PID control, UNSTABLE G(s)=1/s^2

Objective: understand the behaviour of **proportional-integral-derivative** control (simulations in companion videos) on a double-integrator system.

Videos:

ENGLISH

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

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

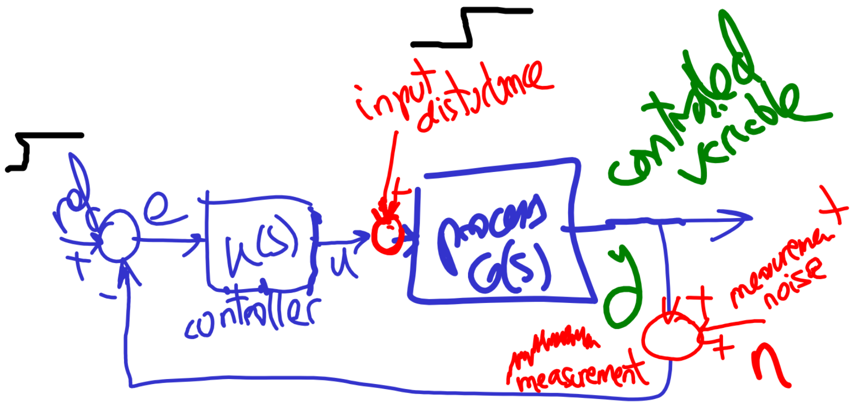
SPANISH

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

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

Closed-loop transfer functions

In a closed-loop:



The equations are:

```
syms G  K K_p K_d r u y n du e

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

We will be interested in output (**controlled** variable) trajectories:

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

ans =

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

Or, well, error will be $r - y$, but, anyway, we can display it:

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

ans =

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

And, later on, in **manipulated** variable trajectories:

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

ans =

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

Setpoint change response

```
syms s %Laplace variable
G=2/s^2
```

G =

$$\frac{2}{s^2}$$

```
%K=Kc*(1+Td*s)
%Kp=Kc, Kd=Kc*Td
K=K_p+K_d*s
```

K = $K_p + K_d\,s$

```
G*K/(1+G*K)
```

ans =

$$\frac{2\,(K_p + K_d\,s)}{s^2\,\left(\frac{2\,(K_p+K_d\,s)}{s^2}+1\right)}$$

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

CLrefSYM(s) =

$$\frac{(2\,K_d)\,s + 2\,K_p}{s^2 + (2\,K_d)\,s + 2\,K_p}$$

```
ErrFromref=collect((1-CLrefSYM),s)
```

ErrFromref(s) =

$$\frac{s^2}{s^2 + (2\,K_d)\,s + 2\,K_p}$$

Disturbance input response

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

```
CLduSYM(s) =
```

$$\frac{2}{s^2 + (2\,K_d)\,s + 2\,K_p}$$

Dynamics: We have "total controllability", i.e., capability of assigning the closed-loop poles at any arbitrary position by tweaking K_c and K_i. This is a standard "exam question" in starting courses in Control Theory. Say, we wish a time constant of 0.5 (poles at -2), then

```
[N,D]=numden(ErrFromref);
collect(D,s) %denominator is the same for any chosen output variable
```

`ans(s) =` $s^2 + (2\,K_d)\,s + 2\,K_p$

```
cD=coeffs(D,s)
```

`cD(s) =` $\begin{pmatrix} 2\,K_p & 2\,K_d & 1 \end{pmatrix}$

```
DesiredDenominator=(s+5)^2
```

`DesiredDenominator =` $(s+5)^2$

```
cDD=coeffs(DesiredDenominator,s)
```

`cDD =` $\begin{pmatrix} 25 & 10 & 1 \end{pmatrix}$

```
solve(cD==cDD,[K_p, K_d])
```

```
ans = struct with fields:
    K_p: 25/2
    K_d: 5
```

In reverse, for a given controller, the closed-loop poles are at:

```
solve(D==0,s) %closed-loop poles
```

```
ans =
```

$$\begin{pmatrix} \sqrt{{K_d}^2 - 2\,K_p} - K_d \\ -K_d - \sqrt{{K_d}^2 - 2\,K_p} \end{pmatrix}$$

This starts getting too complicated... We will switch to Control systems Toolbox.

```
s=tf('s');
G=2/s^2;
```

1. Small T_d 2**. Medium** T_d**, "cancellation" controller, 3. Large** T_d.

```
K_cmax=140;
T_d=0.125; %case 1
%T_d=0.5;%case 2
```

```
%T_d=10/11;%case 3
%T_d=0 %only PROPORTINAL
K1=1*(1+T_d*s);
K_c_percent=0.75; Kc=K_c_percent*K_cmax
```

```
Kc = 105
```

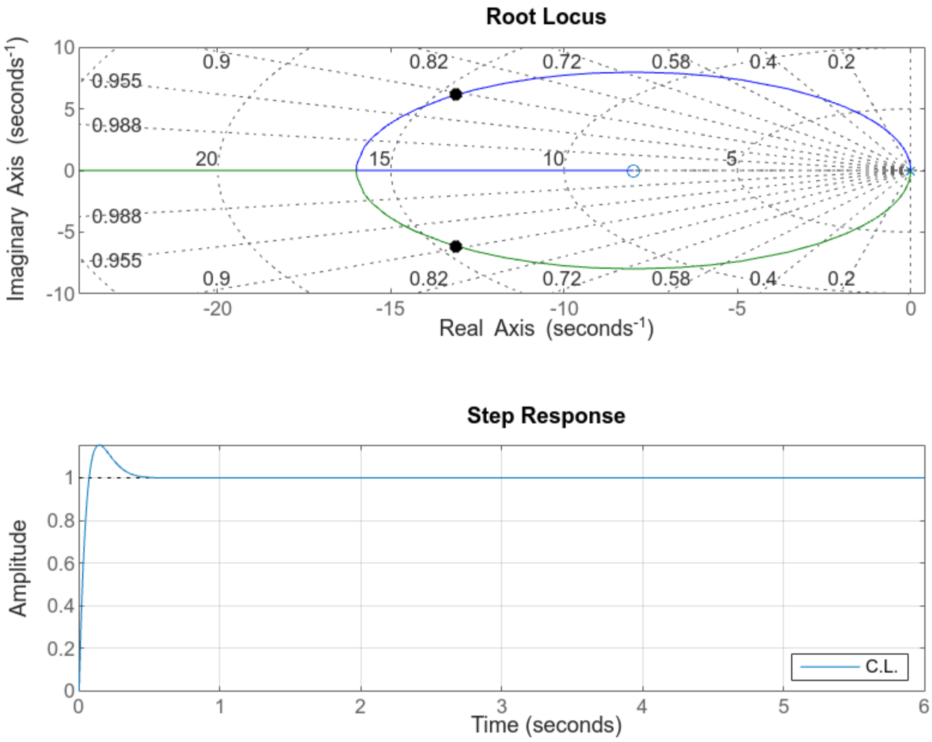
```
r=rlocus(G*K1,Kc)'
```

```
r = 1×2 complex
 -13.1250 - 6.1428i -13.1250 + 6.1428i
```

```
settlingtimeapprox=-4./real(r)
```

```
settlingtimeapprox = 1×2
    0.3048    0.3048
```

```
figure(), subplot(2,1,1), rlocus(G*K1), hold on,
plot(real(r),imag(r),'*k',LineWidth=3)
hold off, axis equal, xlim([-3/T_d 0.4]), grid on, subplot(2,1,2)
CLref=feedback(G*K1*Kc,1);
step(CLref,6), grid on,legend("C.L.",Location="southeast")
```




```
figure()
CLuref=feedback(K1*Kc,G)
```

```
CLuref =

  13.12 s^3 + 105 s^2
  -------------------
  s^2 + 26.25 s + 210

Continuous-time transfer function.
Model Properties
```

No realizable, nunca funcionará en la práctica (necesitamos filtro de ruido).

```
figure()
CLdu=feedback(G,K1*Kc);
step(CLdu,6), grid on, legend("C.L.",Location="southeast")
```