

Diseño de reguladores discretos por "continuización" Tustin

© 2018, Antonio Sala Piqueras, Todos los derechos reservados.

Este código ha ejecutado sin errores en Matlab R2018a

Presentación en vídeo en <http://personales.upv.es/asala/YT/V/d2cml.html>

Tabla de Contenidos

Objetivo.....	1
Modelado y discretización exacta de planta a controlar.....	1
Continuización por Tustin para diseño continuo de un PID.....	1
diseñamos regulador continuo:.....	2
Discretización del regulador y simulación discreta (con la discretización exacta del proceso).....	3
Simulación de la respuesta intermuestreo:.....	4

Objetivo

Comprender el diseño por continuización y sus limitaciones.

Modelado y discretización exacta de planta a controlar

```
s=tf('s');  
Gcorig=5/(s^2+2*s+6);  
%Ts=0.1;  
%Ts=0.4;  
Ts=0.48;  
  
Gd=c2d(Gcorig,Ts,'zoh')
```

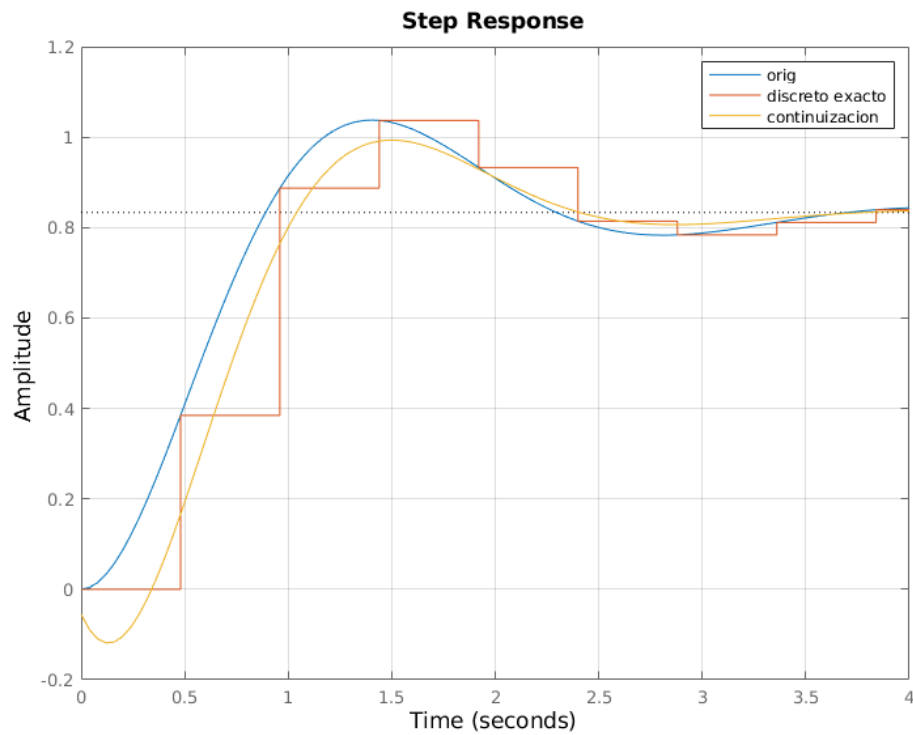
Gd =

$$\frac{0.3846 z + 0.2757}{z^2 - 0.5906 z + 0.3829}$$

Sample time: 0.48 seconds
Discrete-time transfer function.

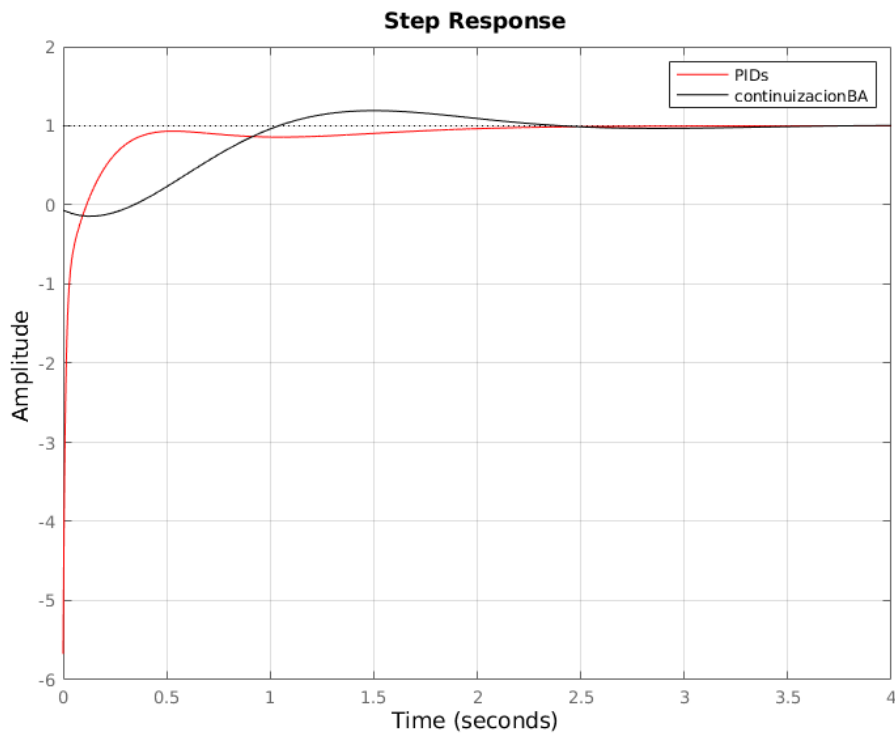
Continuización por Tustin para diseño continuo de un PID

```
Gcparadiseno=d2c(Gd,'tustin');  
step(Gcorig,Gd,Gcparadiseno,4), grid on  
legend('orig','discreto exacto','continuizacion')
```



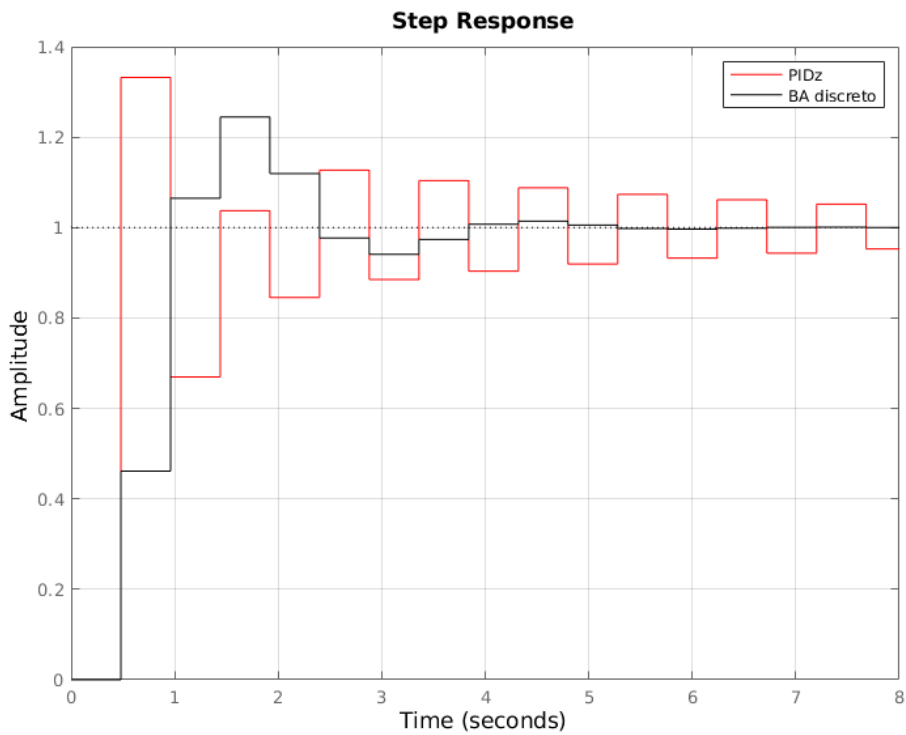
diseñamos regulador continuo:

```
PIDs=1.4*(1+1.7/s+0.33*s)*1/(0.03*s+1);
BCparadiseno=feedback(Gcparadiseno*PIDs,1);
step(BCparadiseno,'r',Gcparadiseno/dcgain(Gcparadiseno),'k',4), grid on
legend('PIDs','continuizacionBA')
```



Discretización del regulador y simulación discreta (con la discretización exacta del proceso)

```
PIDz=c2d(PIDs,Ts,'tustin');
BCz=feedback(Gd*PIDz,1);
step(BCz,'r',Gd/dcgain(Gd),'k',8), grid on
legend('PIDz','BA discreto')
```



Simulación de la respuesta intermuestreo:

```

PPP=minreal(ss([0 Gcorig;1 -Gcorig]));
t=[0:0.1:8]';r=ones(size(t));
sdlsim(PPP,PIDz,r,t,8)
hold on
[yd,td]=step(BCz,8);
plot(td,yd,'r*')
step(Gcorig/dcgain(Gcorig),'k')
grid on
hold off
legend('Control digital','Muestras','EscalonBA')

```

