

Asignación de polos con control **Proporcional**, sistema de **segundo** orden, symbolic toolbox

© 2023, Antonio Sala Piqueras, Universitat Politècnica de València. Todos los derechos reservados.

Presentación en vídeo: <http://personales.upv.es/asala/YT/V/ord2ejP.html>

*Este código funcionó sin errores en Matlab **R2022b** (Linux)

Objetivo: calcular control **proporcional** para cumplir especificaciones de *error ante cambios de referencia* en bucle cerrado en un proceso de **SEGUNDO orden** sin integradores (**tipo 0**). Determinar si es posible o no cumplir ciertas especificaciones; en este desarrollo introductorio no se consideran perturbaciones, saturación del actuador, etc.

Tabla de Contenidos

Control PROPORCIONAL, análisis bucle cerrado.....	2
Análisis de especificaciones estáticas (error de posición).....	2
Análisis de especificaciones dinámicas (polos de bucle cerrado).....	3
Ejemplo de especificaciones factibles con regulador proporcional	6
Ejemplo de especificaciones NO factibles con regulador proporcional	8

```
syms s
G(s)=27/(s+1)/(s+9); %Proceso a controlar, Symbolic Toolbox
G(0) %ganancia estática
```

```
ans = 3
```

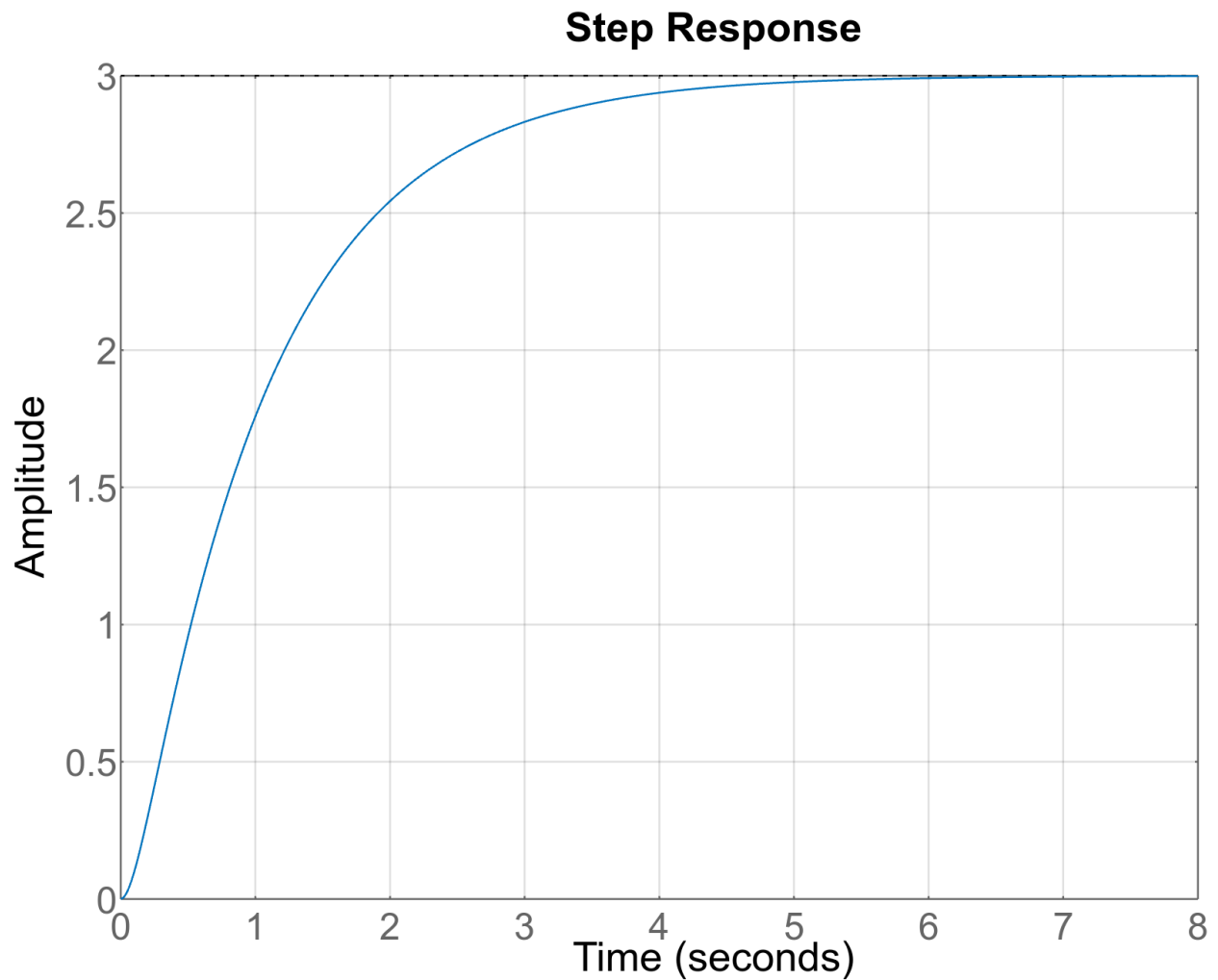
```
Gs=tf(27,[1 10 9]); %Proceso a controlar, Control Toolbox [incompatible con
Symbolic]
dcgain(Gs) %ganancia estática
```

```
ans = 3
```

```
t_est_bucleabierto=4/1 %tiempo 98% de establecimiento sin "control".
```

```
t_est_bucleabierto = 4
```

```
step(Gs), grid on
```



Control PROPORCIONAL, análisis bucle cerrado

```
syms K_p real
K=K_p;
E(s)=1/(1+G(s)*K) %error ante cambios de referencia
```

$E(s) =$

$$\frac{1}{\frac{27 K_p}{(s+1)(s+9)} + 1}$$

```
E(s)=simplifyFraction(E(s)) %puesto en "bonito"
```

$E(s) =$

$$\frac{(s+1)(s+9)}{s^2 + 10s + 27K_p + 9}$$

Análisis de especificaciones estáticas (error de posición)

```
e_p=simplify(E(0)) %valor final error ante ref = 1
```

$$e_p = \frac{1}{3K_p + 1}$$

Haciendo K_p grande, el error puede ser todo lo pequeño que se desee (pero siempre >0 , no "exactamente cero"). Aparte, si se quiere error "muuuuy" pequeño, K_p muy grande va a tener oscilaciones, saturación y amplificación de ruido inaceptables en la práctica.

Análisis de especificaciones dinámicas (polos de bucle cerrado)

```
[Ne, De]=numden(E)
```

$$\begin{aligned} N_e(s) &= (s+1)(s+9) \\ D_e(s) &= s^2 + 10s + 27K_p + 9 \end{aligned}$$

```
solve(De==0,s) %polos de bucle cerrado en función de K_p
```

$$\text{ans} = \begin{pmatrix} -\sqrt{16-27K_p}-5 \\ \sqrt{16-27K_p}-5 \end{pmatrix}$$

Las raíces serán reales si $K_p \leq \frac{16}{27}$, una a la izquierda y otra a la derecha de -5 .

Si $K_p > \frac{16}{27}$ entonces tendrá dos raíces complejas, $-5 \pm \omega_p \cdot j$ de parte real -5 .

```
Klim_reales=16/27
```

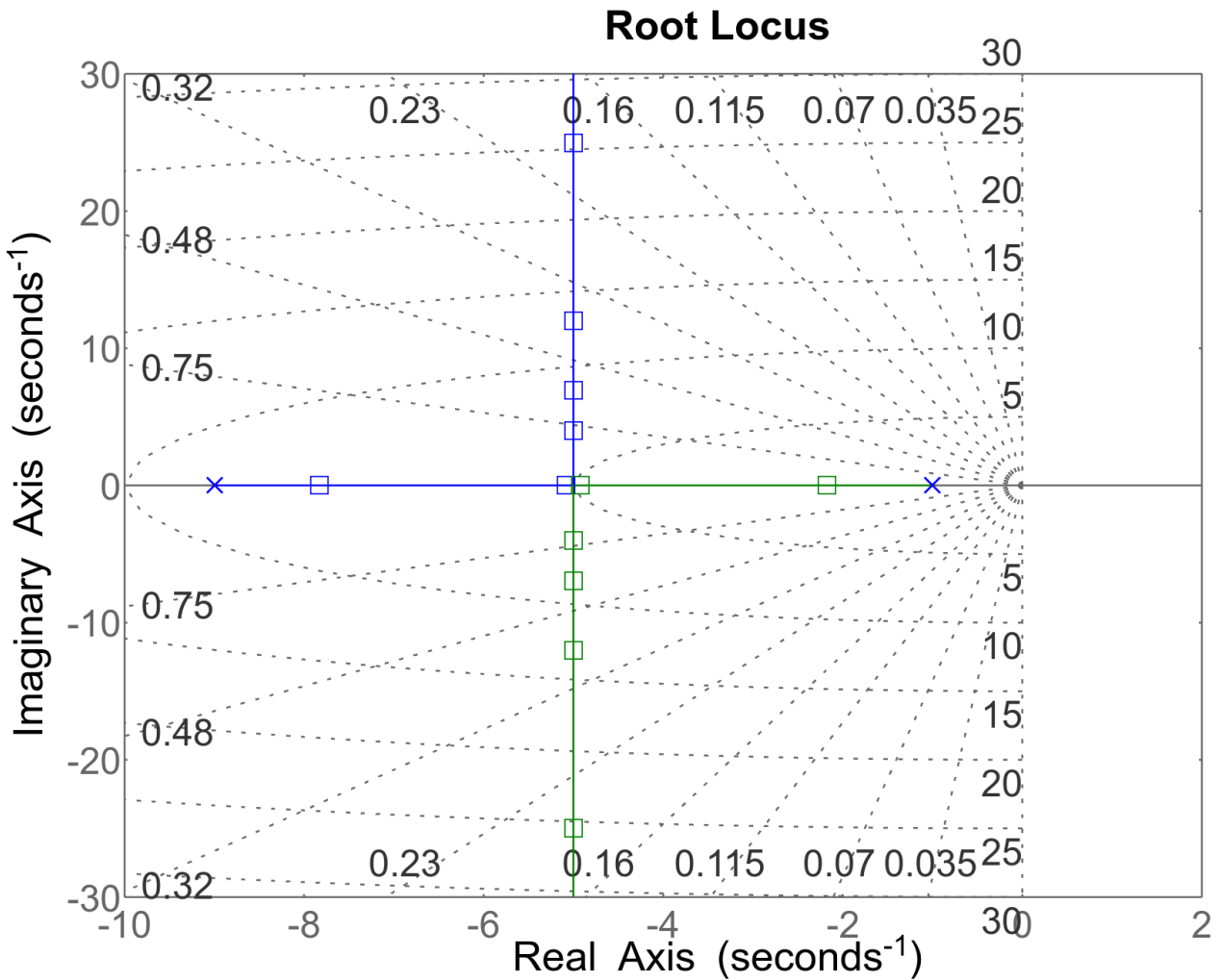
```
Klim_reales = 0.5926
```

Por tanto, la parte real "más a la izquierda" del plano complejo que se podrá conseguir es -5 , con tiempo de establecimiento aproximado de $4/5 = 0.8$ segundos.

En cuanto a sobreoscilación, $\omega_n = \sqrt{27K_p + 9}$, y el factor de amortiguamiento se obtendría a partir de

$$2\xi\omega_n = 10, \text{ o sea } \xi = \frac{5}{\sqrt{27K_p + 9}}.$$

```
rlocus(Gs), grid on  
Kp_test=16/27* [.5 0.9996 2 4 10 40];  
hold on, rlocus(Gs,Kp_test,'square'), hold off
```



Si se prescribiese que en bucle cerrado $t_{e,deseado} < 4/5 = 0.8$ (crit. 98%) entonces NO podrá hacerse con regulador proporcional.

Si el error de posición es muy pequeño, se necesita K_p elevada y, seguramente, la sobreoscilación sea inaceptable.

Antes de entrar en detalle con especificaciones concretas, vamos a simular varias constantes para entender "intuitivamente" el lugar de las raíces:

```
BC1=feedback(Gs*Kp_test(1),1);
eig(BC1)
```

```
ans = 2x1
    -7.8284
    -2.1716
```

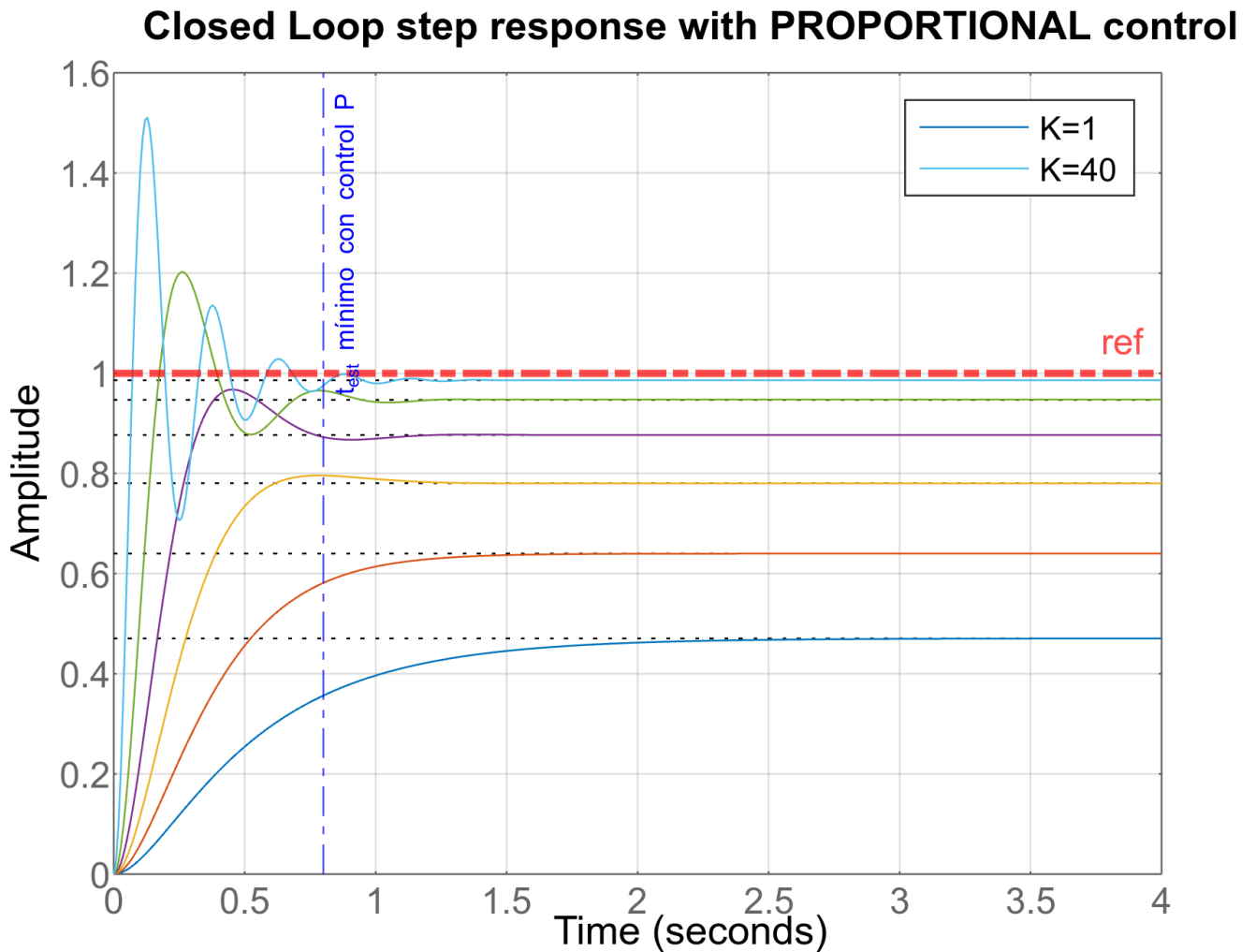
```
BC2=feedback(Gs*Kp_test(2),1);
eig(BC2)
```

```
ans = 2×1
    -5.0800
    -4.9200
```

```
BC3=feedback(Gs*Kp_test(3),1);
eig(BC3)
```

```
ans = 2×1 complex
    -5.0000 + 4.0000i
    -5.0000 - 4.0000i
```

```
BC4=feedback(Gs*Kp_test(4),1);
BC5=feedback(Gs*Kp_test(5),1);
BC6=feedback(Gs*Kp_test(6),1);
step(BC1,BC2,BC3,BC4,BC5,BC6), grid on,
yline(1,'-.r',Label="ref",LineWidth=2)
xline(0.8,"-.b",label="t_{est} mínimo con control P",FontSize=7)
legend("K=1","","","","","K=40"), title("Closed Loop step response with
PROPORTIONAL control")
```



Ejemplo de especificaciones factibles con regulador proporcional

Por ejemplo, $t_{est} (98\%) \leq 1.5$ s, $e_p \leq 0.05$, $\delta < 0.3$ se conseguiría con $\sigma > 4/1.5$,

```
sol_sigma=solve(-5+sqrt(16-27*K_p)<=-4/1.5,ReturnConditions=true)
```

```
sol_sigma = struct with fields:
    K_p: x
    parameters: x
    conditions: x <= 16/27 & 95/243 <= x
```

```
vpa(sol_sigma.conditions,4)
```

```
ans =  $x \leq 0.5926 \wedge 0.3909 \leq x$ 
```

```
ep_deseado=0.05;
sol_e_p=solve(e_p<=ep_deseado, K_p>=0, ReturnConditions=true); %para obtener
todas las soluciones que cumplen
vpa(sol_e_p.conditions,4)
```

```
ans =  $6.333 \leq x$ 
```

```
delta_deseado=0.3;
tmp=log(delta_deseado)^2;
xi_deseado=sqrt(tmp/(tmp+pi^2))
```

```
xi_deseado = 0.3579
```

```
xi=5/sqrt(27*K_p+9);
sol_delta=solve(xi>=xi_deseado,ReturnConditions=true);
vpa(sol_delta.conditions,4)
```

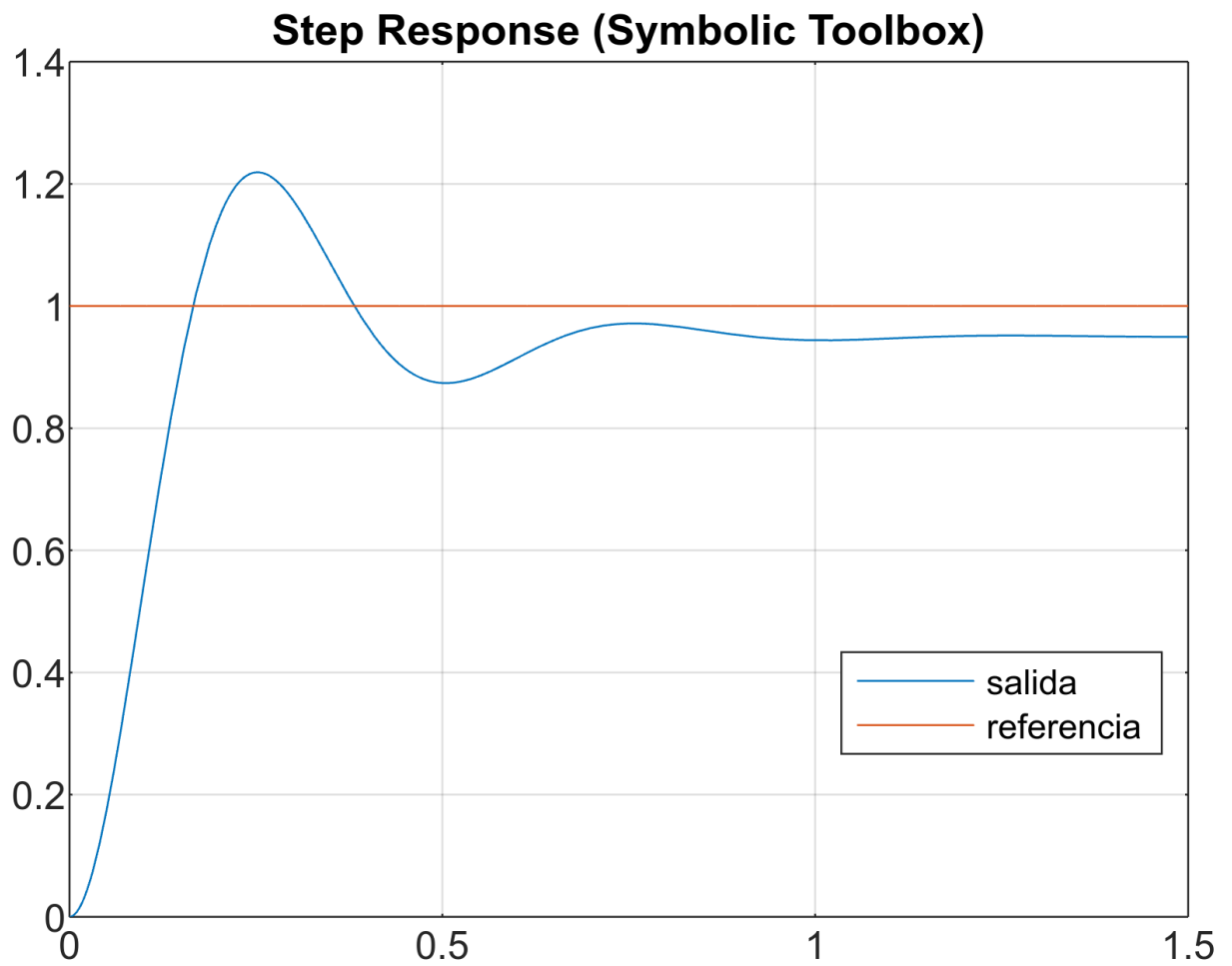
```
ans =  $x \leq 6.897 \wedge -0.3333 < x$ 
```

Representemos la respuesta escalón del regulador con la menor ganancia que cumple las especificaciones:

```
K_p=6.333333
```

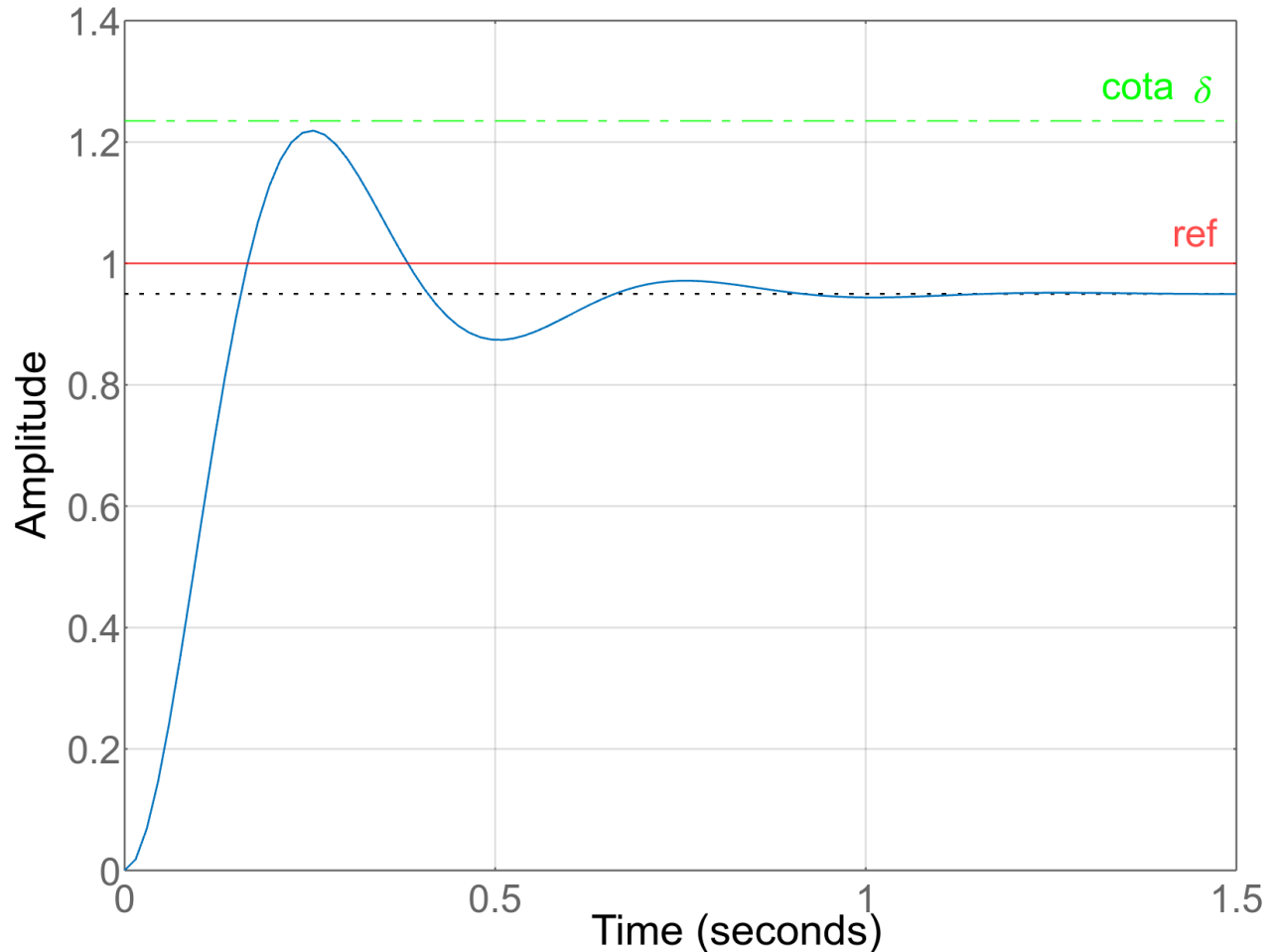
```
K_p = 6.3333
```

```
salida=ilaplace(eval(G*K/(1+G*K)*1/s)); %symbolic toolbox
fplot([salida;1],[0 1.5]), grid on, ylim([0 1.4])
legend("salida","referencia", Location="best")
title("Step Response (Symbolic Toolbox)")
```



```
step(feedback(Gs*K_p,1),1.5), yline(1,'-r',Label="ref"),
title("Step Response (Control Systems Toolbox)")
yline(0.95*1.3,'-.g',Label="cota \delta"), grid on %control toolbox
```

Step Response (Control Systems Toolbox)



Ejemplo de especificaciones NO factibles con regulador proporcional

Por ejemplo, $t_{est} (98\%) \leq 1 \text{ s}$, $e_p \leq 0.04$, $\delta < 0.2$ se conseguiría con $\sigma > 4/1$,

```
syms K_p real
sol_sigma=solve(-5+sqrt(16-27*K_p)<=-4/1,ReturnConditions=true);
vpa(sol_sigma.conditions,4)
```

ans = $0.5556 \leq x \wedge x \leq 0.5926$

```
ep_deseado=0.04;
sol_e_p=solve(e_p<=ep_deseado, K_p>=0, ReturnConditions=true); %para obtener
todas las soluciones que cumplen
vpa(sol_e_p.conditions,4)
```

ans = $8.0 \leq x$

```
delta_deseado=0.2;
tmp=log(delta_deseado)^2;
```

```
xi_deseado=sqrt(tmp/(tmp+pi^2))
```

```
xi_deseado = 0.4559
```

```
xi=5/sqrt(27*K_p+9);  
sol_delta=solve(xi>=xi_deseado,ReturnConditions=true);  
vpa(sol_delta.conditions,4)
```

```
ans =  $x \leq 4.121 \wedge -0.3333 < x$ 
```

No podemos encontrar un K_p que verifique todas las desigualdades que se deducen de cada especificación. Necesitaríamos control más complicado (**PI** que daría error cero, o **PD** que daría error no cero pero permitiría subir K_p al amortiguar mejor con K_d). El detalle de estos nuevos diseños NO es objeto de este material por brevedad.