Control proporcional para sistema de 1er orden, análisis teórico (symbolic toolbox) G(s)=2/(2*s+1)

Objetivo: comprender el comportamiento del control proporcional (simulaciones en videos complementarios) en un sistema de 1er orden.

Presentaciones en vídeo:

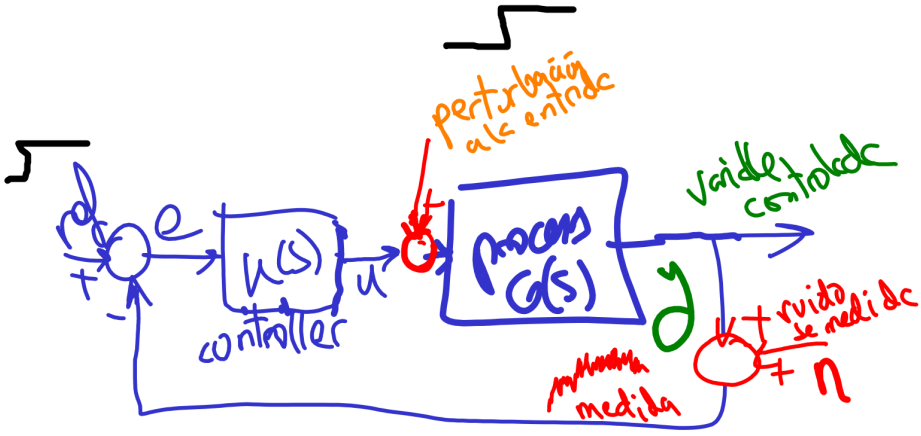
https://personales.upv.es/asala/YT/V/Pteo1.html (seguimiento referencia)

https://personales.upv.es/asala/YT/V/Pteo2.html (perturbación entrada, ruido de medida)

Nota: como la ganancia del controlador será una "variable de decisión", la mantendremos "simbólica" y usaremos la "symbolic math toolbox" para operaciones. Para un valor FIJADO de ganancia del controlador, existe una interfaz mucho más amigable en la "control systems toolbox" (e incluso para el diseño de control).

Funciones de transferencia de bucle cerrado

En este bucle:



Las ecuaciones son:

```
syms G K r u y n du e
LoopEquations={u == K*e,  y == G*(u+du), e == r-(y+n)};
```

Resolviendo:

```
sol=solve(LoopEquations, [u y e])
```

```
sol = struct with fields:
    u: -(K*(n - r + G*du))/(G*K + 1)
    y: (G*(du - K*n + K*r))/(G*K + 1)
    e: -(n - r + G*du)/(G*K + 1)
```

La salida controlada es:

```
collect(sol.y,[r du n])
```

```
ans =
```

$$\frac{G\,K}{G\,K+1}\,r+\frac{G}{G\,K+1}\,\mathrm{du}+\left(-\frac{G\,K}{G\,K+1}\right)n$$

El error $r-medida$, es:

```
collect(sol.e,[r du n])
```

```
ans =
```

$$\frac{r}{G\,K+1}+\left(-\frac{G}{G\,K+1}\right)\mathrm{du}+\left(-\frac{1}{G\,K+1}\right)n$$

La entrada al proceso que calcula el controlador (variable manipulada):

```
collect(sol.u,[r du n])
```

```
ans =
```

$$\frac{K}{G\,K+1}\,r+\left(-\frac{G\,K}{G\,K+1}\right)\mathrm{du}+\left(-\frac{K}{G\,K+1}\right)n$$

Ejemplo

```
clear %we discard the above code
syms s %Laplace Variable
syms K_c real %proportional control constant
G(s)=2/(2*s+1);
DCgainG=G(0)
```

```
DCgainG = 2
```

```
K=K_c; %Proportional Control (we might try other expressions in the
future)
```

Respuesta ante cambio de referencia

```
CLref(s)=collect(simplify(G*K/(1+G*K)),s)
```

```
CLref(s) =
```

$$\frac{2\,K_c}{2\,s+2\,K_c+1}$$

Dinámica:

```
[N,D]=numden(CLref);
solve(D==0,s) %closed-loop poles
```

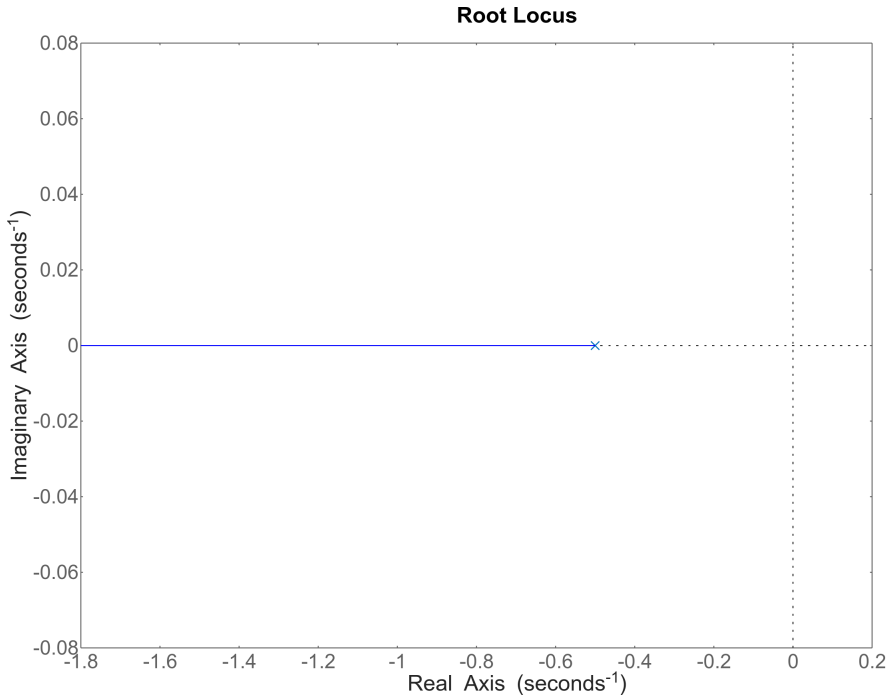
```
ans =
```

$$-K_c - \frac{1}{2}$$

El polo de bucle cerrado es $s = -(1/2 + K_c)$, tiempo de establecimiento (95%) es $3\tau = 3 \cdot \frac{1}{|polo|} = 3 \cdot \frac{1}{0.5 + K_c}$. Estable si $K_c > -1/2$... pero salvo que seas idiota, vas a elegir $K_c > 0$ (empuja hacia arriba si la salida está por debajo de la referencia), con lo que el controlador "P" es ESTABLE, y cuanto más ganancia K_c, más rápido es el transitorio.

Para trazar los polos en circuito cerrado en función de K_c, podemos usar el comando "lugar de las raices (root locus)" de Control Systems Toolbox, porque probablemente estarás familiarizado con el siguiente gráfico:

```
rlocus(tf(2,[2 1]))
```



Error estacionario (valor final): La ganancia estática de CLref es

```
DCGainREF=CLref(0) %subs(CLref,s,0)
```

```
DCGainREF =
```

$$\frac{2K_c}{2K_c+1}$$

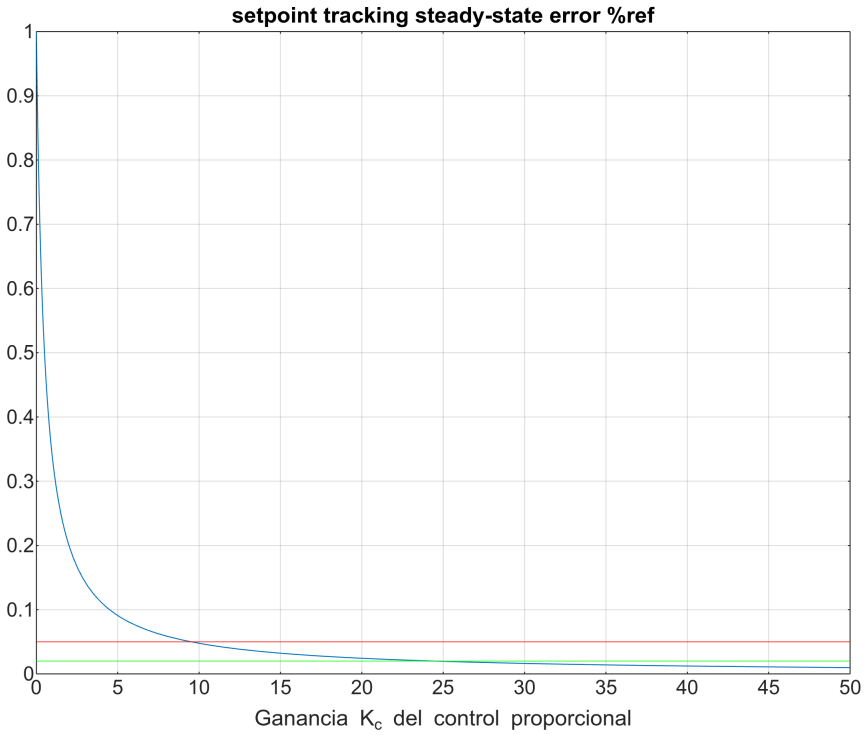
El valor final de la salida será $DCGainREF \cdot r_{finalvalue}$ y el error será $(1 - DCGainREF) \cdot r_{finalvalue}$, esto es:

```
DCGainErrREF=simplify(1-DCGainREF)
```

```
DCGainErrREF =
```

$$\frac{1}{2K_c+1}$$

```
fplot(DCGainErrREF,[0 50]), grid on, title("setpoint tracking
steady-state error %ref")
xlabel("Ganancia K_c del control proporcional")
yline(0.05,'r')
yline(0.02,'g')
ylim([0 1])
```



Respuesta a perturbación a la entrada

```
CLdu(s)=collect(simplify(G/(1+G*K)),s)
```

```
CLdu(s) =
```

$$\frac{2}{2s+2K_c+1}$$

Dinámica: los polos son los mismos que ante cambio de referencia.

Error estacionario final:

```
DCGain_Err_du=CLdu(0)%subs(CLdu,s,0)
```

```
DCGain_Err_du =
```

$$\frac{2}{2K_c+1}$$

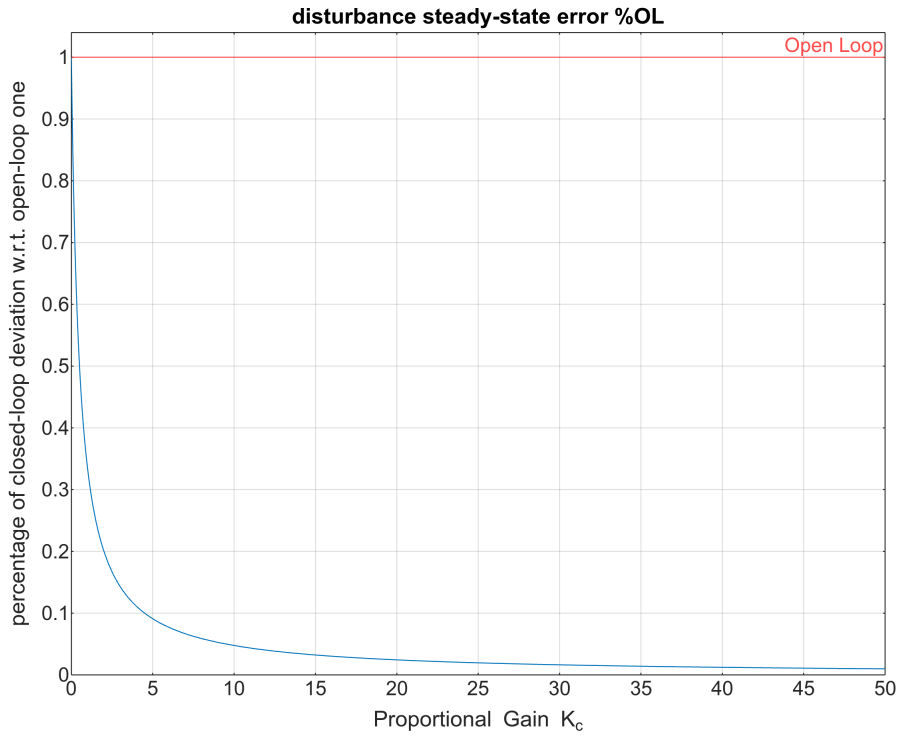
La cantidad "adimensional" que resulta de interés es la ratio entre el valor de error de "bucle abierto sin control" y de "bucle cerrado":

```
RatioCL2OL=DCGain_Err_du/DCgainG
```

```
RatioCL2OL =
```

$$\frac{1}{2K_c+1}$$

```
fplot(RatioCL2OL,[0 50]), grid on, title("disturbance steady-state
error %OL")
xlabel("Proportional Gain K_c")
yline(1,'r',label='Open Loop')
ylim([0 1.04]), ylabel("percentage of closed-loop deviation w.r.t.
open-loop one")
```



Respuesta a ruido de medida

En este caso normalmente nos interesa la amplificación del ruido de alta frecuencia al comando del actuador (variable manipulada):

```
CL_dy_2_u(s)=collect(simplify(K/(1+G*K)),s)
```

CL_dy_2_u(s) =

$$\frac{(2\,K_c)\,s + K_c}{2\,s + 2\,K_c + 1}$$

```
syms w real
FreqResponse=simplify( abs( CL_dy_2_u(1j*w) ), 100)
```

FreqResponse =

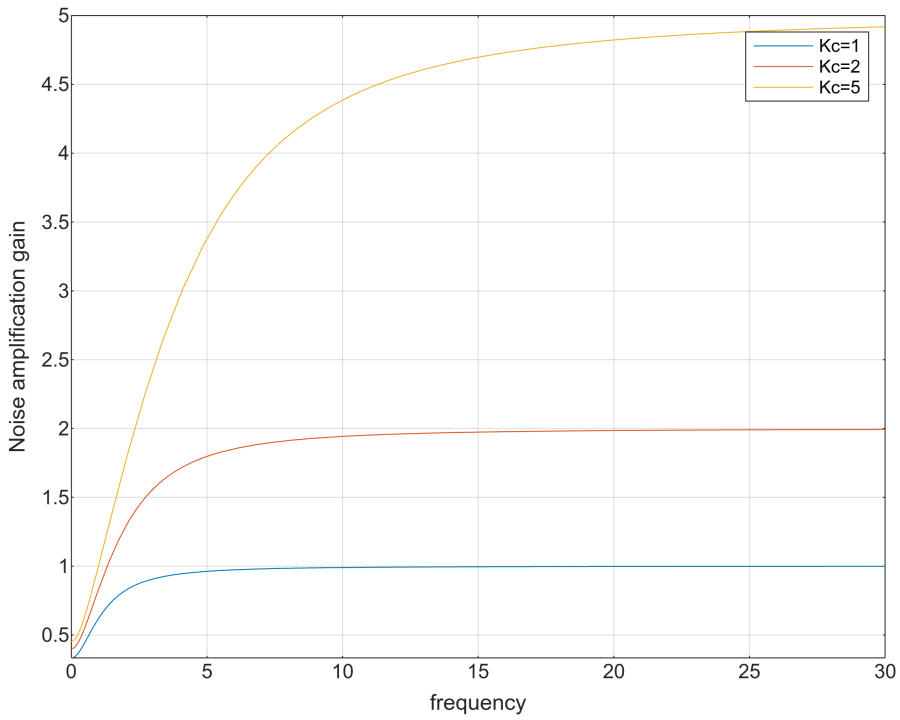
$$\frac{|K_c|\,\sqrt{4\,|w|^2 + 1}}{\sqrt{(2\,K_c + 1)^2 + 4\,w^2}}$$

```
limit(FreqResponse,w,inf) %high-frequency amplification
```

ans = $|K_c|$

Comparemos diferentes valores de K_c:

```
f1=subs(FreqResponse,K_c,1);
f2=subs(FreqResponse,K_c,2);
f5=subs(FreqResponse,K_c,5);
fplot([f1 f2 f5],[0 30]), grid on, xlabel("frequency"),
ylabel("Noise amplification gain")
legend("Kc=1","Kc=2","Kc=5")
```



NOTA: este es un gráfico NO logarítmico; normalmente se utilizará el diagrama de Bode, que tiene escalas logarítmicas tanto en el eje de "frecuencia" como en el de "ganancia de amplitud", pero por simplificar el código lo dejamos así.