CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA

Video 03 — Parte 2A: Modelado en Ingeniería: Sistemas Rotacional-Traslacional y Electromecánicos — Metodología de Modelado, Trenes de Engranajes, Inercia Reflejada, Husillo de Bolas y Motor de CC Acoplado

Parte del Curso: Parte 2A — Modelado en Ingeniería: Sistemas Rotacional-Traslacional y Electromecánicos
Planta / Sistema Modelado: Servomotor de CC con Reductor de Engranajes y Husillo de Bolas CNC

$$J_{eq} = J_1 + (N_1/N_2)^2 \cdot J_2 + M \cdot (p/(2\pi))^2 \mid v_a = R_a \cdot i_a + L_a \cdot (di_a/dt) + K_e \cdot \omega,\ T_m = K_t \cdot i_a$$

1. Síntesis de Conceptos Clave y Fórmulas de Ingeniería

- Reflexión de impedancias mecánicas al eje primario del motor: $J_{eq} = J_1 + (N_1/N_2)^2 \cdot J_2$ y $c_{eq} = c_1 + (N_1/N_2)^2 \cdot c_2$.
- Conversión rotacional a traslacional: en piñón-cremallera o polea $J_{ref} = M \cdot r^2$; en husillo de bolas de paso p, $J_{ref} = M \cdot (p / 2\pi)^2$.
- Acoplamiento electromecánico: Ley de Lorentz ($T_m = K_t \cdot i_a$) y Ley de Inducción de Faraday ($e_b = K_e \cdot \omega$).
- Modelo completo de segundo orden del motor de CC controlado por armadura y reducción a primer orden cuando $\tau_e = L_a/R_a << \tau_m$.

2. Parámetros Nominales del Modelo Matemático (Video 03)

Parámetro Físico / Canónico	Símbolo	Valor Nominal	Significado en el Modelado Dinámico
Masa / Momento de Inercia Equivalente	`m_eq / J_eq`	0.002	Almacenador de energía cinética opositor a cambios de velocidad
Amortiguamiento / Resistencia Equivalente	`c_eq / R_eq`	0.25	Elemento disipador de energía que gobierna la tasa de atenuación
Rigidez / Elasticidad Equivalente	`k_eq / 1/C`	0.68	Almacenador de energía potencial restauradora
Frecuencia Natural No Amortiguada	`ω_n [rad/s]`	18.44 rad/s	Distancia radial de los polos al origen en el plano complejo s
Factor de Amortiguamiento Relativo	`ζ [-]`	0.339	Determina el régimen dinámico y el sobreimpulso máximo M_p
Ganancia Estática / Constante de Tiempo	`K / τ [s]`	K = 1.176, τ = 0.08 s	Sensibilidad estacionaria y rapidez del modo exponencial e^{-t/τ}

3. Desarrollo Académico Diapositiva por Diapositiva

Diapositiva 01: Metodología de Modelado de Sistemas Reales en Ingeniería

Fronteras del Sistema, Hipótesis Simplificadoras, Subsistemas y Acoplamiento Energético

- **Turno 1 (Masculina):** En la Parte dos damos el salto de los bloques aislados hacia el modelado de sistemas mecatrónicos completos, donde interactúan simultáneamente dominios eléctricos, magnéticos, rotacionales, traslacionales e hidráulicos.
- **Turno 2 (Femenina):** El primer paso en ingeniería de sistemas es trazar la frontera de análisis e identificar con precisión cuáles son las entradas manipulables de control, cuáles son las perturbaciones de carga no controladas y cuáles son las salidas medidas por los sensores.
- **Turno 3 (Masculina):** El segundo paso es declarar las hipótesis de modelado: decidir qué elementos son suficientemente rígidos para despreciar su flexión, qué inductancias o masas secundarias tienen efecto despreciable y en qué rango de frecuencias operará la máquina.
- **Turno 4 (Femenina):** Finalmente, escribimos las ecuaciones de cada subdominio y las enlazamos mediante las relaciones cinemáticas o transductoras de frontera, reduciendo todo el conjunto a un modelo compacto listo para el diseño de control.

Diapositiva 02: Sistemas Rotacional-Traslacional I: Trenes de Engranajes

Relación de Dientes y Radios N_1/N_2, Transformación de Velocidad Angular y Multiplicación de Par

- **Turno 1 (Masculina):** Casi ningún servomotor eléctrico mueve directamente la carga sin pasar antes por una etapa reductora. Analicemos un par de engranajes donde el piñón motriz tiene radio erre uno y ene uno dientes, y la rueda conducida tiene radio erre dos y ene dos dientes.
- **Turno 2 (Femenina):** En el punto de contacto primitivo no existe deslizamiento, por lo que las velocidades lineales tangenciales son idénticas: erre uno por omega uno es igual a erre dos por omega dos.
- **Turno 3 (Masculina):** Como el radio primitivo es proporcional al número de dientes, la velocidad angular de salida omega dos es igual a ene uno sobre ene dos multiplicado por la velocidad del motor omega uno; y lo mismo se cumple para los ángulos teta dos y teta uno.
- **Turno 4 (Femenina):** Por conservación de potencia mecánica en un reductor ideal sin pérdidas, el producto par por velocidad angular se mantiene constante, de modo que el par entregado en el eje lento se multiplica por la relación inversa ene dos sobre ene uno.

Diapositiva 03: Inercia, Amortiguamiento y Rigidez Reflejados al Eje del Motor

Deducción de la Ley del Cuadrado de la Relación de Transmisión $J_{eq} = J_1 + (N_1/N_2)^2 \cdot J_2$

- **Turno 1 (Masculina):** Para no arrastrar dos ecuaciones diferenciales con dos ángulos distintos en un tren de engranajes, podemos reflejar toda la inercia jota dos, el amortiguamiento ce dos y la rigidez ka dos del eje de carga directamente hacia el eje del motor.
- **Turno 2 (Femenina):** El par resistente que exige acelerar el eje secundario vale jota dos por alfa dos más ce dos por omega dos; pero para reflejar ese par de vuelta al piñón primario debemos multiplicarlo por la relación ene uno sobre ene dos.
- **Turno 3 (Masculina):** Como además la aceleración alfa dos es a su vez ene uno sobre ene dos por alfa uno, el factor de relación de dientes aparece dos veces multiplicando, quedando elevado al cuadrado.
- **Turno 4 (Femenina):** Así obtenemos la regla de oro del diseño mecatrónico: la inercia equivalente total vista por el motor es jota uno más el cuadrado de ene uno sobre ene dos multiplicado por jota dos; un reductor diez a uno divide por cien la inercia de la carga vista por el motor.

Diapositiva 04: Conversión Rotacional-Traslacional: Piñón-Cremallera y Polea-Banda

Cinemática Lineal $x = r \cdot \theta$ y Masa Traslacional Reflejada como Inercia $J_{eq} = J + M \cdot r^2$

- **Turno 1 (Masculina):** Estudiemos ahora los mecanismos que convierten el giro continuo de un motor en movimiento lineal rectilíneo, empezando por el sistema de piñón y cremallera y la transmisión por polea y banda dentada síncrona.
- **Turno 2 (Femenina):** Si el piñón o la polea motriz tiene radio primitivo erre y gira un ángulo teta en radianes, el carro lineal de masa eme se desplaza una distancia equis igual a erre por teta, con velocidad lineal erre por omega y aceleración erre por alfa.
- **Turno 3 (Masculina):** Para acelerar linealmente el carro de masa eme y vencer un amortiguamiento lineal ce, la banda o la cremallera debe ejercer una fuerza lineal efe igual a masa por equis dos puntos más ce por equis punto.
- **Turno 4 (Femenina):** Esa fuerza lineal multiplicada por el brazo de palanca erre del radio primitivo produce sobre el eje del motor un par resistente igual a eme por erre al cuadrado por teta dos puntos; por tanto, toda masa traslacional eme se refleja al eje rotatorio como una inercia equivalente igual a masa por radio al cuadrado.

Diapositiva 05: Husillo de Bolas Recirculantes (Ball Screw) en Ejes Lineales CNC

Relación Paso-Ángulo $x = (p / 2\pi) \cdot \theta$ e Inercia Equivalente $J_{eq} = J_h + M \cdot (p / 2\pi)^2$

- **Turno 1 (Masculina):** En centros de mecanizado ce ene ce, robots cartesianos y prensas electromecánicas de alta fuerza, la conversión rotacional a traslacional se realiza mediante un husillo de bolas recirculantes caracterizado por su paso o avance pe en metros por revolución.
- **Turno 2 (Femenina):** Como una vuelta completa equivale a dos pi radianes y produce un avance lineal igual al paso pe, el radio equivalente de conversión del husillo es simplemente el paso pe dividido entre dos pi.
- **Turno 3 (Masculina):** Igualando la energía cinética traslacional de la mesa de masa eme con la energía cinética rotacional equivalente en el eje del husillo, demostramos que la inercia reflejada de la mesa es igual a la masa multiplicada por el cuadrado del paso dividido entre dos pi.
- **Turno 4 (Femenina):** Gracias a que el paso pe suele ser de apenas cinco o diez milímetros por vuelta, el factor paso sobre dos pi al cuadrado es del orden de diez a la menos seis, permitiendo que un servomotor compacto mueva mesas de cientos de kilogramos con altísima rigidez y precisión.

Diapositiva 06: Principios Físicos de los Sistemas Electromecánicos Acoplados

Ley de Fuerza de Lorentz $T_m = K_t \cdot i_a$ y Ley de Inducción de Faraday $e_b = K_e \cdot \omega$

- **Turno 1 (Masculina):** Entremos ahora al corazón de la actuación mecatrónica: los sistemas electromecánicos, donde un circuito eléctrico y un mecanismo móvil intercambian energía a través de un campo magnético.
- **Turno 2 (Femenina):** Existen dos leyes electromagnéticas fundamentales que acoplan ambos dominios. La primera es la ley de fuerza de Lorentz: un conductor de longitud ele por el que circula una corriente i dentro de un campo magnético be experimenta una fuerza mecánica proporcional a la corriente.
- **Turno 3 (Masculina):** En el rotor de un motor de corriente continua, la suma de fuerzas sobre todos los conductores de la armadura produce un par electromagnético te sub eme directamente proporcional a la corriente de armadura: par motor es igual a la constante de par ka sub te por i sub a.
- **Turno 4 (Femenina):** La segunda ley es la ley de inducción de Faraday: al girar el rotor a velocidad angular omega cortando las líneas de flujo magnético, se induce en sus devanados una tensión opuesta llamada fuerza contraelectromotriz e sub be, igual a la constante eléctrica ka sub e multiplicada por la velocidad omega.

Diapositiva 07: Modelado del Motor de Corriente Continua Controlado por Armadura

Ecuación Eléctrica v_a = R_a·i_a + L_a·di_a/dt + K_e·ω y Ecuación Mecánica J·dω/dt + b·ω = K_t·i_a - T_L

- **Turno 1 (Masculina):** Planteemos las ecuaciones diferenciales acopladas del servomotor de corriente continua controlado por tensión de armadura ve sub a, que cuenta con imanes permanentes o flujo de excitación constante en el estator.
- **Turno 2 (Femenina):** Aplicando la ley de voltajes de Kirchhoff en la malla eléctrica del inducido, la tensión aplicada ve sub a es igual a la caída resistiva erre sub a por i sub a, más la caída inductiva ele sub a por la derivada de i sub a, más la fuerza contraelectromotriz ka sub e por omega.
- **Turno 3 (Masculina):** Simultáneamente, aplicando la segunda ley de Newton rotacional sobre el eje que agrupa la inercia equivalente jota y la fricción viscosa be, la inercia jota por la derivada de omega, más be por omega, es igual al par motor ka sub te por i sub a menos el par de carga te sub ele.
- **Turno 4 (Femenina):** Observemos el acoplamiento bidireccional interno: la corriente eléctrica genera el par que acelera la mecánica, pero la velocidad mecánica resultante genera la fuerza contraelectromotriz que reduce la corriente de entrada, actuando como una realimentación negativa natural de velocidad.

Diapositiva 08: Ecuación Diferencial de Segundo Orden y Aproximación de Primer Orden del Motor CC

Constante de Tiempo Eléctrica τ_e = L_a/R_a frente a la Constante Mecánica τ_m = J_m/b_ef

- **Turno 1 (Masculina):** Si despejamos la corriente de armadura de la ecuación mecánica y la sustituimos junto con su derivada en la ecuación eléctrica para par de carga nulo, obtenemos una única ecuación diferencial de segundo orden para la velocidad angular omega.
- **Turno 2 (Femenina):** La ecuación resultante es: ele sub a por jota por la segunda derivada de omega, más erre sub a por jota más ele sub a por be por la primera derivada de omega, más erre sub a por be más ka sub te por ka sub e multiplicado por omega, igual a ka sub te por el voltaje de armadura.
- **Turno 3 (Masculina):** En la mayoría de los motores reales, la inductancia de armadura ele sub a es muy pequeña, por lo que la constante de tiempo eléctrica ele sobre erre, de apenas un milisegundo, es muchísimo más rápida que la constante de tiempo mecánica del rotor.
- **Turno 4 (Femenina):** Cuando despreciamos ele sub a frente a erre sub a, la ecuación colapsa a un modelo de primer orden donde el término ka sub te por ka sub e dividido entre erre sub a actúa como un amortiguamiento electromagnético adicional que frena y estabiliza el motor.

Diapositiva 09: Motor de CC Controlado por Campo y Actuador Lineal de Bobina Móvil (Voice Coil)

Desacoplamiento de Fuerza Contraelectromotriz en Malla de Campo y Dinámica Traslacional Electrodinámica

- **Turno 1 (Masculina):** Existe una segunda configuración clásica del motor de corriente continua: mantener constante la corriente de armadura mediante una fuente de corriente y controlar el par variando la tensión del devanado de campo del estator ve sub efe.
- **Turno 2 (Femenina):** En el motor controlado por campo, la malla de control satisface ve sub efe igual a erre sub efe por i sub efe más ele sub efe por la derivada de i sub efe, sin que aparezca fuerza contraelectromotriz dependiente de la velocidad del rotor en dicha malla.
- **Turno 3 (Masculina):** Por otra parte, la versión traslacional directa del motor de corriente continua es el actuador lineal de bobina móvil o voice coil, empleado en enfoque óptico de láseres, cabezales de disco duro y agitadores de ensayo modal.
- **Turno 4 (Femenina):** En el actuador de bobina móvil, la ecuación eléctrica es ele por la derivada de i, más erre por i, más el factor de fuerza be por ele por la velocidad lineal equis punto igual al voltaje; y la ecuación mecánica es masa por equis dos puntos, más amortiguamiento por equis punto, más rigidez por equis, igual a be por ele por la corriente.

Diapositiva 10: Modelado de Solenoides Electromagnéticos y Transductores Sísmicos

Reluctancia Variable con Inductancia Dependiente de Posición L(x) y Acelerómetro Masa-Resorte

- **Turno 1 (Masculina):** Cerramos la Parte dos A con dos dispositivos electromecánicos fundamentales: el solenoide de reluctancia variable que acciona electroválvulas e inyectores, y el acelerómetro sísmico MEMS que mide vibraciones.
- **Turno 2 (Femenina):** En un solenoide, a medida que el émbolo ferromagnético penetra una distancia equis cerrando el entrehierro de aire, la inductancia de la bobina ele de equis aumenta, por lo que el enlace de flujo magnético depende simultáneamente de la corriente y de la posición.
- **Turno 3 (Masculina):** A partir de la coenergía magnética se demuestra que la fuerza de atracción del solenoide es proporcional al cuadrado de la corriente multiplicado por la derivada de la inductancia respecto a la posición equis, constituyendo un sistema intrínsecamente no lineal.
- **Turno 4 (Femenina):** Finalmente, en un acelerómetro sísmico, una masa de prueba eme unida a la carcasa por micro-resortes ka y amortiguamiento ce experimenta un desplazamiento interno relativo zeta: la ecuación masa por zeta dos puntos, más ce por zeta punto, más ka por zeta igual a menos masa por la aceleración de la base permite medir cualquier aceleración externa.

4. Firmware C++/Arduino: Integrador RK4 y Filtro de Tustin (simulacion_hil_y_adquisicion_video_03.ino)

```
/*
 * ============================================================================
 * CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
 * VIDEO 03: Video 03 — Parte 2A: Modelado en Ingeniería: Sistemas Rotacional-Traslacional y Electromecánicos
 * Archivo: simulacion_hil_y_adquisicion_video_03.ino
 * ----------------------------------------------------------------------------
 * Descripción Técnica:
 *   Firmware en tiempo real (Ts = 2 ms -> 500 Hz) para simulación Hardware-in-
 *   the-Loop (HIL), integración numérica Runge-Kutta de 4.º orden (RK4) de la
 *   planta mecatrónica "Servomotor de CC con Reductor de Engranajes y Husillo de Bolas CNC",
 *   filtro discreto por Transformada Bilineal de Tustin y adquisición por ADC
 *   para identificación experimental de parámetros (K, tau, wn, zeta).
 *
 * Ecuación Modelo:
 *   J_eq = J_1 + (N_1/N_2)²·J_2 + M·(p/(2π))²   |   v_a = R_a·i_a + L_a·(di_a/dt) + K_e·ω,  T_m = K_t·i_a
 * ============================================================================
 */

#include <Arduino.h>

// --- Definición de Pines de Hardware ---
const uint8_t PIN_ADC_PLANTA    = A0;  // Entrada analógica desde sensor real (0..5 V)
const uint8_t PIN_ADC_PERTURB   = A1;  // Perturbación externa / carga medida
const uint8_t PIN_PWM_ACTUADOR  = 9;   // Salida PWM hacia driver / puente H / válvula
const uint8_t PIN_SYNC_SCOPE    = 8;   // Pulso de sincronismo para osciloscopio

// --- Parámetros Físicos del Modelo (Video 03) ---
const float M_EQ       = 0.0020f;   // Masa / Inercia equivalente
const float C_EQ       = 0.2500f;   // Amortiguamiento / Resistencia equivalente
const float K_EQ       = 0.6800f;   // Rigidez / Elasticidad equivalente
const float WN_RAD_S   = 18.440f;  // Frecuencia natural no amortiguada [rad/s]
const float ZETA_REL   = 0.339f; // Factor de amortiguamiento relativo [-]
const float K_GAIN     = 1.176f;   // Ganancia estática de la planta
const float TAU_SEC    = 0.080f; // Constante de tiempo equivalente [s]

// --- Configuración del Reloj de Muestreo en Tiempo Real ---
const float TS_SEC              = 0.002f; // Periodo de integración Ts = 2.0 ms (500 Hz)
const unsigned long TS_MICROS   = 2000UL;
unsigned long prevMicros        = 0;
unsigned long stepCounter       = 0;

// --- Vector de Estado del Integrador RK4: x1 = y(t), x2 = dy(t)/dt ---
float x1_pos = 0.0f;
float x2_vel = 0.0f;
float u_cmd  = 0.0f;

// --- Filtro Digital de 1.er Orden Discretizado por Transformada de Tustin ---
// G_f(s) = K / (tau*s + 1)  -->  y[k] = b0*u[k] + b1*u[k-1] - a1*y[k-1]
float b0_tustin = 0.0f;
float b1_tustin = 0.0f;
float a1_tustin = 0.0f;
float u_prev1   = 0.0f;
float y_tustin  = 0.0f;

// --- Derivadas del Espacio de Estados Continuo: dx1/dt = x2, dx2/dt = f(x1,x2,u) ---
inline void evaluarDerivadasEstado(float x1, float x2, float u, float &dx1, float &dx2) {
  dx1 = x2;
  // Forma canónica de 2.º orden: x2' = K*wn^2*u - 2*zeta*wn*x2 - wn^2*x1
  dx2 = (K_GAIN * WN_RAD_S * WN_RAD_S * u)
      - (2.0f * ZETA_REL * WN_RAD_S * x2)
      - (WN_RAD_S * WN_RAD_S * x1);
}

// --- Paso de Integración Numérica Runge-Kutta de 4.º Orden (RK4) ---
void integrarPasoRK4(float u_in, float dt) {
  float k1_x1, k1_x2, k2_x1, k2_x2, k3_x1, k3_x2, k4_x1, k4_x2;
```

```
  evaluarDerivadasEstado(x1_pos, x2_vel, u_in, k1_x1, k1_x2);
  evaluarDerivadasEstado(x1_pos + 0.5f * dt * k1_x1, x2_vel + 0.5f * dt * k1_x2, u_in, k2_x1, k2_x2);
  evaluarDerivadasEstado(x1_pos + 0.5f * dt * k2_x1, x2_vel + 0.5f * dt * k2_x2, u_in, k3_x1, k3_x2);
  evaluarDerivadasEstado(x1_pos + dt * k3_x1,        x2_vel + dt * k3_x2,        u_in, k4_x1, k4_x2);

  x1_pos += (dt / 6.0f) * (k1_x1 + 2.0f * k2_x1 + 2.0f * k3_x1 + k4_x1);
  x2_vel += (dt / 6.0f) * (k1_x2 + 2.0f * k2_x2 + 2.0f * k3_x2 + k4_x2);
}

// --- Inicialización de Coeficientes Discretos por Regla Trapezoidal (Tustin) ---
void inicializarFiltroTustin() {
  float denom = 2.0f * TAU_SEC + TS_SEC;
  b0_tustin   = (K_GAIN * TS_SEC) / denom;
  b1_tustin   = b0_tustin;
  a1_tustin   = (TS_SEC - 2.0f * TAU_SEC) / denom;
}

void setup() {
  Serial.begin(115200);
  pinMode(PIN_PWM_ACTUADOR, OUTPUT);
  pinMode(PIN_SYNC_SCOPE, OUTPUT);
  inicializarFiltroTustin();
  prevMicros = micros();

  Serial.println(F("# ============================================================"));
  Serial.println(F("# CURSO MODELADO DE SISTEMAS EN MECATRONICA - VIDEO 03"));
  Serial.println(F("# Columnas CSV: t_s,u_cmd,y_rk4_2do,dy_rk4,y_tustin_1ro,adc_V"));
  Serial.println(F("# ============================================================"));
}

void loop() {
  unsigned long nowMicros = micros();
  if (nowMicros - prevMicros >= TS_MICROS) {
    prevMicros += TS_MICROS;
    float t_sec = stepCounter * TS_SEC;

    // 1. Generar entrada de prueba: Escalón unitario + barrido senoidal suave
    if (t_sec < 0.2f) {
      u_cmd = 0.0f;
    } else if (t_sec < 3.0f) {
      u_cmd = 1.0f; // Respuesta al escalón unitario u_s(t)
      digitalWrite(PIN_SYNC_SCOPE, HIGH);
    } else {
      // Excitación armónica para medir respuesta en frecuencia G(jw)
      u_cmd = 1.0f + 0.5f * sinf(WN_RAD_S * (t_sec - 3.0f));
      digitalWrite(PIN_SYNC_SCOPE, LOW);
    }

    // 2. Actualizar gemelo digital de 2.º orden por RK4
    integrarPasoRK4(u_cmd, TS_SEC);

    // 3. Actualizar modelo de 1.er orden por Transformada de Tustin
    y_tustin = b0_tustin * u_cmd + b1_tustin * u_prev1 - a1_tustin * y_tustin;
    u_prev1  = u_cmd;

    // 4. Leer planta física real en el ADC y comandar salida PWM
    float adcVolt = analogRead(PIN_ADC_PLANTA) * (5.0f / 1023.0f);
    int pwmVal    = constrain((int)(x1_pos * 127.0f + 128.0f), 0, 255);
    analogWrite(PIN_PWM_ACTUADOR, pwmVal);

    // 5. Transmitir telemetría cada 5 pasos (10 ms -> 100 Hz)
    if (stepCounter % 5 == 0) {
      Serial.print(t_sec, 3);      Serial.print(',');
      Serial.print(u_cmd, 4);      Serial.print(',');
      Serial.print(x1_pos, 4);     Serial.print(',');
      Serial.print(x2_vel, 4);     Serial.print(',');
      Serial.print(y_tustin, 4);   Serial.print(',');
      Serial.println(adcVolt, 3);
```

```
    }
    stepCounter++;
  }
}
```

5. Script Python: Polos, Respuesta Temporal y Bode (modelado_laplace_y_bode_video_03.py)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
================================================================================
CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
VIDEO 03: Video 03 — Parte 2A: Modelado en Ingeniería: Sistemas Rotacional-Traslacional y Electromecánicos
Archivo: modelado_laplace_y_bode_video_03.py
--------------------------------------------------------------------------------
Descripción:
  Script integral en Python para simulación de ecuaciones diferenciales (RK4),
  análisis de sistemas de 1.er y 2.º orden, cálculo exacto de polos y ceros
  en el plano complejo s, especificaciones transitorias (tr, tp, Mp, ts),
  evaluación fasorial G(jw), trazado de Diagramas de Bode y cálculo de
  Márgenes de Ganancia (MG) y de Fase (MF).
================================================================================
"""

import math
from typing import Dict, List, Tuple

# Parámetros nominales del Video 03: Servomotor de CC con Reductor de Engranajes y Husillo de Bolas CNC
M_EQ     = 0.002
C_EQ     = 0.25
K_EQ     = 0.68
WN_RAD   = 18.44
ZETA     = 0.339
K_STATIC = 1.176
TAU_SEC  = 0.08


def metricas_primer_orden(K: float, tau: float, escalon_A: float = 1.0) -> Dict[str, float]:
    """Calcula las medidas de desempeño exactas de G(s) = K / (tau*s + 1)."""
    y_ss = K * escalon_A
    y_1tau = y_ss * (1.0 - math.exp(-1.0))
    t_d = tau * math.log(2.0)          # 50% del valor final
    t_r = tau * math.log(9.0)          # 10% al 90% (~2.197 * tau)
    t_s_5pct = 3.0 * tau               # Banda del +-5%
    t_s_2pct = 4.0 * tau               # Banda del +-2%
    omega_c = 1.0 / tau                # Frecuencia de esquina (-3 dB, -45 deg)
    return {
        "y_ss": round(y_ss, 4),
        "y_en_1tau_63pct": round(y_1tau, 4),
        "tiempo_retardo_td_s": round(t_d, 4),
        "tiempo_subida_tr_s": round(t_r, 4),
        "tiempo_establecimiento_5pct_s": round(t_s_5pct, 4),
        "tiempo_establecimiento_2pct_s": round(t_s_2pct, 4),
        "frecuencia_esquina_wc_rads": round(omega_c, 4),
    }


def metricas_segundo_orden(K: float, wn: float, zeta: float) -> Dict[str, object]:
    """Calcula polos en el plano s y métricas (wd, tr, tp, Mp, ts, wr, Mr, w_BW)."""
    sigma = zeta * wn
    if zeta < 1.0:
        wd = wn * math.sqrt(1.0 - zeta ** 2)
        polos = f"-{sigma:.3f} +- j*{wd:.3f}"
        regimen = "Subamortiguado (0 < zeta < 1)" if zeta > 0 else "No amortiguado"
        tr = (math.pi - math.acos(zeta)) / wd
        tp = math.pi / wd
        mp = math.exp(-math.pi * zeta / math.sqrt(1.0 - zeta ** 2))
    elif abs(zeta - 1.0) < 1e-6:
        wd = 0.0
        polos = f"-{sigma:.3f} (Raíz real doble)"
        regimen = "Críticamente amortiguado (zeta = 1)"
        tr = 3.35 / wn
        tp = float("inf")
        mp = 0.0
```

```
    else:
        wd = 0.0
        r1 = -sigma + wn * math.sqrt(zeta ** 2 - 1.0)
        r2 = -sigma - wn * math.sqrt(zeta ** 2 - 1.0)
        polos = f"s1 = {r1:.3f}, s2 = {r2:.3f}"
        regimen = "Sobreamortiguado (zeta > 1)"
        tr = 2.2 / abs(r1)
        tp = float("inf")
        mp = 0.0

    ts_2pct = 4.0 / sigma if sigma > 0 else float("inf")

    # Resonancia y Ancho de Banda en frecuencia
    if zeta < 1.0 / math.sqrt(2.0):
        wr = wn * math.sqrt(1.0 - 2.0 * (zeta ** 2))
        mr = 1.0 / (2.0 * zeta * math.sqrt(1.0 - zeta ** 2))
        mr_db = 20.0 * math.log10(mr)
    else:
        wr = 0.0
        mr = 1.0
        mr_db = 0.0

    w_bw = wn * math.sqrt((1.0 - 2.0 * zeta ** 2) + math.sqrt(4.0 * zeta ** 4 - 4.0 * zeta ** 2 + 2.0))

    return {
        "regimen": regimen,
        "polos_plano_s": polos,
        "atenuacion_sigma": round(sigma, 4),
        "frecuencia_amortiguada_wd_rads": round(wd, 4),
        "tiempo_subida_tr_s": round(tr, 4),
        "tiempo_pico_tp_s": round(tp, 4) if math.isfinite(tp) else None,
        "sobreimpulso_Mp_pct": round(mp * 100.0, 2),
        "tiempo_establecimiento_ts_2pct_s": round(ts_2pct, 4),
        "frecuencia_resonancia_wr_rads": round(wr, 4),
        "pico_resonancia_Mr_dB": round(mr_db, 2),
        "ancho_banda_wBW_rads": round(w_bw, 4),
    }


def evaluar_respuesta_frecuencia_bode(
    K: float, wn: float, zeta: float, frecuencias_rads: List[float]
) -> List[Dict[str, float]]:
    """Evalúa G(jw) = K / ((1 - u^2) + j*(2*zeta*u)) con u = w / wn."""
    tabla_bode = []
    for w in frecuencias_rads:
        u = w / wn
        parte_real = 1.0 - (u ** 2)
        parte_imag = 2.0 * zeta * u
        mag_abs = K / math.sqrt(parte_real ** 2 + parte_imag ** 2)
        mag_db = 20.0 * math.log10(max(mag_abs, 1e-12))
        fase_deg = -math.degrees(math.atan2(parte_imag, parte_real))
        tabla_bode.append({
            "w_rads": round(w, 3),
            "u_norm": round(u, 3),
            "mag_abs": round(mag_abs, 4),
            "mag_dB": round(mag_db, 2),
            "fase_deg": round(fase_deg, 2),
        })
    return tabla_bode


def identificar_segundo_orden_desde_escalon(
    u_escalon: float, y_estacionario: float, y_pico_max: float, t_pico_s: float
) -> Dict[str, float]:
    """Identifica experimentalmente (K, Mp, zeta, wn) a partir de la curva medida."""
    K_id = y_estacionario / u_escalon
    mp_id = max(0.0, (y_pico_max - y_estacionario) / y_estacionario)
    if mp_id > 0.0:
```

```
        ln_mp = abs(math.log(mp_id))
        zeta_id = ln_mp / math.sqrt(math.pi ** 2 + ln_mp ** 2)
    else:
        zeta_id = 1.0
    wn_id = math.pi / (t_pico_s * math.sqrt(max(1e-6, 1.0 - zeta_id ** 2)))
    return {
        "K_identificada": round(K_id, 4),
        "Mp_medido": round(mp_id, 4),
        "zeta_identificado": round(zeta_id, 4),
        "wn_identificada_rads": round(wn_id, 4),
    }


if __name__ == "__main__":
    print("=" * 76)
    print(" CURSO: MODELADO DE SISTEMAS EN MECATRÓNICA — VIDEO 03")
    print(" Planta:", "Servomotor de CC con Reductor de Engranajes y Husillo de Bolas CNC")
    print("=" * 76)

    m1 = metricas_primer_orden(K_STATIC, TAU_SEC)
    print("\n[1] Análisis de Sistema de Primer Orden (K, tau):", m1)

    m2 = metricas_segundo_orden(K_STATIC, WN_RAD, ZETA)
    print("\n[2] Análisis de Sistema de Segundo Orden (K, wn, zeta):", m2)

    freqs = [0.1 * WN_RAD, 0.5 * WN_RAD, WN_RAD, 2.0 * WN_RAD, 10.0 * WN_RAD]
    bode_pts = evaluar_respuesta_frecuencia_bode(K_STATIC, WN_RAD, ZETA, freqs)
    print("\n[3] Tabla de Respuesta en Frecuencia y Diagrama de Bode G(jw):")
    for row in bode_pts:
        print(f"    w = {row['w_rads']:7.2f} rad/s | |G| = {row['mag_abs']:6.4f} ({row['mag_dB']:6.2f} dB) | fase
= {row['fase_deg']:7.2f} deg")

    id_res = identificar_segundo_orden_desde_escalon(1.0, 1.0, 1.254, 0.343)
    print("\n[4] Ejemplo de Identificación Experimental desde Escalón:", id_res)
```

6. Script MATLAB/Octave: Funciones de Transferencia y Estabilidad (analisis_control_mecatronico_video_03.m)

```
% ========================================================================
% CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
% VIDEO 03: Video 03 — Parte 2A: Modelado en Ingeniería: Sistemas Rotacional-Traslacional y Electromecánicos
% Archivo: analisis_control_mecatronico_video_03.m
% ------------------------------------------------------------------------
% Descripción:
%   Script para MATLAB / GNU Octave que construye las funciones de
%   transferencia en Laplace con tf(), calcula polos, ceros y factor de
%   amortiguamiento con damp(), simula la respuesta al escalón step() y al
%   impulso impulse(), cierra el lazo de realimentación con feedback(),
%   traza el Diagrama de Bode con márgenes de estabilidad margin() y el
%   Lugar Geométrico de las Raíces rlocus().
% ========================================================================

clear; clc; close all;
fprintf('========================================================================\n');
fprintf(' CURSO: MODELADO DE SISTEMAS EN MECATRONICA - VIDEO 03\n');
fprintf(' Planta: Servomotor de CC con Reductor de Engranajes y Husillo de Bolas CNC\n');
fprintf('========================================================================\n\n');

%% 1. Definicion de Parametros Fisicos y Canónicos
m_eq   = 0.002;
c_eq   = 0.25;
k_eq   = 0.68;
wn     = 18.44;
zeta   = 0.339;
K_gain = 1.176;
tau    = 0.08;

s = tf('s');

%% 2. Funcion de Transferencia de Primer Orden: G1(s) = K / (tau*s + 1)
G1 = K_gain / (tau * s + 1);
fprintf('--- Sistema de Primer Orden G1(s) ---\n');
disp(G1);

%% 3. Funcion de Transferencia de Segundo Orden: G2(s) = K*wn^2 / (s^2 + 2*zeta*wn*s + wn^2)
num_G2 = [0.8];
den_G2 = [0.002, 0.25, 0.68];
G2 = tf(num_G2, den_G2);
fprintf('--- Sistema Principal del Video 03: G2(s) ---\n');
disp(G2);

fprintf('--- Polos, Factor de Amortiguamiento (zeta) y Frecuencia Natural (wn) ---\n');
damp(G2);
polos_G2 = pole(G2);
ceros_G2 = zero(G2);

%% 4. Reduccion de Diagramas de Bloques: Serie, Paralelo y Lazo Cerrado T(s)
Kp_ctrl = 4.0;
Ki_ctrl = 8.0;
Gc = Kp_ctrl + Ki_ctrl / s;          % Controlador PI en paralelo
L_abierto = series(Gc, G2);          % Sistemas en serie L(s) = Gc(s)*G2(s)
H_sensor  = 1.0;                     % Realimentacion unitaria H(s) = 1
T_cerrado = feedback(L_abierto, H_sensor, -1); % T(s) = G(s) / (1 + G(s)H(s))

fprintf('--- Funcion de Transferencia en Lazo Cerrado T(s) ---\n');
disp(T_cerrado);

%% 5. Graficas en el Dominio del Tiempo (Escalon e Impulso) y Plano Complejo s
figure('Name', 'Video 03 - Respuestas Dinamicas y Mapa de Polos', 'Color', 'w', 'Position', [80 80 1100 720]);

subplot(2,2,1);
step(G1, 'b--', G2, 'r-', T_cerrado, 'g-.');
grid on;
title('Respuesta al Escalon Unitario u_s(t)');
legend('1.er Orden G_1(s)', 'Planta G_2(s)', 'Lazo Cerrado T(s)', 'Location', 'best');
```

```
subplot(2,2,2);
impulse(G2, 'r-', T_cerrado, 'g-.');
grid on;
title('Respuesta al Impulso Unitario \delta(t)');
legend('Planta G_2(s)', 'Lazo Cerrado T(s)', 'Location', 'best');

subplot(2,2,3);
pzmap(G2, 'r', T_cerrado, 'g');
grid on;
title('Ubicacion de Polos (x) y Ceros (o) en el Plano Complejo s');

subplot(2,2,4);
rlocus(L_abierto);
grid on;
title('Lugar Geometrico de las Raices (Root Locus)');

%% 6. Respuesta en Frecuencia: Diagrama de Bode y Margenes de Estabilidad (MG, MF)
figure('Name', 'Video 03 - Diagrama de Bode y Estabilidad Relativa', 'Color', 'w', 'Position', [140 100 980 680]);
margin(L_abierto);
grid on;

[Gm, Pm, Wcg, Wcp] = margin(L_abierto);
Gm_dB = 20 * log10(Gm);
fprintf('\n=== ESPECIFICACIONES DE ESTABILIDAD EN FRECUENCIA (VIDEO 03) ===\n');
fprintf('  Margen de Ganancia (MG)            : %.2f dB (en w_pc = %.3f rad/s)\n', Gm_dB, Wcp);
fprintf('  Margen de Fase (MF)                : %.2f deg (en w_gc = %.3f rad/s)\n', Pm, Wcg);
fprintf('  Amortiguamiento equivalente (MF/100): %.3f\n', Pm / 100.0);
```