

Ejemplo de discretización de reguladores

© 2019, Antonio Sala Piqueras, Universitat Politècnica de València. Todos los derechos reservados.

Presentación en vídeo en: <http://personales.upv.es/asala/YTV/dreml.html>

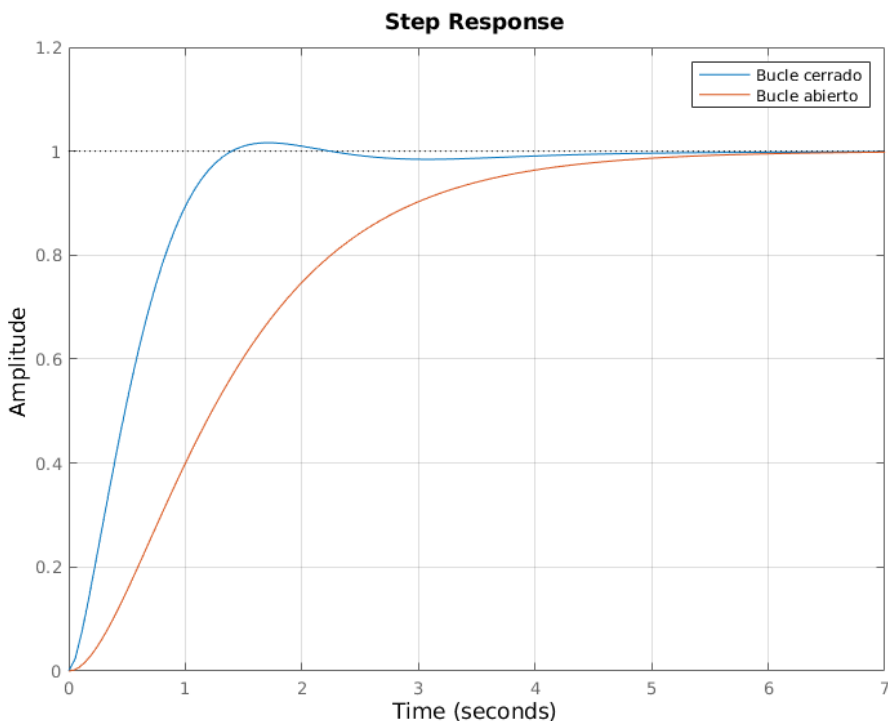
Este código ejecutó correctamente en Matlab R2018b.

Tabla de Contenidos

Modelado, diseño continuo a imitar.....	1
Criterios de ancho de banda para selección de T.....	2
Selección del período... aproximación de la pérdida de prestaciones con cero FNM.....	2
Discretización y estabilidad en bucle cerrado discreto.....	3
Simulación respuesta temporal.....	4
todos juntos:.....	8
Apéndice: Funciones complementarias.....	10

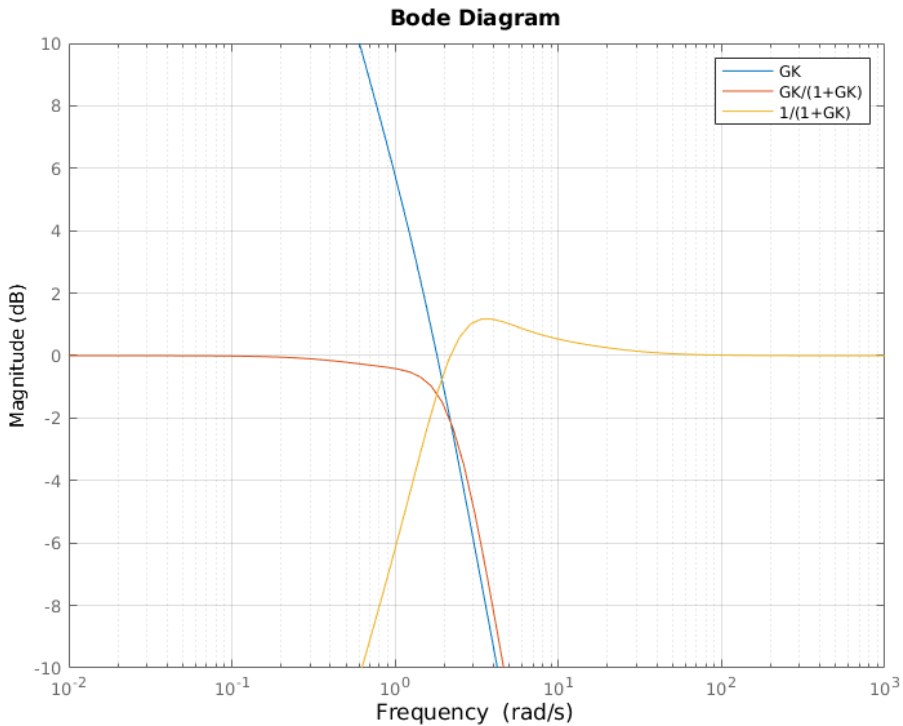
Modelado, diseño continuo a imitar.

```
s=tf('s');  
G=3/(s+1)/(s+2);  
K=@(ese) 1.8+1.2/ese+0.25*ese/(0.05*ese+1);  
Kc=K(s); %continuo  
CL=minreal(feedback(G*Kc,1)); %bucle continuo "ideal" a imitar.  
step(CL,G/dcgain(G)), grid on  
legend('Bucle cerrado','Bucle abierto')
```



Criterios de ancho de banda para selección de T

```
bodemag(G*Kc,CL,1-CL), grid on, legend('GK', 'GK/(1+GK)', '1/(1+GK)')
axis([-0.5 2 -10 10])
```



Picos de resp. en frecuencia:

```
[nr,fr]=norm(CL,'inf')
```

```
nr = 1.0000
fr = 0
```

```
[nsr,fsr]=norm(1-CL,'inf')
```

```
nsr = 1.1464
fsr = 3.7447
```

Selección del período... aproximación de la pérdida de prestaciones con cero FNM

```
PruebasT=[pi/100    0.099    pi/(3.74*5)    pi/(1.7*5)    0.5    0.8    1.1]
```

```
PruebasT = 1x7
    0.0314    0.0990    0.1680    0.3696    0.5000    0.8000    1.1000
```

```
%pi/100, frecuencia muestreo "muy alta"
```

```
%el segundo, 1/10 veces tiempo subida, 1/20 tiempo establec... tope
%estabilidad KdEuler con Euler Forward
```

```
%el tercero, 5 veces el pico de sensibilidad (ref->error)

%el cuarto, 5 veces el crossover de G*K

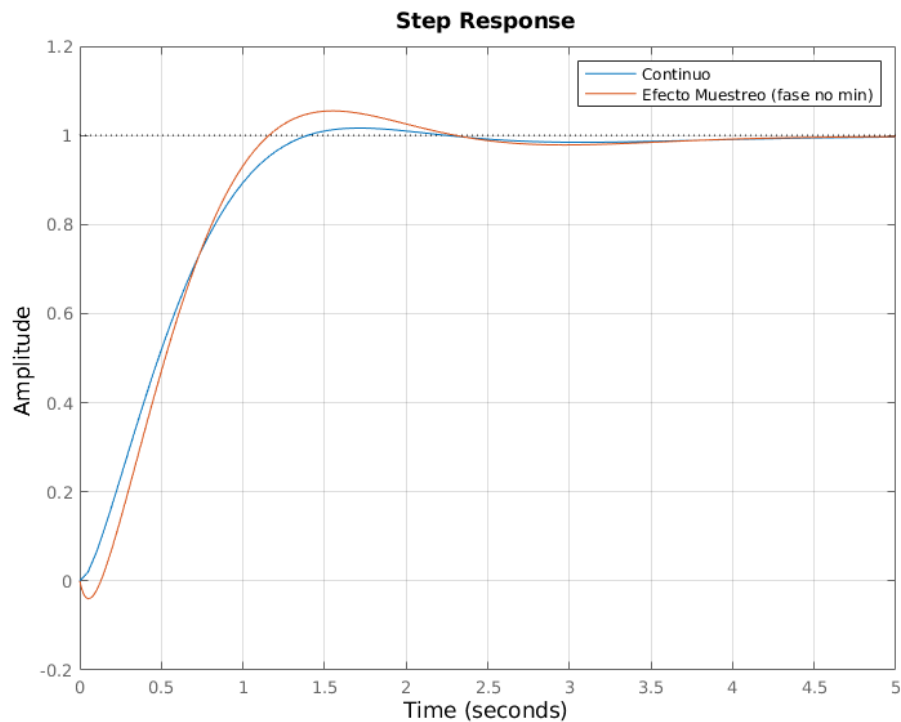
%el resto, períodos "grandes"...

T=PruebasT(3)
```

```
T = 0.1680
```

```
CLefectoZOH=minreal(feedback(G*(1-T*s/2)*Kc,1)); %predicción del efecto del muestreo...

tfinal=5;
step(CL,CLefectoZOH,tfinal), grid on, legend('Continuo','Efecto Muestreo (fase no min)')
```



Discretización y estabilidad en bucle cerrado discreto

```
z=tf('z',T); Gd=c2d(G,T,'zoh'); %discretización exacta del proceso
KdEulerForw=minreal(K*((z-1)/T)); %Forward
abs(eig(KdEulerForw))
```

```
ans = 2×1
    2.3600
    1.0000
```

```
KdEulerBack=minreal(K*((z-1)/T/z)); %Backward
abs(eig(KdEulerBack))
```

```
ans = 2×1
    1.0000
    0.2294
```

```
KdTustin=minreal(K(2*(z-1)/T/(z+1)));
Kdfoh=c2d(Kc,T,'foh');

CLTustin=feedback(Gd*KdTustin,1);
abs(eig(CLTustin))
```

```
ans = 4×1
    0.8912
    0.7809
    0.7809
    0.2020
```

```
CLEulerB=feedback(Gd*KdEulerBack,1);
abs(eig(CLEulerB))
```

```
ans = 4×1
    0.8983
    0.7599
    0.7599
    0.3583
```

```
CLEulerF=feedback(Gd*KdEulerForw,1);
abs(eig(CLEulerF))
```

```
ans = 4×1
    2.4563
    0.8829
    0.8002
    0.8002
```

```
CLfoh=feedback(Gd*Kdfoh,1);
abs(eig(CLfoh))
```

```
ans = 4×1
    0.8908
    0.7705
    0.7705
    0.1254
```

Simulación respuesta temporal

```
[yc0,t0]=step(CL,tfinal); %respuesta continua nominal
[ycz,tz]=step(CLefectoZOH,tfinal); %respuesta continua nominal con efecto ZOH
```

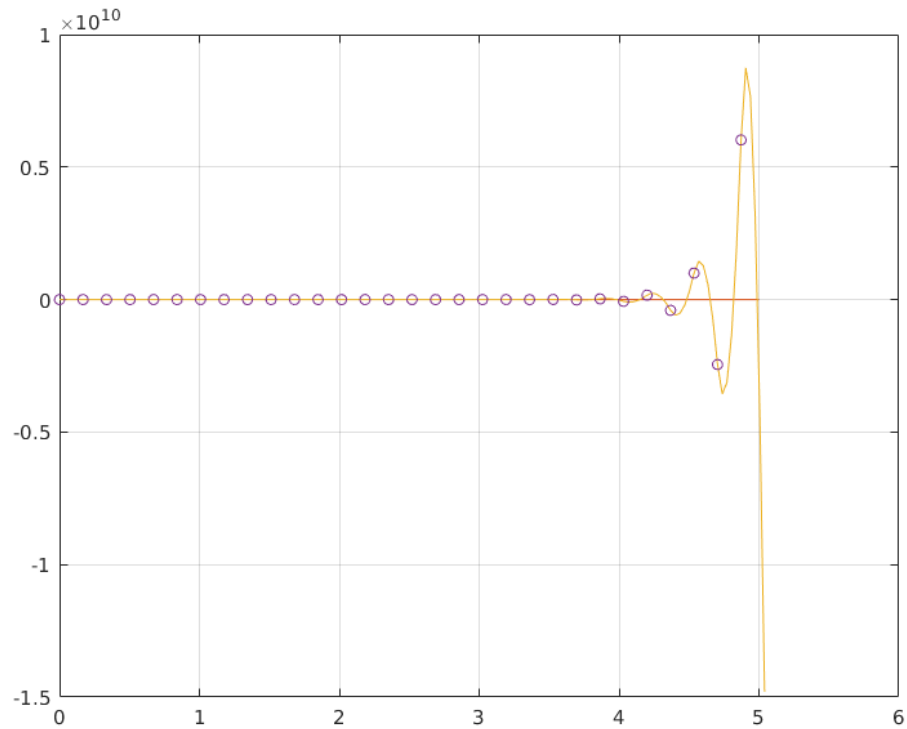
Euler hacia adelante:

```
plot(t0,yc0)
hold on
```

```

plot(tz,ycz)
%simulemos con respuesta intermuestreo, sdlsim:
[ycontE,ydE,tdE]=step_intermuestreo(G,KdEulerForw,tfinal);
plot(ycontE{:})
plot(tdE,ydE,'o'), grid on
hold off

```

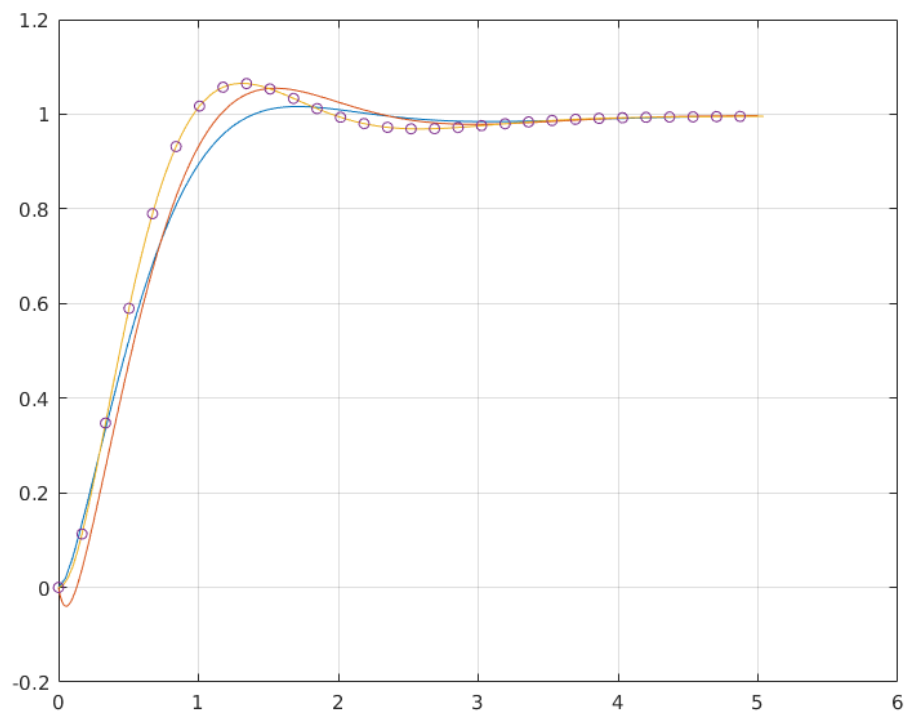


Euler hacia atrás:

```

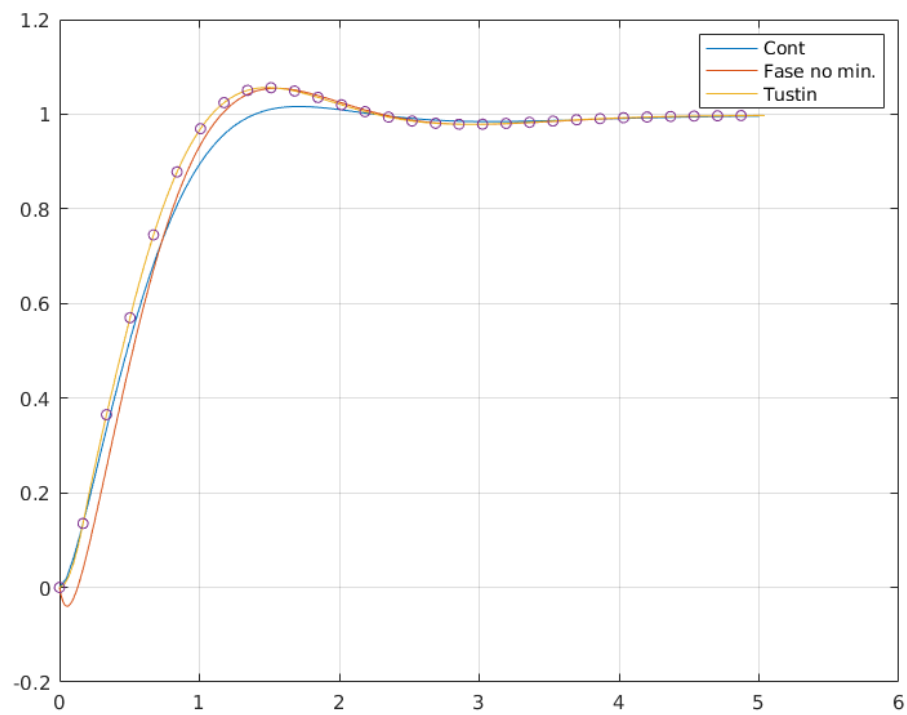
plot(t0,yc0)
hold on
plot(tz,ycz)
%simulemos con respuesta intermuestreo, sdlsim:
[ycontE,ydE,tdE]=step_intermuestreo(G,KdEulerBack,tfinal);
plot(ycontE{:})
plot(tdE,ydE,'o'), grid on
hold off

```



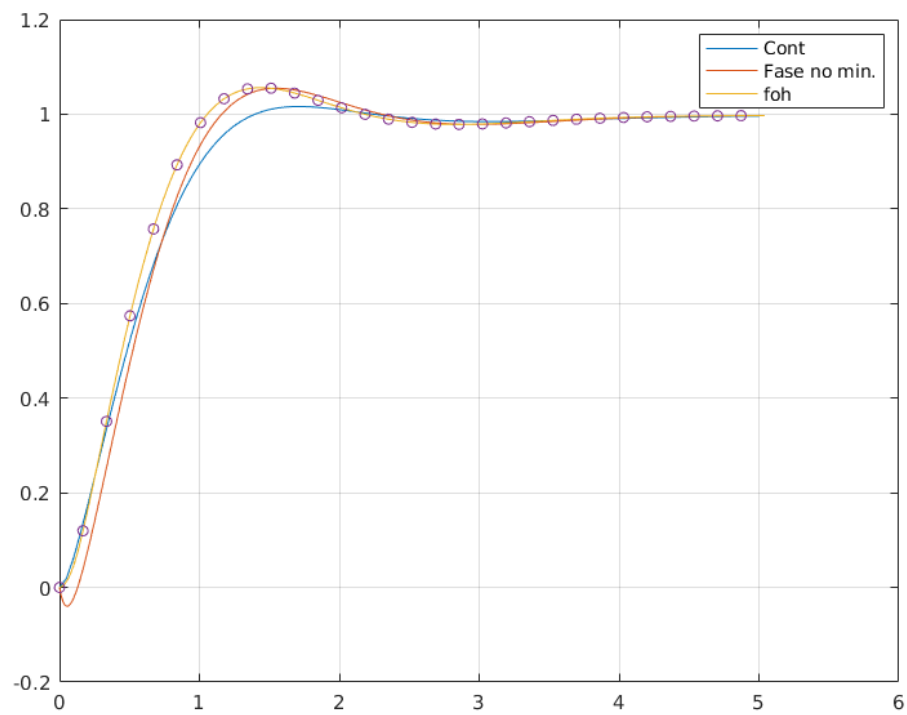
Bilinear (Tustin):

```
plot(t0,yc0)
hold on
plot(tz,ycz)
[ycontT,ydT,tdT]=step_intermuestreo(G,KdTustin,tfinal);
plot(ycontT{:})
plot(tdT,ydT,'o'), grid on
hold off
legend('Cont','Fase no min.','Tustin')
```



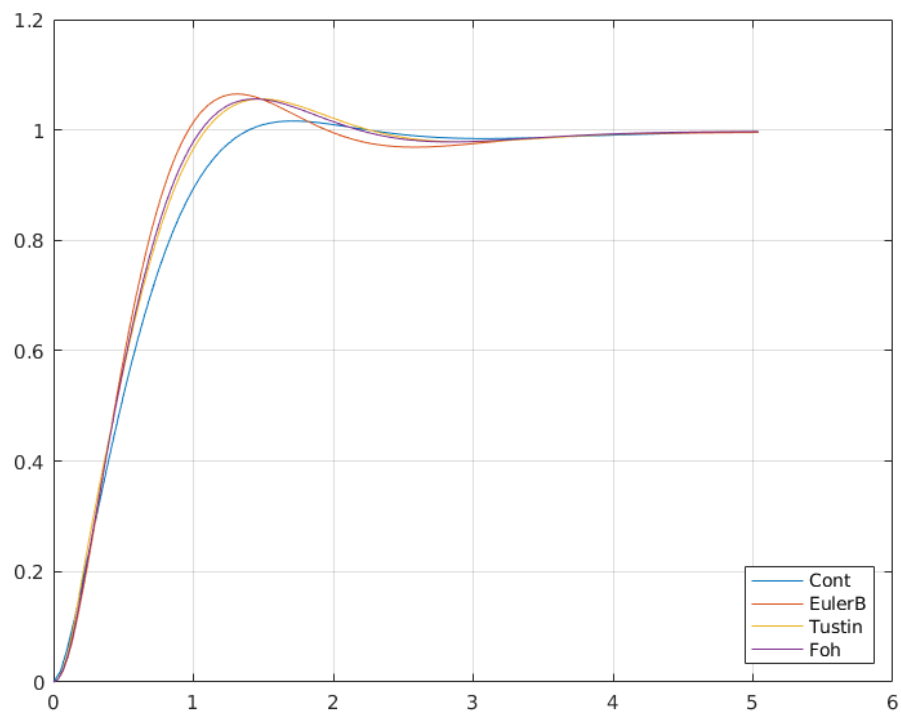
FOH:

```
plot(t0,yc0)
hold on
plot(tz,ycz)
[ycontF,ydF,tdF]=step_intermuestreo(G,Kdfoh,tfinal);
plot(ycontF{:})
plot(tdF,ydF,'o'), grid on
hold off
legend('Cont','Fase no min.','foh')
```

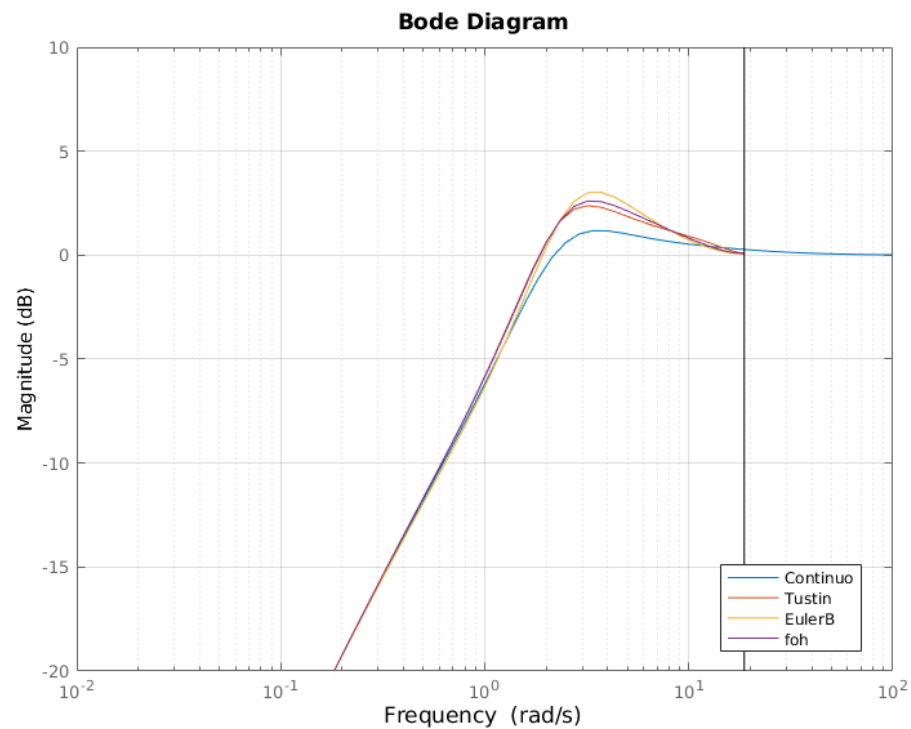


todos juntos:

```
plot(t0,yc0,ycontE{:},ycontT{:},ycontF{:}), grid on  
legend('Cont','EulerB','Tustin','Foh','location','best')
```



```
bodemag(1-CL,1-CLTustin,1-CLEulerB,1-CLfoh), grid on
axis([-2 3 -20 10])
legend('Contينو', 'Tustin', 'EulerB', 'foh', 'Location', 'Best')
```



Apéndice: Funciones complementarias

```
function [ycont,yd,td]=step_intermuestreo(G,Kd,tiempo_final)
    Ts=Kd.Ts;
    tiempos=[0:Ts:tiempo_final]'; ref=ones(size(tiempos));
    PPP=minreal(ss([0 G; 1 -G]));
    [ycont,~,~,~]=sdlsim(PPP,Kd,ref,tiempos,tiempo_final);%intermuestreo
    Gd=c2d(G,Ts,'zoh');
    [yd,td]=step(feedback(Gd*Kd,1),tiempo_final); %en muestras, discreto
end
```