# Práctica 5. Control digital

**Asignatura:** Sistemas Electrónicos de Control
**Curso:** 2013/2014-1
**Realización:** D4-211, 24/5/13 (g12) y 31/5/13 (g19), 18h-20h

**Nota:** Para la realización de la práctica es imprescindible traer el **estudio previo** hecho individualmente. El estudio previo consiste en resolver los ejercicios marcados con el símbolo ✐ y se recogerá a final de la sesión.

## 1. Discretización

**Ejercicio 1. Diseño analógico. Corrector de avance.** Considerar la planta

$$G(s) = \frac{10}{s^3 + 7s^2 + 6s}$$

Se pide:

✐ **Estudio Previo:**
Diseñar un corrector de avance de fase tal que el servo resultante presente un margen de fase superior a 45º. Indicar claramente el procedimiento que se ha seguido para su diseño.

**Simulación:**

Suponer que el corrector de avance diseñado es $C(s) = 1.5\dfrac{s+1}{s+3}$. Se pide:

**1)** Representar el diagrama de Bode del lazo $L(s) = G(s)C(s)$ y obtener el margen de fase (MF) (unciones **tf**, **margin**).
**2)** Obtener la función de transferencia del servo $M(s)$ (función **feedback**).
**3)** Representar la respuesta indicial del servo (función **step**).

**Ejercicio 2. Discretización de la planta.** Considerar de nuevo la planta $G(s) = \dfrac{10}{s^3 + 7s^2 + 6s}$ y el controlador $C(s) = 1.5\dfrac{s+1}{s+3}$. Se pide:

✐ **Estudio Previo:**
**1)** Discretizar la planta anteponiéndole un ZOH. Indicar claramente los pasos seguidos. Tomar como periodo de muestreo $T = 0.1$s.
**2)** Obtener el controlador digital equivalente $D_1(z)$ aplicando la transformada de Tustin $s = \dfrac{2}{T}\dfrac{z-1}{z+1}$. Tomar como periodo de muestreo $T = 0.1$s.

**Simulación:**

**1)** Comprobar los resultados del estudio previo con ayuda de la función **c2d**.

**2)** Obtener la función de transferencia del servo discreto $M(z)$ y representar su respuesta indicial (funciones **feedback**, **step**).

**3)** Representar el diagrama de polos y ceros de $M(z)$ (función **pzmap**). ¿Es de fase mínima?

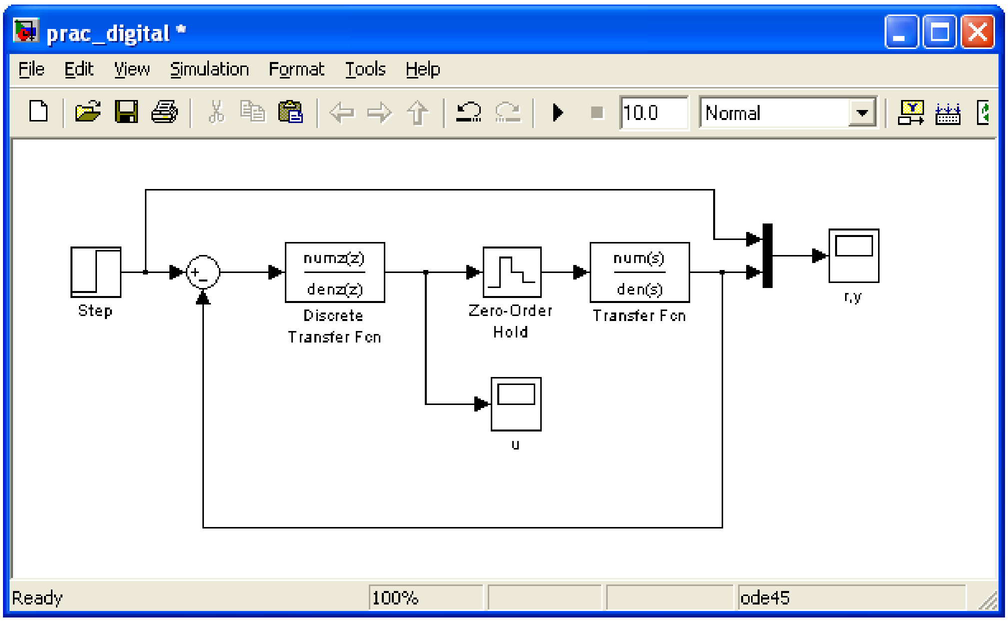
**4)** Obtener la función de transferencia en lazo cerrado $M_u(z)$ que relaciona la entrada de referencia ($r$) con el esfuerzo de control (salida del controlador, $u$) y representar esta última señal cuando la referencia es un escalón unitario (funciones **feedback**, **step**).

**Ejercicio 3. Efecto del periodo de muestreo.** Considerar de nuevo la planta $G(s) = \dfrac{10}{s^3 + 7s^2 + 6s}$ y el controlador $C(s) = 1.5\dfrac{s+1}{s+3}$. Se pide:

**Simulación:**

**1)** Obtener el numerador y el denominador del controlador digital equivalente aplicando la transformación bilineal (funciones **c2d, tfdata**).

**2)** Implementar el siguiente modelo Simulink del servo:




**3)** Realizar la simulación y representar la salida del servo y el esfuerzo de control (salida del controlador). Comentar el resultado.

**4)** Tomar ahora como periodo de muestreo $T$ = 0.5s y 0.01s y calcular de nuevo el controlador. Realizar de nuevo la simulación. Comentar el resultado.

**Ejercicio 4. Métodos de discretización.** Considerar de nuevo la planta $G(s) = \dfrac{10}{s^3 + 7s^2 + 6s}$ y el controlador $C(s) = 1.5\dfrac{s+1}{s+3}$. Se pide:

**Simulación:**
Tomando como periodo de muestreo T=0.1s,

**1)** Obtener el controlador $D_2(z)$ discretizando mediante el método de diferencias regresivas $s = \frac{1}{T}\frac{z-1}{z}$ (funciones **ss, bilin**) y simular la respuesta del servo y el esfuerzo de control mediante el modelo Simulink del ejercicio anterior.

**2)** Ídem mediante la transformación de polos y ceros $z = e^{-sT}$ con igual ganancia de continua. Nota: La ganancia en continua es $C(0) = D_3(1)$. (Función **c2d**)

**Ejercicio 5. Efecto de la saturación y el *slew rate*.** Considerar de nuevo la planta $G(s) = \dfrac{10}{s^3 + 7s^2 + 6s}$ y el controlador $C(s) = 1.5\dfrac{s+1}{s+3}$ (con discretización Tustin a 0.1s). Se pide:

**Simulación:**

**1)** Efecto de la saturación: Añadir un bloque “Saturation” para que tenga en cuenta restricciones en la amplitud de salida del controlador *u*. Tomar $|u|_{max} = 0.2$. Representar de nuevo el esfuerzo de control y la respuesta del servo y comentar el resultado.

**2)** Efecto del *slew rate*: Añadir un bloque “Rate Limiter Dynamic” para que tenga en cuenta la limitación en la velocidad de cambio (*slew rate*) $\Delta u$ de la salida del controlador. Tomar $|\Delta u|_{max} = 0.1$. Representar de nuevo el esfuerzo de control y la respuesta del servo y comentar el resultado.

## 2. Discretización de PIDs

El regulador de 3 términos o PID es el controlador industrial más utilizado y se usa a menudo en su versión digital. Se puede discretizar por diversos métodos. En esta práctica veremos algunos de ellos.

Considerar la planta de la Práctica 3,

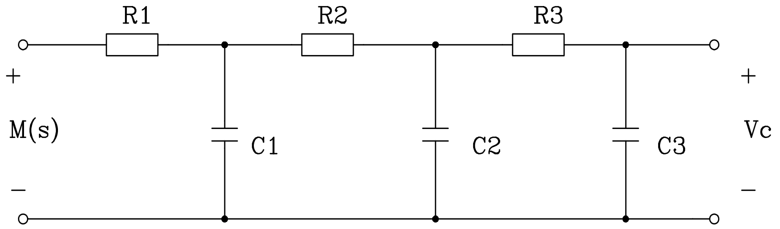



**Fig. 1. Planta**

Datos: R1=R2=1kΩ, R3=10kΩ, C1=1nF, C2=100nF, C3=22nF.

cuya función de transferencia es $G(s)=\dfrac{V_c(s)}{M(s)}=\dfrac{4545{,}45\cdot 10^{10}}{(s+2.005\cdot 10^{6})(s+7513)(s+3018)}$.

La respuesta indicial de la planta puede aproximarse por la respuesta indicial de un sistema equivalente de primer orden con retardo puro

$$\frac{k}{\tau s+1}\exp(-\tau_0 s)$$

con parámetros $k=1$, $\tau=0.4\cdot 10^{-3}$ y $\tau_0=0.08\cdot 10^{-3}$.

Estos tres parámetros permiten sintonizar reguladores PID según Ziegler-Nichols. En concreto,

- el regulador P vendrá determinado por $C(s)=k_p$ donde $k_p=\tau/(k\tau_0)$,
- el regulador PI por $C(s)=k_p\left(1+\dfrac{1}{T_i s}\right)$ donde $k_p=0.9\cdot\tau/(k\tau_0)$, $T_i=3.3\cdot\tau_0$, y
- el regulador PID por $C(s)=k_p\left(1+\dfrac{1}{T_i s}+T_d s\right)$ donde $k_p=1.2\cdot\tau/(k\tau_0)$, $T_i=2\cdot\tau_0$, y $T_d=0.5\cdot\tau_0$.

**Ejercicio 6. Diseño de reguladores PID vía curva sigmoide.**
Se pide:
**1)** Representar, en una misma gráfica, la respuesta indicial de la planta y el sistema equivalente a fin de valorar la bondad de la aproximación (funciones **tf**, **set**, **step**).
**2)** Calcular la función de transferencia en lazo cerrado para cada uno de los tres reguladores, $M_P(s)$, $M_{PI}(s)$ y $M_{PID}(s)$ (función **feedback**).
**3)** Representar, en una misma gráfica, la respuesta indicial de cada uno de los sistemas en lazo cerrado en el intervalo 0-2.5ms. (funciones **linspace**, **step**).

**Ejercicio 7. Discretización de la planta.** Obtener la planta discreta equivalente por el método de invarianza indicial (i.e., anteponiéndole un ZOH). Tomar como periodo de muestreo $T_s=10^{-5}$. (Función **c2d**)

## 2.1 Discretización del PID vía transformada bilineal (Tustin)

**Ejercicio 8.**
Se pide:
**1)** Discretizar el regulador PID aplicando la transformada bilineal con $T_s=10^{-5}$. (Función **c2d**).

**2)** Combinar el controlador con la planta equivalente discreta del ejercicio anterior a fin de obtener la función de transferencia en lazo cerrado $M_{PID}(z)$ (función **feedback**).
**3)** Obtener la respuesta indicial del sistema en lazo cerrado en el intervalo 0-2.5ms. (funciones **linspace**, **step**).
**4)** Explicar el comportamiento anterior a la vista del diagrama de polos y ceros del sistema (función **pzmap**).

**Ejercicio 9. Efecto del periodo de muestro.** Se pide:
**1)** Obtener el equivalente discreto del controlador PID vía Tustin para los periodos de muestreo $T_s=10^{-5}$, $T_s=2\cdot10^{-6}$, $T_s=3\cdot10^{-5}$. Para cada uno de ellos obtener la función de transferencia en lazo cerrado.
**2)** Representar, en una misma gráfica, las respuestas indiciales. Comparar los resultados.

## 2.2 Discretización del PID acción a acción vía diferencias regresivas

Una alternativa a la transformada bilineal consiste en discretizar cada una de las acciones del PID aplicando la aproximación de la derivada en diferencias regresivas, $s = \left(1 - z^{-1}\right)/T_s$ .

**Ejercicio 10. Diferencias regresivas.**
Se pide:
**1)** Obtener el PID discreto vía diferencias regresivas. Tomar $T_s=5\cdot10^{-5}$s.
**2)** Obtener la función de transferencia del servo y representar su respuesta indicial.
**3)** Obtener la función de transferencia del esfuerzo de control y representar su respuesta indicial.

## 2.3 Discretización de D vía diferencias regresivas y I vía Simpson

Otra opción directa es discretizar cada una de las acciones de forma distinta. Para la acción derivativa podemos usar la aproximación de la derivada, $s = \left(1 - z^{-1}\right)/T_s$ . Para la acción integral se puede usar la regla trapezoidal o de Simpson.

Esta regla consiste en ir sumando las áreas del error: la integral en el instante $n$ es la integral en el instante $n$-1 (la llamaremos *PrevInt*) más el área que hay entre los instantes $n$ y $n$-1 y que podemos calcular como $\left(e_n + e_{n-1}\right)T_s/2$ . Así, la ecuación en diferencias del algoritmo PID queda como

$$u_n = k_p\left(e_n + \frac{T_d\left(e_n - e_{n-1}\right)}{T_s} + \frac{1}{T_i}\left[\frac{\left(e_n + e_{n-1}\right)T_s}{2} + PrevInt\right]\right)$$

o, lo que es lo mismo,

$$u_n = k_p\left(1+\frac{T_d}{T_s}+\frac{T_s}{2T_i}\right)e_n + k_p\left(-\frac{T_d}{T_s}+\frac{T_s}{2T_i}\right)e_{n-1}+\frac{k_p}{T_i}PrevInt$$

donde en cada intervalo hay que actualizar *PrevInt* → $PrevInt + \frac{(e_n + e_{n-1})T_s}{2}$.

**Ejercicio 11. Diferencias regresivas y regla trapezoidal.**
Se pide:
En un fichero m copiar el siguiente programa, que simula las ecuaciones en diferencias del PID digital y la planta discretizada.

```
Ts=5e-5;
t=0:Ts:2.5e-3;nmax=length(t);
a = Kp*(1 + Td/Ts + 0.5*Ts/Ti);
b = Kp*(-Td/Ts + 0.5*Ts/Ti);
c = Kp/Ti;
[numz,denz]=tfdata(Gz,'v');
integ=0; % integral previa
un=0;un_1=0; %u(n), u(n-1)
yn=0;yn_1=0;yn_2=0;  %y(n), y(n-1), y(n-2)
en=0; %e(n)
y(1)=0;
for n=1:nmax
    % error
    e=1-yn;
    % PID
    u(n) = a*e + b*en + c*integ;
    % planta
    y(n+1) = numz(2)*u(n) + numz(3)*un + numz(4)*un_1;
    y(n+1) = y(n+1) - denz(2)*yn - denz(3)*yn_1 - denz(4)*yn_2;
    % integral
    integ=integ + 0.5*Ts*(e+en);
    % actualiz
    en=e;
    un_2=un_1;un_1=un;un=u(n);
    yn_2=yn_1;yn_1=yn;yn=y(n+1);
end
ta=linspace(0,2.5e-3);
figure,step(Mpid,ta),hold on,stairs(t,y(1:nmax),'r'),
legend('analógico','digital'),
```

Ejecutar el fichero y comprobar su correcto funcionamiento.

## 2.4 PID incremental

Una versión del PID anterior mucho más utilizada consiste en calcular sólo los cambios de la señal de control con respecto a la muestra anterior. Este algoritmo recibe el nombre de *algoritmo de velocidad* o *PID incremental* y tiene diversas ventajas: no hay que guardar el valor absoluto de la integral hasta ese momento y se puede pasar de manera más suave del control manual al automático (*bumpless transfer*). Ello es así porque cuando el controlador ya está en marcha solo hay que calcular el primer movimiento incremental y no se debe inicializar para dar la misma señal absoluta $u_n$ que se tenía en control manual. Además, muchos actuadores, como los motores paso a paso, ya esperan entradas de control incrementales.

Para obtener el incremento de la señal de control basta con calcular $\Delta u = u_n - u_{n-1}$. Notar que el área añadida a la integral en el instante n es $\dfrac{(e_n + e_{n-1})T_s}{2}$ mientras que el área añadida en el instante n-1 es $\dfrac{(e_{n-1} + e_{n-2})T_s}{2}$ siendo $PrevInt_{n-1}$ la integral hasta el momento n-1. Así,

$$u_n = k_p\left(e_n + \frac{T_d(e_n - e_{n-1})}{T_s} + \frac{1}{T_i}\left[\frac{(e_n + e_{n-1})T_s}{2} + \underbrace{PrevInt_{n\text{-}1} + \frac{(e_{n-1} + e_{n-2})T_s}{2}}_{PrevInt}\right]\right)$$

$$u_{n-1} = k_p\left(e_{n-1} + \frac{T_d(e_{n-1} - e_{n-2})}{T_s} + \frac{1}{T_i}\left[PrevInt_{n\text{-}1} + \frac{(e_{n-1} + e_{n-2})T_s}{2}\right]\right)$$

Restando, el algoritmo incremental queda como:

$$\Delta u = u_n - u_{n-1} = k_p\left(e_n - e_{n-1} + \frac{T_d(e_n - 2e_{n-1} + e_{n-2})}{T_s} + \frac{1}{T_i}\left[\frac{(e_n + e_{n-1})T_s}{2}\right]\right)$$

**Ejercicio 12. PID incremental.**
Se pide:

**1)** Modificar el programa del ejercicio anterior a fin de poder simular el comportamiento de un PID incremental.

**2)** Ejecutar el programa y comprobar su correcto funcionamiento representando la señal de salida del servo y el esfuerzo de control.

## 3. Otros controladores digitales. Control dead beat

Para finalizar se presenta un tipo de control digital puro (no proveniente de ninguna discretización) con el objetivo de ilustrar el comportamiento entre muestras.

Una respuesta *dead beat* es la que sigue exactamente la excitación con una única muestra de retardo. Ello sólo puede conseguirse adecuadamente en sistemas de primer orden.

Suponer que la planta es $G(z)$. Si se excita con un escalón unitario, la respuesta deseada es $Y(z) = \dfrac{z}{z-1} z^{-1} = \dfrac{1}{z-1}$.

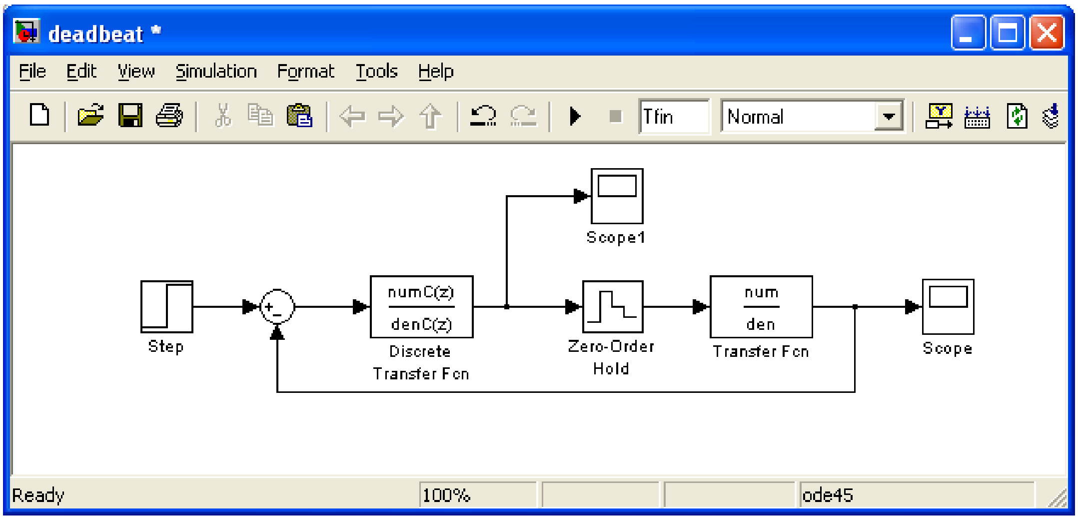
La función de transferencia en lazo cerrado será $T(z)=\dfrac{Y(z)}{R(z)}=\dfrac{1/(z-1)}{z/(z-1)}=\dfrac{1}{z}$.  Así, el controlador será $C(z)=\dfrac{1}{G(z)}\dfrac{T(z)}{1-T(z)}=\dfrac{1/z}{(1-1/z)G(z)}$.

**Ejercicio 13.  Comportamiento entre muestras.  Control *dead beat*.**
Se pide:

**1)** Combinar el controlador *dead beat* con la planta discretizada a fin de obtener la función de transferencia del servo y del esfuerzo de control.

**2)** Representar la respuesta indicial de ambas y comprobar que el sistema es capaz de seguir perfectamente la señal escalón con una muestra de retardo.

**3)** Construir ahora el siguiente modelo Simulink




**4)** Ejecutarlo y ver la forma de la señal (analógica) de salida de la planta.