Transformaciones para sistemas muestreados
El operador de desplazamiento (*q*) y la transformada Z

Para el diseño de controladores por emulación, se parte del diseño del controlador en continua para después discretizarlo y obtener la función de transferencia discreta para la implementación digital.

Cualquier sistema continuo lineal puede representarse por su función de transferencia (FT) en transformada de Laplace:

$$G(s) = \frac{Y(s)}{X(s)}$$

Alternativamente puede representarse mediante una ecuación diferencial o integral.

El objetivo de la discretización es aproximar las ecuaciones integrales y diferenciales para su resolución numérica. Debe recordarse que se trata de APROXIMACIONES al comportamiento continuo, y que distintas transformaciones pueden dar lugar a comportamientos ligeramente diferentes.

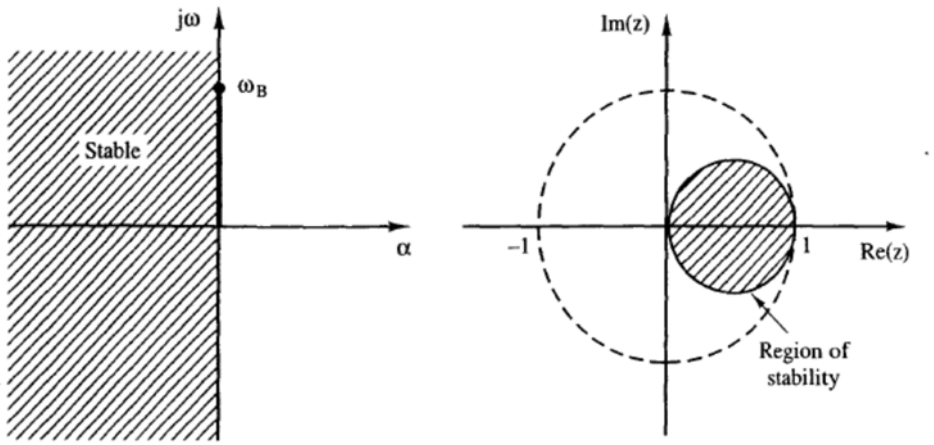
1. **Diferencias hacia atrás (*Backward difference*):** Sustituye la función derivada por

$$\frac{d}{dt}y(t) \triangleq \frac{y(t) - y(t - T_s)}{T_s}$$

   Así,

$$s \triangleq \frac{1 - z^{-1}}{T_s}$$

   Propiedades: Aplicada a la discretización de un filtro controlador, el semiplano izquierdo en s se traslada al círculo unidad en z. Así, la transformación de un controlador continuo estable siempre produce un controlador discreto estable (de hecho, algunos controladores inestables se vuelven estables). La desventaja está en la respuesta en frecuencia, ya que el eje jω del plano s no se traslada al círculo unidad en el plano z y dicha respuesta se degrada según nos alejamos de $s = 0$ ($z = 1$). La solución es aumentar la frecuencia de muestreo (naturalmente).



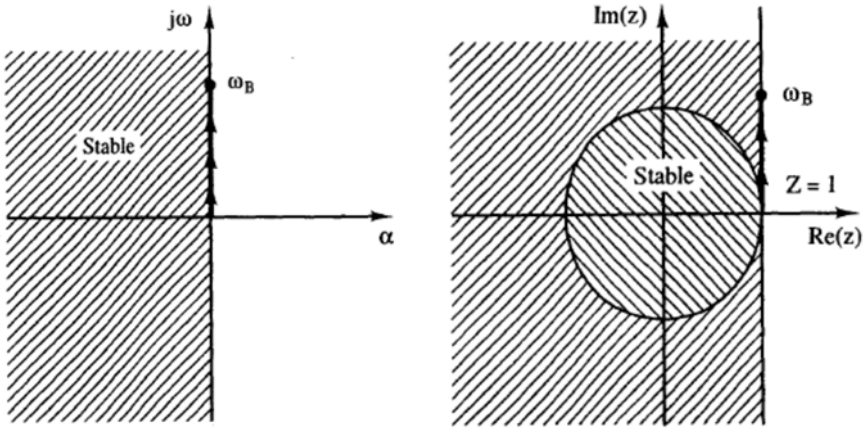
2. **Diferencias hacia adelante (*Forward difference*):** Sustituye la función derivada por

$$\frac{d}{dt}y(t) \triangleq \frac{y(t+T_s)-y(t)}{T_s}$$

Así,

$$s \triangleq \frac{z-1}{T_s}$$

Propiedades: Aplicada a la discretización de un filtro, el semiplano izquierdo en el dominio *s* se mapea al semiplano izquierdo en el dominio *z*, por lo que no se garantiza la estabilidad, Tampoco el contorno de frecuencia resultante se circunscribe al círculo unidad en el plano z. No es pues un buen método de discretización para controladores.



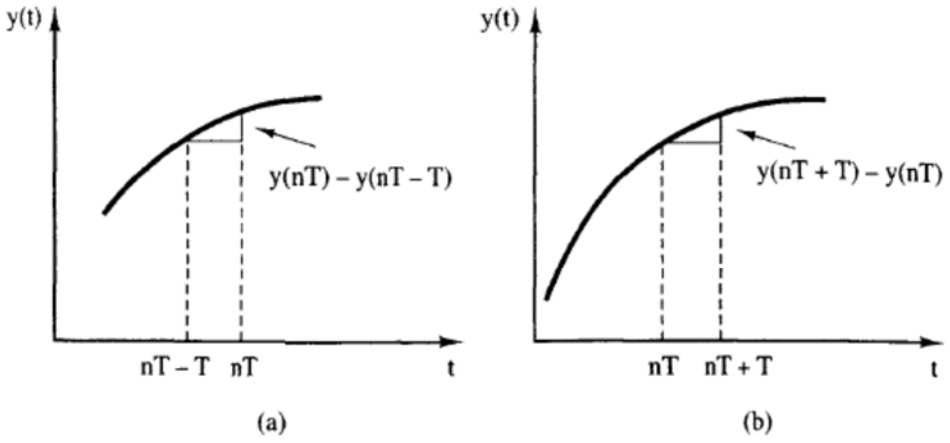


Figure 11-2 Difference approximations: (a) backward difference; (b) forward difference.

3. **Regla rectangular (izquierda y derecha):** Se utiliza para aproximar la integral (el integrador continuo es s^{-1})

$$y(t) = \int_0^t x(t)dt = \int_0^{nT_s} x(t)dt$$

Si aproximamos “a la izquierda”:

$$y(nT_s) = y(nT_s) + T_s x(nT_s)$$

$$\frac{1}{s} \triangleq \frac{T_s}{z-1}$$

Si aproximamos “a la derecha”:

$$y(nT_s) = y(nT_s - T_s) + T_s x(nT_s)$$

$$\frac{1}{s} \triangleq \frac{T_s}{1-z^{-1}}$$

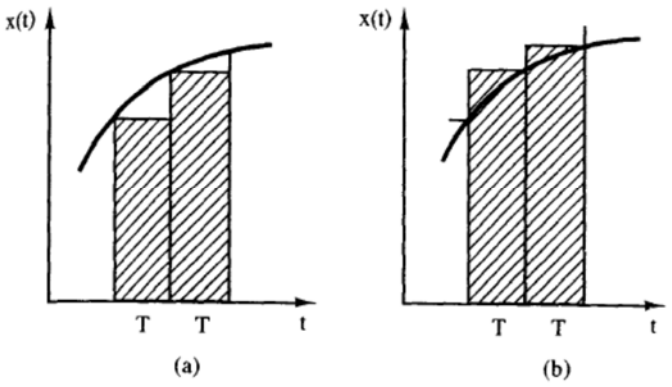


Figure 11-3 Rectangular rule: (a) left-side rule; (b) right-side rule.

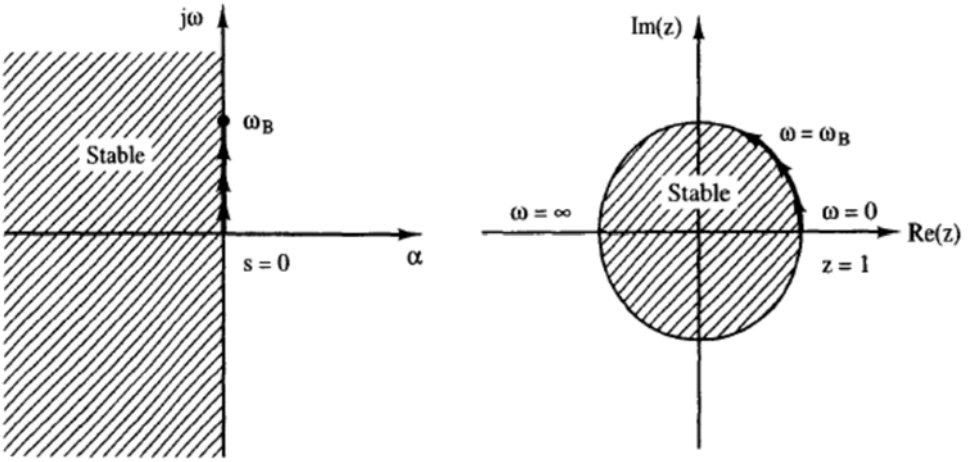
4. **Regla trapezoidal (de Tustin o bilinear):** Se toma el valor medio de las reglas rectangulares izquierda y derecha para aproximar el término integral. Así,

$$y(nT_s) = \frac{1}{2}\left[T_s \sum_{i=0}^{n-1} x(iT_s) + T_s \sum_{i=1}^{n} x(iT_s)\right]$$

Así,

$$\frac{1}{s} \triangleq \frac{T_s}{2} \frac{1+z^{-1}}{1-z^{-1}}$$

Propiedades: Aplicada a la discretización de un filtro, el semiplano izquierdo en *s* se mapea por completo en el círculo unidad en el dominio z. Todo controlador estable en s produce pues un controlador estable en *z*. Esta es la principal razón de que tradicionalmente se haya impuesto esta transformación en control discreto. También el eje $j\omega$ en el plano s se traslada al círculo unidad en z; sin embargo, todo el eje $j\omega$ se mapea precisamente en la circunferencia del círculo unidad, lo que produce un discordancia de frecuencias en la transformación (una especie de cambio de escala, de lineal a no lineal). Aun así, se asegura que no se produce *alliasing* en la frecuencia de la FT.



5. **Regla de Simpson:** Se evalúa la función integral por la siguiente expresión.

$$y(nT_s) = \frac{T_s}{3}\left[x(0) + 4x(T_s) + 2x(2T_s) + \ldots + 4x(nT_s - T_s) + x(nT_s)\right]$$

Así,

$$\frac{1}{s} \triangleq \frac{T_s}{3} \frac{1+4z^{-1}+z^{-2}1}{1-z^{-2}}$$

Propiedades: Produce filtros (controladores) estables, pero el inconveniente para implementación es que transforma las FTs de segundo orden en FTs de cuarto orden.

6. **Integrador invariante ante el impulso**[1]**:** Si se busca una discretización $D(z)$ de $G(s)$ tal que su respuesta impulsional sea igual a la respuesta impulsional en continua se obtiene que la transformada z estándar $s = (1/T_S).\ln(z)$ tiene esta propiedad. Por lo tanto, el integrador invariante ante el impulso (para T_S pequeñas) es

$$\frac{1}{s} \triangleq \frac{T_s}{1-z^{-1}}$$

Es decir, que **las aproximaciones por diferencias hacia atrás, la regla rectangular a la derecha, y el integrador invariante ante el impulso son transformaciones equivalentes, y todos producen una discretización que responde a la transformación *z* estándar.**

7. **Integrador invariante ante el escalón**[1]**:** Si se busca una discretización $D(z)$ de $G(s)$ tal que su respuesta ante una entrada escalón sea igual a la respuesta en continua se obtiene, para el integrador,

$$\frac{1}{s} \triangleq \frac{T_s z^{-1}}{1-z^{-1}}$$

8. **Emparejamiento de polos y ceros:** Asegura que se mantienen los mismos polos y ceros de la FT continua (*matched* z*)*. Los polos de la transformación son idénticos a la transformación *z* estándar, pero la de los ceros es diferente:

$$s+\alpha \rightarrow 1-z^{-1}e^{-\alpha T_s}$$

9. **Otras transformaciones:** Hay otras propuestas, como la que o la de Boxer and Thaler, que produce respuestas temporales iguales a las producidas por el sistema continuo. Sin embargo, son complejas desde el punto de vista de la implementación.

[1] En los instantes de muestreo

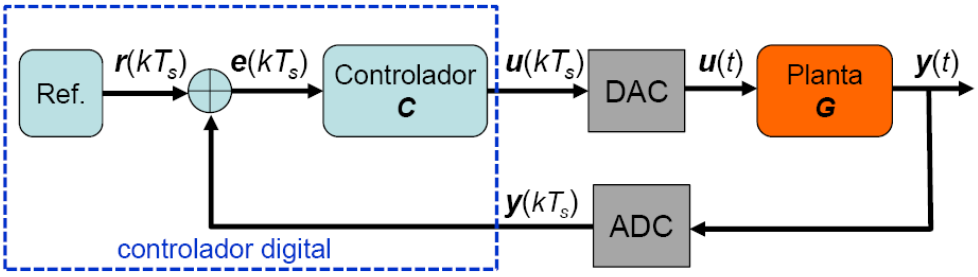
Ejercicio: Se trata de comprobar experimentalmente (simulación por Matlab/Simulink) los efectos que teóricamente producen las distintas formas de discretización en un compensador diseñado en el dominio continuo. Tomaremos como referencia en continua el modelo de un servomotor con la siguiente representación:

$$G(s) = \frac{52.1}{s(1.21s + 1)}$$

Se ha ajustado un compensador de tipo PID para controlarlo cuya FT es la siguiente:

$$C(s) = \frac{0.525s^2 + 5.022s + 4.4}{0.005s^2 + s}$$

El compensador se ha colocado por delante de la planta del motor con realimentación negativa.



1. Crea estas dos FT en el entorno de trabajo de Matlab (función ***tf***): *C* y *G*
2. Ahora puedes obtener la FT del sistema en lazo cerrado mediante los siguientes comandos:

```
C.InputName = 'e'; C.OutputName = 'u';
G.InputName = 'u'; G.OutputName = 'y';
Sum = sumblk('e', 'r', 'y' , '+-');
controlled = connect(G, C, Sum, 'r', 'y');
```

   Alternativamente puedes utilizar la función ***feedback*** ¿Cómo sería el código? Comprueba que los resultados son idénticos.

3. Utilizar las funciones de análisis del Control Toolbox de Matlab para analizar las características frecuenciales y temporales de este **controlador** continuo:

   Polos y ceros:

```
pzmap(C)
pole(C)
zero(C)
```

Análisis frecuencial
`Bode(C)`
`Nyquist(C)`

Si lo prefieres puedes usar la aplicación Linear System Analyzer (APPS > CONTROL SYSTEM DESIGN AND ANALYSIS > Linear System Analyzer), que te permitirá visualizar las respuestas combinadas de este sistema lineal de forma gráfica.

4. Ahora realiza una discretización de Tustin (bilinear) del controlador de la función ***c2d*** de Matlab (ver opciones de esta función en el *help* de Matlab). Debes realizar esta discretización para tres periodos de muestreo diferentes. Para ello, primero debes calcular BW_{cl}, el ancho de banda del sistema controlado en lazo cerrado (controlled).

   T_{s1}: T_s en el rango clásico recomendado ($1/T_{s1} \approx 10 \times BW_{cl}$).
   T_{s2}: Sobremuestreo. Aumenta la frecuencia en un orden de magnitud ($1/T_{s2} \approx 10/T_{s1} \approx 100 \times BW_{cl}$).

   Vuelve a analizar estos controladores (ahora discretos), empezando por las nuevas posiciones de ceros y polos.

5. Estudia cómo ha variado la estabilidad, si es que lo ha hecho, del sistema en lazo cerrado con el controlador discretizado con respecto al sistema continuo original. Para ello puedes simplemente determinar el margen de ganancia y de fase del sistema **en lazo abierto** (tan solo tienes que multiplicar C(s)×G(s); alternativamente puedes utilizar la función `series` de Matlab) para todos los casos analizados, es decir, compara el caso continuo con los dos casos discretizados para ambos periodos de muestreo. La función `allmargin` de Matlab te puede resultar útil para este análisis. También puedes explorar el uso de la función (`niquist`) .

6. Finalmente estudia la respuesta del sistema en lazo cerrado con los distintos controladores discretos que has creado (ambas frecuencias de muestreo). Puedes hacerlo en el entorno Matlab utilizando las FT que has creado en el punto 1 del ejercicio. En este caso obtendrás una respuesta en tiempo discreto. También puedes simular en Simulink utilizando el esquema que se adjunta (*simulink_discrete.mdl*). Esto te permitirá visualizar la respuesta del sistema en tiempo “continuo” (más realista). Analiza:

   - La respuesta temporal ante entrada escalón (ángulo de 0 a Pi).
   - La respuesta frecuencial del sistema (bode).

   Obtén gráficos comparativos y extrae conclusiones