares diferentes a planta en bucle abierto).....	3
Simulemos el regulador diseñado.....	4
Regulador PD de "cancelación", teoría (cancelación polo estable).....	7

CANCELACIÓN inestable NO FUNCIONA.....	10
Razón 1: error de modelado.....	10
Razón 2: perturbaciones, ¡incluso sin error de modelado!.....	11
Anexo: symbolic toolbox, caso 1.....	12

## Preliminares

El diagrama de bloques de la figura:



tiene como ecuación de la salida controlada  $y(s)$  en dominio de Laplace:

$$y(s) = \frac{GK}{1 + GK} \cdot r(s) + \frac{G}{1 + GK} \cdot \delta_u(s)$$

\*De forma equivalente, podemos trabajar con el error, llegando a conclusiones idénticas en el razonamiento a continuación:

$$e(s) = \frac{1}{1 + GK} \cdot r(s) - \frac{G}{1 + GK} \cdot \delta_u(s)$$

Si el modelo de  $G$  fuera  $G(s) = \frac{2}{(s-1)(s+1)}$ , un control proporcional  $K(s) = K_P$  nunca estabilizaría, dado que la ec. característica de bucle cerrado sería  $s^2 + (2K_P - 1) = 0$ , que nunca tendrá dos raíces de parte real negativa.

```
s=tf('s');
G=2/(s-1)/(s+1);
```

Un control PD  $K(s) = K_D \cdot s + K_P$  sí que podría estabilizar\*, porque, operando, tendríamos, para cambios de referencia:

$$y(s) = \frac{\frac{2}{(s-1)(s+1)}(K_D s + K_P)}{1 + \frac{2}{(s-1)(s+1)}(K_D s + K_P)} r(s) = \frac{2(K_D s + K_P)}{s^2 + 2K_D s + (2K_P - 1)} r(s)$$

$$e(s) = \frac{1}{1 + \frac{2}{(s-1)(s+1)}(K_D s + K_P)} r(s) = \frac{(s-1)(s+1)}{s^2 + 2K_D s + (2K_P - 1)} r(s)$$

y para cambios de perturbación (entonces  $e = 0 - y$ ):

$$y(s) = \frac{\frac{2}{(s-1)(s+1)}}{1 + \frac{2}{(s-1)(s+1)}(K_D s + K_P)} \delta_u(s) = \frac{2}{s^2 + 2K_D s + (2K_P - 1)} \delta_u(s)$$

y tenemos "**controlabilidad total**": los polos pueden asignarse donde queramos en el plano complejo... bueno, podríamos tener limitaciones si necesitáramos error estacionario pequeño,

$$y(\infty) = \frac{2K_P}{2K_P - 1} \cdot r(\infty) + \frac{2}{2K_P - 1} \cdot \delta_u(\infty),$$

$$e(\infty) = \frac{-1}{2K_P - 1} \cdot r(\infty) - \frac{2}{2K_P - 1} \cdot \delta_u(\infty)$$

ya que  $K_P$  no sería enteramente libre.

**\*Nota:** el PD necesitará "filtro de ruido paso-bajo" para poder funcionar en la práctica y ser realizable, pero dicho detalle no es objetivo de este material.

## Ejemplo diseño PD sin usar cancelación (asignando 2 polos en lugares diferentes a planta en bucle abierto)

A partir de  $s^2 + 2K_D s + (2K_P - 1) = \text{poliniodeseado}(s)$ , Si se deseara  $t_{est,98\%}$  en cierto valor prefijado, sin oscilaciones, podríamos plantear

```
t_est_deseado=2;
sigma_deseado=4/t_est_deseado
```

```
sigma_deseado =
2
```

```
%sigma_deseado=sqrt(20)
```

Asignando dos polos reales  $(s + \sigma)^2 = s^2 + 2\sigma \cdot s + \sigma^2$ , tendremos  $2K_D = 2\sigma$ ,

$2K_P - 1 = \sigma^2$ , o sea  $K_P = (\sigma^2 + 1)/2$ ,  $K_D = \sigma$ .

```
Kd=sigma_deseado
```

```
Kd =
2
```

```
Kp=(sigma_deseado^2+1)/2
```

```
Kp =
2.5000
```

El error estacionario en bucle cerrado será:

```
Gan=dcgain(G)
```

```
Gan =  
-2
```

```
e_pos_ref=1/(1+Gan*Kp)
```

```
e_pos_ref =  
-0.2500
```

```
e_pos_pert=Gan/(1+Gan*Kp)
```

```
e_pos_pert =  
0.5000
```

Si se considera suficientemente pequeño, el diseño "PD" estaría terminado.

En caso contrario, habría que hacer una de estas opciones

- "acelerar" el transitorio pidiendo mayor  $\sigma$ , o poner dos polos diferentes, p.ej.

$$(s + 2)(s + 11) = s^2 + 13s + 22, \text{ o}$$

- admitir polos complejos, con polinomio deseado  $(s + 2)^2 + \omega_p^2 = s^2 + 4s + (4 + \omega_p^2)$  lo que no cambiaría  $K_D$  pero subiría  $K_P$ .

o, incluso plantearse ambas cosas a la vez, para conseguir un regulador con  $K_p$  mayor.

**\*NOTA:** Si el error pedido fuera muy pequeño y requiriera constantes "inaceptablemente altas" que harían que esto no funcionara "en la práctica" (bueno, ya no funcionará sin filtro de ruido, pero esa es otra cuestión), podría pensarse en utilizar un **PID** completo: la acción integral hará el error cero sin necesidad de subir en exceso las ganancias proporcional o derivada.

### Simulemos el regulador diseñado

```
K=Kd*s+Kp;  
T2=feedback(G*K,1)
```

```
T2 =
```

$$\frac{4s + 5}{s^2 + 4s + 4}$$

Continuous-time transfer function.  
Model Properties

```
zpk(T2)
```

```
ans =
```

$$\frac{4(s+1.25)}{(s+2)^2}$$

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

```
pole(T2)'
```

```
ans = 1x2  
      -2    -2
```

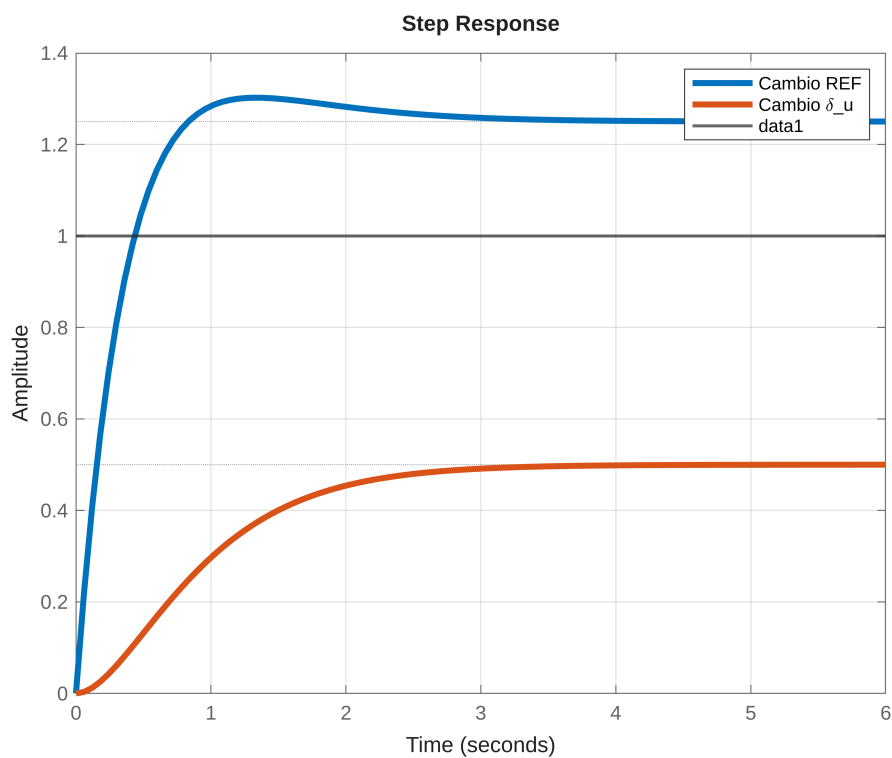
```
BCPert2=feedback(G,K);  
zpk(BCPert2)
```

```
ans =
```

$$\frac{2}{(s+2)^2}$$

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

```
h=stepplot(T2,BCPert2,6); grid on  
h.Responses(1).LineWidth=3;  
h.Responses(2).LineWidth=3; yline(1,LineWidth=1.5)  
legend("Cambio REF", "Cambio \delta_u")
```



```
refToU=feedback(K,G) %NO realizable sin filtro de ruido, esto no
funcionará NUNCA
```

```
refToU =
```

$$\frac{2 s^3 + 2.5 s^2 - 2 s - 2.5}{s^2 + 4 s + 4}$$

```
Continuous-time transfer function.
Model Properties
```

También en muchos casos, en vez del error se deriva la salida (para evitar "patada derivativa" ante cambio de referencia escalón).

Ello no lo vamos a considerar aquí en detalle, por estar fuera de los objetivos de este material, aunque simplemente lo simularemos:

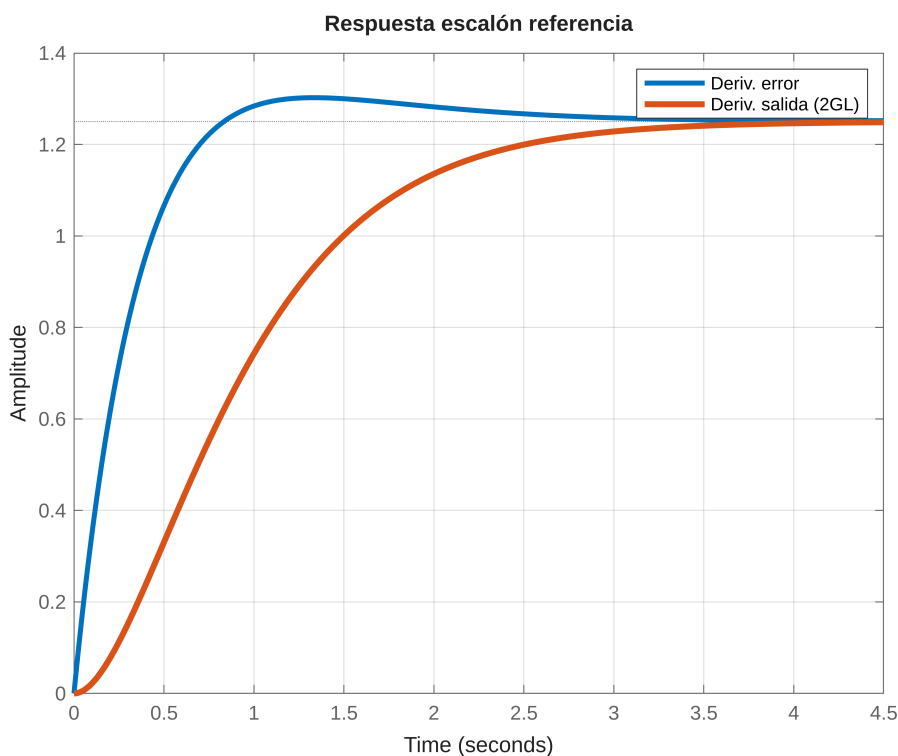
```
tmp=feedback(G,Kd*s);
T2bis=feedback(tmp*Kp,1); zpk(T2bis)
```

```
ans =
```

$$\frac{5}{(s+2)^2}$$

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

```
h=stepplot(T2,T2bis); grid on
h.Responses(1).LineWidth=2.5; h.Responses(2).LineWidth=3;
legend("Deriv. error", "Deriv. salida (2GL)")
title("Respuesta escalón referencia")
```



## Regulador PD de "cancelación", teoría (cancelación polo estable)

Para facilitar los cálculos (o si nos obligan a que de referencia a salida controlada la dinámica sea de **primer orden**), podríamos pensar en un **PD factorizado**

$K(s) = K_c(s + \alpha)$ , siendo  $K_c = K_D$  y  $\alpha = \frac{K_P}{K_D}$ . Si hiciéramos que  $(s + \alpha)$  cancelara alguno

de los polos del proceso, la ec. característica (ante referencia) con la que hacer operaciones sería de **orden 1**.

Tenemos dos opciones:

- 1• Cancelar el polo **estable**  $\alpha = 1$ .

$$y(s) = \frac{\frac{2}{(s-1)(s+1)} K_c(s+1)}{1 + \frac{2}{(s-1)(s+1)} K_c(s+1)} r(s) = \frac{\frac{2}{(s-1)} K_c}{1 + \frac{2}{(s-1)} K_c} r(s) = \frac{2K_c}{s-1+2K_c} r(s) \text{ [referencia]}$$

$$y(s) = \frac{\frac{2}{(s-1)(s+1)}}{1 + \frac{2}{(s-1)(s+1)} K_c(s+1)} \cdot \delta_u(s) = \frac{\frac{2}{(s-1)(s+1)}}{1 + \frac{2}{(s-1)} K_c} \cdot \delta_u(s) = \frac{2}{(s-1+2K_c) \cdot (s+1)} \cdot \delta_u(s)$$

[perturbación entrada]

Asignando el polo de FdT de la referencia:  $-1 + 2K_c = \sigma$  resulta

```
sigma_deseado=20
```

```
sigma_deseado =  
20
```

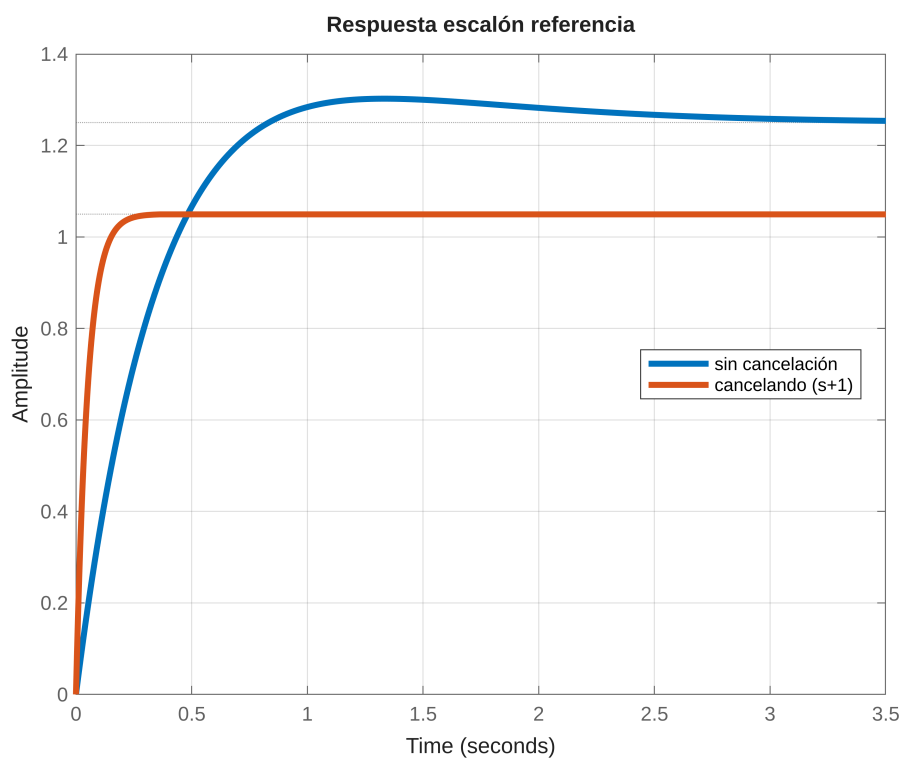
```
Kc=(sigma_deseado+1)/2
```

```
Kc =  
10.5000
```

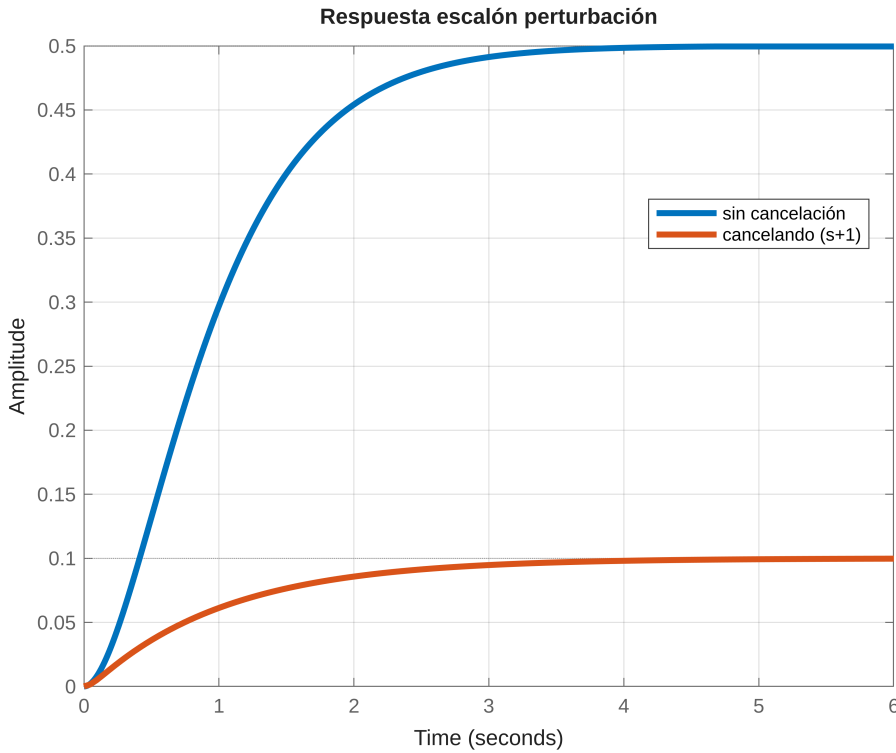
```
epos=-1/(2*Kc-1)
```

```
epos =  
-0.0500
```

```
K=Kc*(s+1);  
T=feedback(G*K,1);  
BCPert=feedback(G,K);  
h=stepplot(T2,T); grid on  
h.Responses(1).LineWidth=3;  
h.Responses(2).LineWidth=3;  
legend("sin cancelación","cancelando (s+1)", Location="best"),  
title("Respuesta escalón referencia")
```



```
h=stepplot(BCPert2,BCPert); grid on  
h.Responses(1).LineWidth=3;  
h.Responses(2).LineWidth=3;  
legend("sin cancelación","cancelando (s+1)", Location="best"),  
title("Respuesta escalón perturbación")
```



\*Nótese que la perturbación tiene un polo en  $-1$ : el sistema es de 2o orden y trabajar "asignando un polo" hace que el otro no se mueva en algunas funciones de transferencia. Depende de la amplitud/rapidez de  $\delta_u$  si esta solución es o no aceptable en la práctica... y  $\delta_u$  muchas veces no sabremos cómo es hasta que hagamos experimentos.

●2● Cancelar el polo **inestable**  $\alpha = -1$ .

$$y(s) = \frac{\frac{2}{(s-1)(s+1)} K_c (s-1)}{1 + \frac{2}{(s-1)(s+1)} K_c (s-1)} r(s) = \frac{\frac{2}{(s+1)} K_c}{1 + \frac{2}{(s+1)} K_c} r(s) = \frac{2K_c}{s+1+2K_c} r(s)$$

$$y(s) = \frac{\frac{2}{(s-1)(s+1)}}{1 + \frac{2}{(s-1)(s+1)} K_c (s-1)} \cdot \delta_u(s) = \frac{\frac{2}{(s-1)(s+1)}}{1 + \frac{2}{(s+1)} K_c} \cdot \delta_u(s) = \frac{2}{(s+1+2K_c) \cdot (s-1)} \cdot \delta_u(s)$$

Si queremos el único polo asignable en  $s = -7$ , necesitaremos  $K_c = 3$ , y la dinámica será

$$y(s) = \frac{6}{s+7} \cdot r(s) + \frac{2}{(s+7)(s-1)} \cdot \delta_u(s)$$

Esta segunda opción **NO funciona**. Es inestable ante  $\delta_u$ , que siempre existirá en la práctica... y ante referencia "parece estable pero no lo es robustamente".

Si sólo miráramos "r", haríamos  $1 + 2K_c = \sigma$ , esto es:

```
Kc=(sigma_deseado-1)/2;  
K=Kc*(s-1);
```

## CANCELACIÓN inestable NO FUNCIONA

### Razón 1: error de modelado

```
s=tf('s');  
G=2/(s-1.0004)/(s+1);  
T=feedback(G*K,1);  
zpk(T)
```

ans =

```
    19 (s-1)  
-----  
 (s+20) (s-1)
```

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

```
format long  
tzero(T)
```

ans =  
1.0000000000000000

```
pole(T)
```

ans = 2x1  
-19.999638095894625  
1.000038095894622

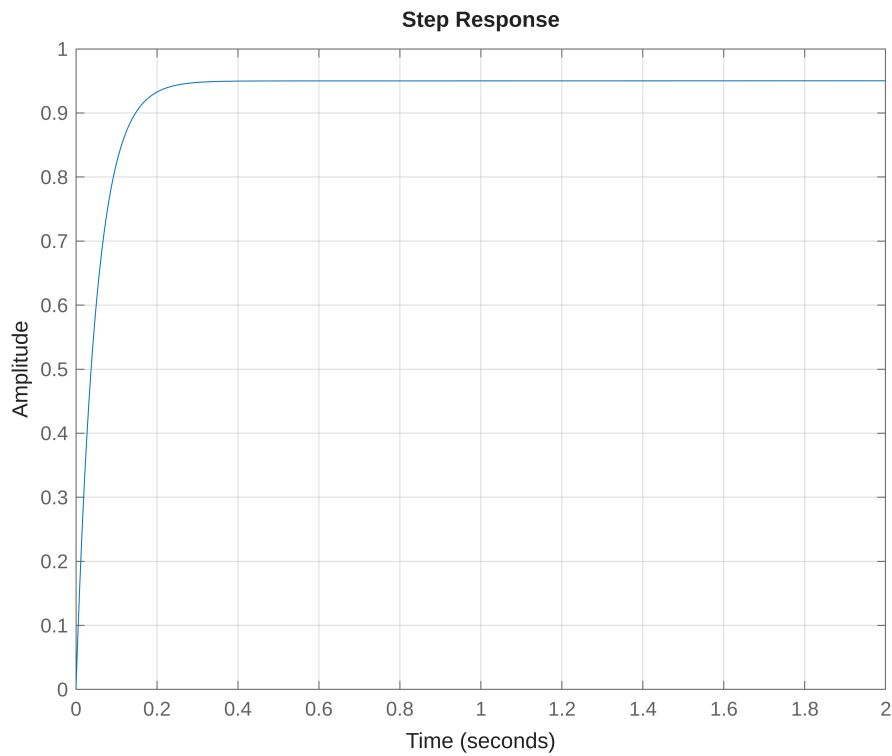
```
format short  
T=minreal(zpk(T))
```

T =

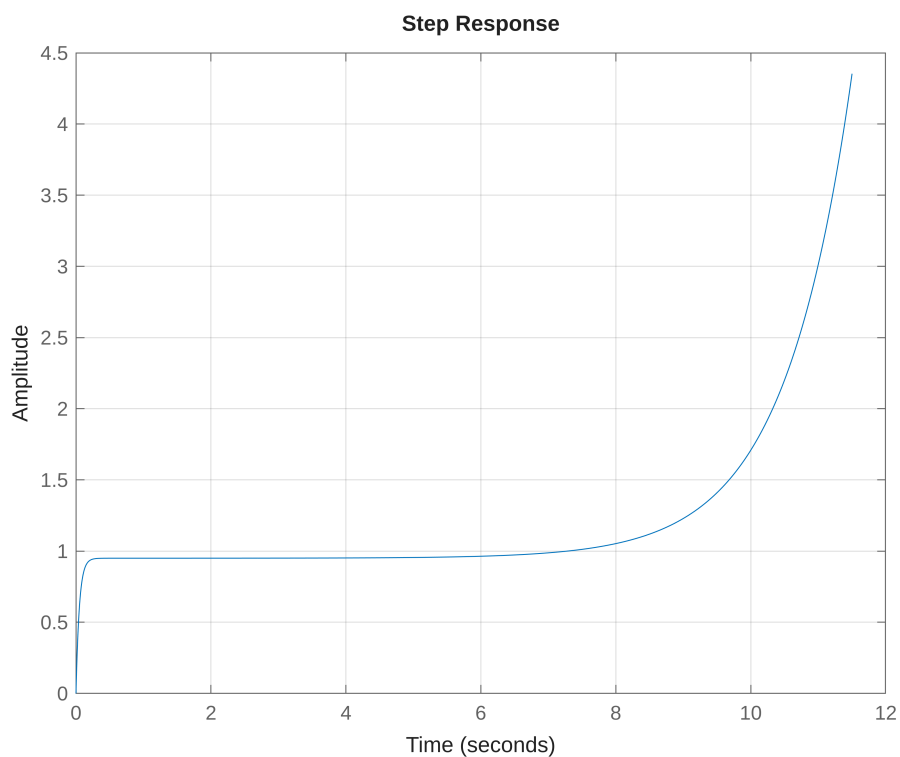
```
    19 (s-1)  
-----  
 (s+20) (s-1)
```

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

```
step(T,2), grid on
```



```
step(T,11.5), grid on
```



## Razón 2: perturbaciones, ¡incluso sin error de modelado!

```
G=2/(s-1)/(s+1);
D=feedback(G,K);
zpk(D)
```

ans =

$$\frac{2}{(s+20)(s-1)}$$

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

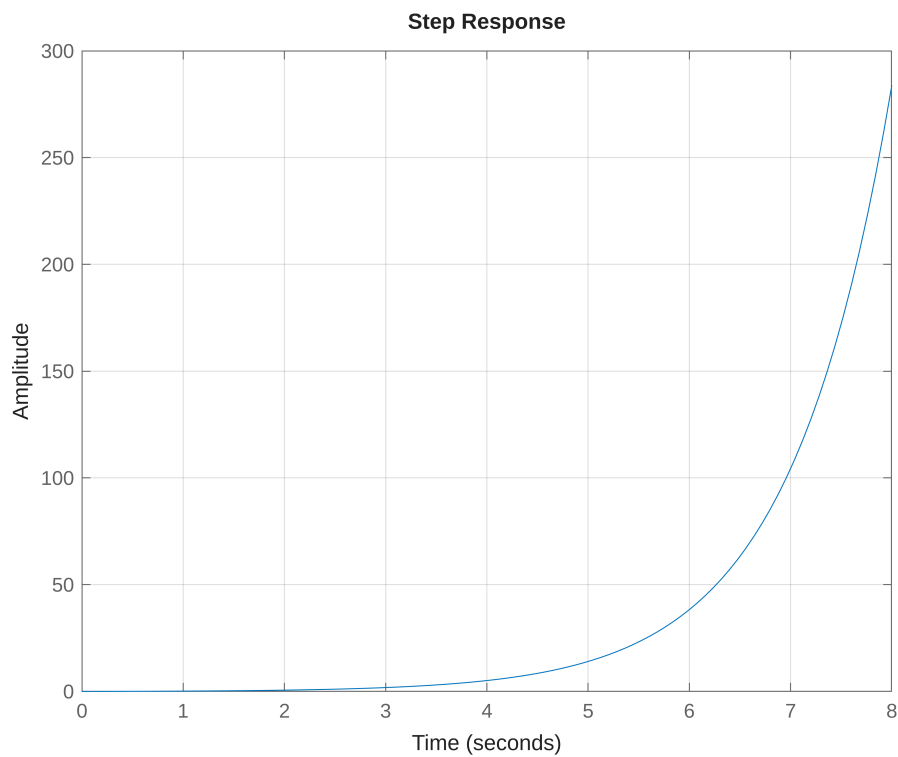
```
D=minreal(zpk(D))
```

D =

$$\frac{2}{(s+20)(s-1)}$$

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

```
step(D,8), grid on
```



## Anexo: symbolic toolbox, caso 1

```
syms s t  
G=2/(s-1.0001)/(s+1)
```

G =

$$\frac{2}{(s+1)\left(s-\frac{10001}{10000}\right)}$$

```
K=3*(s-1)
```

$$K = 3s - 3$$

```
T=simplifyFraction(G*K/(1+G*K));  
vpa(T,4)
```

ans =

$$\frac{60000.0(s-1.0)}{10000.0s^2 + 60000.0s - 70000.0}$$

```
PF=partfrac(T*1/s,FactorMode="real");  
vpa(PF,4)
```

ans =

$$\frac{1.875e-5}{s-1.0} - \frac{0.8571}{s+7.0} + \frac{0.8571}{s}$$

```
salida(t)=vpa(ilaplace(PF),4)
```

salida(t) =

$$0.000018749824220581011772667163234319 e^{1.0000250002343764648460387468329 t} - 0.8571493622440431549$$

```
salida(1)
```

ans = 0.85639990333183827446278003910283

```
salida(2)
```

ans = 0.85726844999901621590417604383725

```
salida(13)
```

ans = 9.1550003401703414411328520013563

```
salida(20)
```

ans = 9102.1688235032349495347444679004