MANUAL DE ESTUDIO · SISTEMAS Y CONTROL EN MECATRÓNICA

Modelado, señales, Laplace, transformada Z y espacio de estados

Capítulo 01 · En el video: **minuto 1:01** · Duración 10:22

Objetivo. Representar un sistema mecatrónico en tiempo continuo y discreto, por función de transferencia y en espacio de estados.

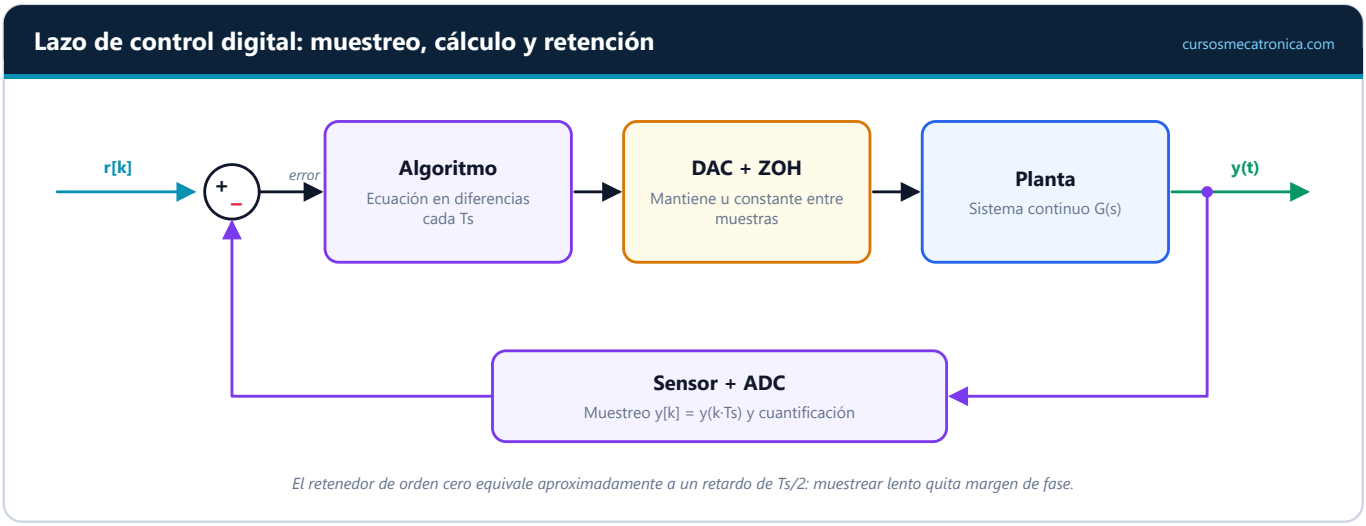
Ideas clave

Rol del control y del modelo

El control aporta precisión, ancho de banda y estabilidad; el modelo, orientado al control, captura solo la dinámica relevante y se valida con mediciones (ciclo en V, gemelos digitales).

Señales muestreadas

El ADC toma $x[k] = x(k \cdot T_s)$ y cuantifica con paso q; el DAC con retenedor de orden cero (ZOH) mantiene la salida constante entre muestras. El ruido de cuantificación es $q/\sqrt{12}$ rms.



Partes de un sistema de control muestreado.

Propiedades

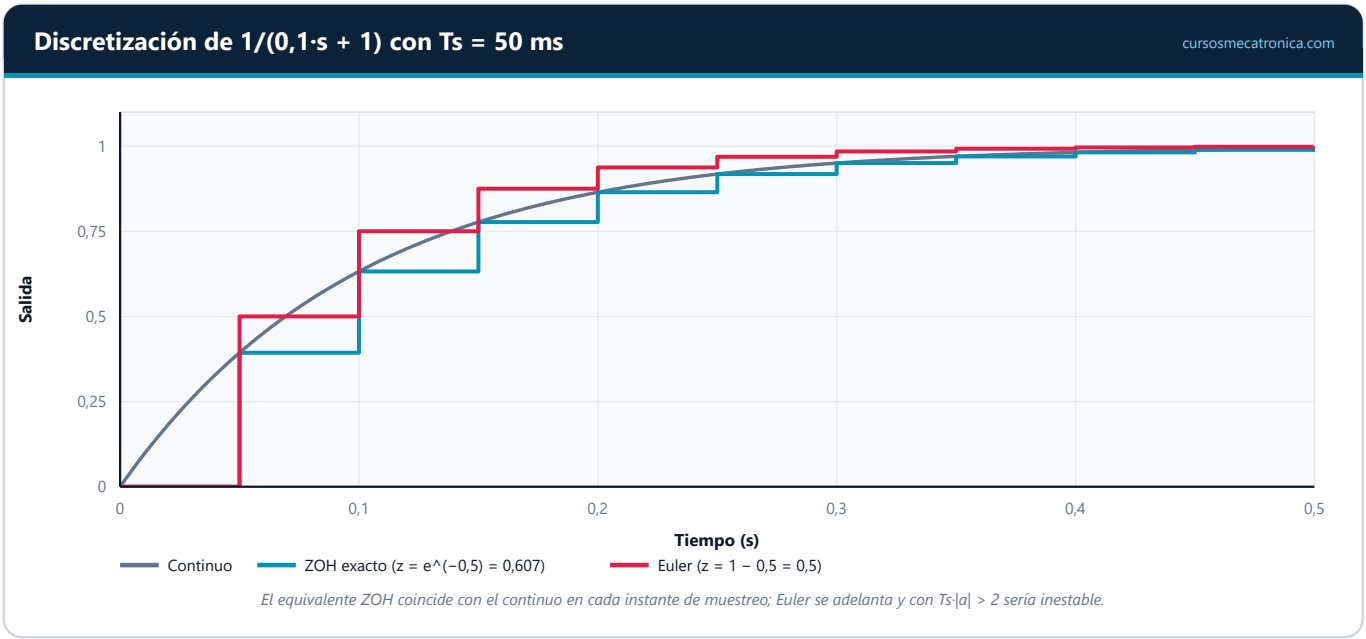
Linealidad, invarianza en el tiempo, causalidad y estabilidad BIBO (toda entrada acotada da salida acotada). La salida es la convolución de la entrada con la respuesta al impulso.

Laplace y transformada Z

G(s) describe sistemas continuos; G(z), con el retardo z^{-1}, describe ecuaciones en diferencias. Un sistema discreto es estable si sus polos están dentro del círculo unitario.

Discretización

Exacta con ZOH ($z = e^{sT_s}$ para los polos), Euler (simple, puede inestabilizar) o Tustin $s = (2/T_s)\cdot(z-1)/(z+1)$, que conserva la estabilidad; con prewarping se respeta una frecuencia crítica.



Equivalentes discretos de un primer orden.

Espacio de estados

$\dot{x} = Ax + Bu$, $y = Cx + Du$; en discreto $x[k+1] = \Phi x[k] + \Gamma u[k]$ con $\Phi = e^{AT_s}$. Representa naturalmente sistemas con varias entradas y salidas, y $G(s) = C(sI - A)^{-1}B + D$.

Fórmulas y relaciones

Relación	Expresión	Cómo leerla
Muestreo	$x[k] = x(k \cdot T_s)$	$f_s = 1/T_s$.
Ruido de cuantificación	$\sigma_q = q/\sqrt{12}$, $q = \text{rango}/2^n$	0,23 mV en un ADC de 12 bits y 3,3 V.
Mapeo de polos	$z = e^{s \cdot T_s}$	Semiplano izquierdo → interior del círculo.
Tustin	$s \approx (2/T_s) \cdot (z-1)/(z+1)$	Bilineal.
Estado discreto	$\Phi = e^{A \cdot T_s}$, $\Gamma = \int_0^{T_s} e^{A\tau} d\tau \cdot B$	Equivalente ZOH exacto.
Función de transferencia	$G(s) = C(sI - A)^{-1}B + D$	Del estado a la entrada-salida.

Ejemplo resuelto: equivalente discreto de un primer orden

1. $G(s) = 1/(0{,}1s + 1)$, polo en -10; $T_s = 50$ ms.
2. ZOH: $z = e^{-10 \cdot 0{,}05} = 0{,}607$; $y[k+1] = 0{,}607 \cdot y[k] + 0{,}393 \cdot u[k]$.
3. Euler: $z = 1 - 10 \cdot 0{,}05 = 0{,}5$. Tustin: $z = (1 - 0{,}25)/(1 + 0{,}25) = 0{,}6$.
4. Con $T_s = 0{,}25$ s Euler daría $z = -1{,}5$: inestable aunque la planta sea estable.

discretizacion.py calcula Φ y Γ de un motor con posición y velocidad por la exponencial de matrices, y compara ZOH, Euler y Tustin.

Métodos de discretización

Método	Sustitución	Ventaja	Cuidado
ZOH exacto	$\Phi = e^{AT_s}$	Coincide en los instantes de muestreo	Requiere calcular la exponencial
Euler hacia adelante	$s = (z - 1)/T_s$	Muy simple	Puede volver inestable un sistema estable
Tustin (bilineal)	$s = (2/T_s)(z - 1)/(z + 1)$	Conserva la estabilidad	Distorsiona las frecuencias altas (usar prewarping)

Código del capítulo

Los archivos completos vienen en el paquete ZIP del capítulo (carpeta `codigo/`).

discretizacion.py — equivalente ZOH por exponencial de matrices, Euler, Tustin, mapeo de polos y cuantificación. *(extracto: primeras 60 de 108 líneas)*

```
"""
discretizacion.py
De un modelo continuo a uno digital (capítulo 01).

Planta: motor con carga, G(s) = 1 / (s·(0,1·s + 1)) (posición), en espacio de
estados con x = [posición, velocidad]:
    A = [[0, 1], [0, -10]],  B = [[0], [10]],  C = [[1, 0]]

  1. Equivalente discreto exacto con retenedor de orden cero (ZOH):
       Phi = e^(A·Ts),  Gamma = ∫ e^(A·t) dt · B   (se obtiene con una matriz aumentada)
  2. Comparación de la respuesta al escalón de velocidad con Euler y con Tustin.
  3. Polos: z = e^(s·Ts) y estabilidad dentro del círculo unitario.
  4. Ruido de cuantificación de un ADC de 12 bits: q/raíz(12).

Uso:  python discretizacion.py
"""
import cmath
import math

def mat(a, b):
    """Producto de matrices (listas de listas)."""
    return [[sum(a[i][k] * b[k][j] for k in range(len(b))) for j in range(len(b[0]))] for i in range(len(a))]


def suma(a, b, s=1.0):
    return [[x + s * y for x, y in zip(fa, fb)] for fa, fb in zip(a, b)]


def esc(a, k):
    return [[k * x for x in f] for f in a]


def tr(a):
    return [list(f) for f in zip(*a)]


def eye(n):
    return [[1.0 if i == j else 0.0 for j in range(n)] for i in range(n)]


def inv(a):
    n = len(a)
```

```
    m = [f[:] + e for f, e in zip(a, eye(n))]
    for c in range(n):
        p = max(range(c, n), key=lambda f: abs(m[f][c]))
        m[c], m[p] = m[p], m[c]
        piv = m[c][c]
        m[c] = [x / piv for x in m[c]]
        for f in range(n):
            if f != c:
                k = m[f][c]
                m[f] = [x - k * y for x, y in zip(m[f], m[c])]
    return [f[n:] for f in m]


def expm(a, t=1.0):
    """e^(A·t) por escalado y elevación al cuadrado con serie de Taylor."""
    n, s = len(a), 0
    norma = max(sum(abs(x) for x in f) for f in a) * abs(t)
    while norma > 0.5:
```

controlador_pi_discreto.ino — PI discretizado con Tustin, periodo fijo con micros() y anti-windup.

Errores frecuentes

- Ejecutar el controlador con delay() y un periodo que varía.
- Discretizar con Euler y un periodo grande.
- Olvidar que el ZOH y el cálculo agregan retardo al lazo.
- Confundir la estabilidad en s (semiplano izquierdo) con la de z (círculo unitario).

Autoevaluación

1. ¿Dónde deben estar los polos de un sistema discreto estable?
2. ¿Qué hace el retenedor de orden cero?
3. ¿Qué método de discretización conserva siempre la estabilidad?
4. ¿Cómo se obtiene G(s) desde el espacio de estados?

Respuestas. 1. Dentro del círculo unitario. **2.** Mantiene constante la salida del DAC entre dos muestras. **3.** Tustin (bilineal) y el ZOH exacto. **4.** $G(s) = C(sI - A)^{-1}B + D$.