CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA

Video 04 — Parte 2B: Linealidad, Linealización por Series de Taylor y Sistemas Hidromecánicos — Principio de Superposición, Punto de Operación, Expansión de Taylor Multivariable y Servocilindro Hidráulico

Parte del Curso: Parte 2B — Linealidad, Linealización por Series de Taylor y Sistemas Hidromecánicos
Planta / Sistema Modelado: Linealización Jacobiana por Taylor y Actuador Hidromecánico Servoválvula-Cilindro

$$\Delta y \approx (\partial f/\partial x)|_0 \cdot \Delta x + (\partial f/\partial u)|_0 \cdot \Delta u \mid q_L = K_q \cdot x_v - K_c \cdot p_L = A_p \cdot y' + C_{tp} \cdot p_L + (V_t/4\beta) \cdot p_L'$$

1. Síntesis de Conceptos Clave y Fórmulas de Ingeniería

- Condiciones de un sistema lineal LTI: Principio de Aditividad (Superposición) y Principio de Homogeneidad (Escalamiento).
- Determinación del punto de equilibrio estacionario (x_0, u_0) anulando derivadas temporales (dx/dt = 0) y definición de variables de desviación Δx = x - x_0.
- Linealización por series de Taylor de una y múltiples variables: péndulo robótico (cos θ_0) y orificio turbulento R_h = 2·Δp_0 / q_0.
- Modelado del servocilindro hidromecánico: ecuación linealizada de servoválvula q_L = K_q·x_v - K_c·p_L, fugas C_tp y compresibilidad volumétrica β.

2. Parámetros Nominales del Modelo Matemático (Video 04)

Parámetro Físico / Canónico	Símbolo	Valor Nominal	Significado en el Modelado Dinámico
Masa / Momento de Inercia Equivalente	`m_eq / J_eq`	1	Almacenador de energía cinética opositor a cambios de velocidad
Amortiguamiento / Resistencia Equivalente	`c_eq / R_eq`	14	Elemento disipador de energía que gobierna la tasa de atenuación
Rigidez / Elasticidad Equivalente	`k_eq / 1/C`	400	Almacenador de energía potencial restauradora
Frecuencia Natural No Amortiguada	`ω_n [rad/s]`	20 rad/s	Distancia radial de los polos al origen en el plano complejo s
Factor de Amortiguamiento Relativo	`ζ [-]`	0.35	Determina el régimen dinámico y el sobreimpulso máximo M_p
Ganancia Estática / Constante de Tiempo	`K / τ [s]`	K = 0.3, τ = 0.14 s	Sensibilidad estacionaria y rapidez del modo exponencial e^{-t/τ}

3. Desarrollo Académico Diapositiva por Diapositiva

Diapositiva 01: Concepto de Linealidad: Principios de Superposición y Homogeneidad

Condiciones Matemáticas de un Sistema Lineal e Invariante en el Tiempo (LTI)

- **Turno 1 (Masculina):** Toda la teoría clásica de funciones de transferencia en Laplace y diagramas de Bode exige un requisito matemático indispensable: que el modelo del sistema sea lineal e invariante en el tiempo.
- **Turno 2 (Femenina):** Decimos que un sistema dinámico es lineal cuando cumple el principio de superposición, el cual engloba dos propiedades simultáneas: la propiedad de aditividad y la propiedad de homogeneidad o escalamiento.
- **Turno 3 (Masculina):** La aditividad significa que si una entrada u uno produce por separado la respuesta ye uno, y otra entrada u dos produce la respuesta ye dos, entonces aplicar la suma u uno más u dos debe producir exactamente la suma de respuestas ye uno más ye dos.
- **Turno 4 (Femenina):** La homogeneidad exige que si multiplicamos la entrada por cualquier constante escalar alfa, la respuesta quede multiplicada exactamente por esa misma constante alfa; y la invariancia temporal implica que los parámetros físicos como masa, resistencia o capacitancia no cambian con el paso del reloj.

Diapositiva 02: No Linealidades en Sistemas Mecatrónicos Reales

No Linealidades Suaves (Seno, Raíz Cuadrada, Arrastre Cuadrático) vs. Duras (Saturación, Zona Muerta, Coulomb)

- **Turno 1 (Masculina):** En la naturaleza y en la industria ningún componente físico es estrictamente lineal en un rango infinito: todo sistema real presenta fenómenos no lineales que clasificamos en dos grandes categorías.
- **Turno 2 (Femenina):** La primera categoría son las no linealidades suaves o continuamente diferenciables: por ejemplo, el par gravitatorio de un brazo robótico proporcional al seno del ángulo, el caudal en un orificio proporcional a la raíz cuadrada de la presión, o la fuerza aerodinámica proporcional al cuadrado de la velocidad.
- **Turno 3 (Masculina):** La segunda categoría son las no linealidades duras o discontinuas: la saturación de voltaje en un amplificador operacional o puente hache, la zona muerta en el centro de una válvula, la fricción seca de Coulomb y el juego mecánico o backlash entre dientes de engranes.
- **Turno 4 (Femenina):** Mientras las no linealidades duras deben compensarse con estrategias específicas de control o diseño mecánico, las no linealidades suaves se pueden transformar con altísima precisión en modelos lineales locales alrededor de su punto normal de trabajo.

Diapositiva 03: Punto de Operación o Equilibrio Estacionario y Variables de Desviación

Cálculo de (x_0, u_0) Anulando Derivadas y Definición de Incrementos Δx(t) = x(t) - x_0

- **Turno 1 (Masculina):** La idea central de la linealización en ingeniería es que cualquier curva suave vista con una lupa en una vecindad pequeña alrededor de un punto se confunde prácticamente con su recta tangente.
- **Turno 2 (Femenina):** Ese centro de referencia se denomina punto de operación o punto de equilibrio estacionario, denotado por equis sub cero y u sub cero, que corresponde al régimen permanente donde la máquina trabaja normalmente.
- **Turno 3 (Masculina):** Para hallar matemáticamente el punto de equilibrio en una ecuación diferencial no lineal, igualamos a cero todas las derivadas respecto al tiempo, tanto velocidades como aceleraciones, y despejamos el valor estacionario equis sub cero correspondiente a la entrada constante u sub cero.
- **Turno 4 (Femenina):** A partir de ese instante definimos nuestras nuevas variables incrementales o de desviación: delta equis de te es igual a la variable real equis de te menos su valor de equilibrio equis sub cero, y delta u de te es la entrada real menos u sub cero.

Diapositiva 04: Linealización Matemática mediante Expansión en Series de Taylor

Aproximación Tangencial de Primer Orden $\Delta y \approx (df/dx)|_{x_0} \cdot \Delta x$ y Truncamiento de Orden Superior

- **Turno 1 (Masculina):** Sea una función no lineal suave ye igual a efe de equis. Si la desarrollamos en serie de potencias de Taylor alrededor del punto de operación equis sub cero, obtenemos una suma infinita de derivadas evaluadas en equis sub cero.
- **Turno 2 (Femenina):** El primer término es efe de equis sub cero, que es exactamente ye sub cero; el segundo término es la primera derivada de efe respecto a equis evaluada en equis sub cero, multiplicada por equis menos equis sub cero; y luego siguen los términos con el cuadrado y el cubo de la desviación.
- **Turno 3 (Masculina):** Si las excursiones delta equis alrededor del punto de operación son pequeñas, por ejemplo una centésima, su cuadrado vale una diezmilésima y su cubo una millonésima, por lo que podemos truncar y despreciar todos los términos de grado dos o superior.
- **Turno 4 (Femenina):** Al pasar ye sub cero restando al miembro izquierdo nos queda la ley linealizada: el incremento delta ye es igual a una constante efectiva ka sub lin multiplicada por el incremento delta equis, donde ka sub lin es simplemente la pendiente o derivada evaluada en el punto de operación.

Diapositiva 05: Linealización de Funciones de Múltiples Variables (Derivadas Parciales)

Expansión de Taylor Multivariable $\Delta z \approx (\partial f/\partial x)|_0 \cdot \Delta x + (\partial f/\partial u)|_0 \cdot \Delta u$

- **Turno 1 (Masculina):** En muchos dispositivos mecatrónicos una magnitud física depende simultáneamente de dos o más variables: por ejemplo, el caudal de una servoválvula depende tanto de la apertura de la corredera equis como de la presión de carga pe.
- **Turno 2 (Femenina):** Cuando tenemos una función no lineal de dos variables, zeta igual a efe de equis coma u, la expansión de Taylor de primer orden reemplaza la superficie curva por el plano tangente en el punto de equilibrio equis sub cero coma u sub cero.
- **Turno 3 (Masculina):** La desviación delta zeta se aproxima entonces como la derivada parcial de efe respecto a equis evaluada en el punto de operación multiplicada por delta equis, más la derivada parcial de efe respecto a u evaluada en el punto de operación multiplicada por delta u.
- **Turno 4 (Femenina):** En sistemas de estado con múltiples entradas y múltiples salidas, el conjunto de todas estas derivadas parciales evaluadas en el equilibrio forma las matrices jacobianas constantes A, B, C y D del modelo linealizado en el espacio de estados.

Diapositiva 06: Ejemplos Clásicos de Linealización: Péndulo Robótico y Válvula Turbulenta

Linealización de $mgL \cdot sen(\theta)$ en Equilibrio Vertical/Inclinado y de $q = K_v \cdot \sqrt{(\Delta p)}$ en un Tanque

- **Turno 1 (Masculina):** Veamos dos ejemplos fundamentales. Primero, un brazo robótico o péndulo de masa eme y longitud ele gobernado por la ecuación no lineal: eme por ele cuadrado por teta dos puntos, más ce por teta punto, más eme por ge por ele por el seno de teta, igual al par aplicado.
- **Turno 2 (Femenina):** La derivada de la función seno de teta es el coseno de teta. Si operamos cerca de la vertical inferior teta sub cero igual a cero, el coseno de cero vale uno y el término no lineal se reemplaza por eme ge ele por teta; si operamos en un ángulo inclinado teta sub cero, la rigidez gravitatoria efectiva queda multiplicada por el coseno de teta sub cero.
- **Turno 3 (Masculina):** Segundo ejemplo: una válvula de orificio turbulento donde el caudal cu es igual a una constante ka por la raíz cuadrada de la caída de presión delta pe, o lo que es igual, delta pe es proporcional al caudal al cuadrado.
- **Turno 4 (Femenina):** Derivando la caída de presión respecto al caudal en el punto de operación cu sub cero, obtenemos que la resistencia hidráulica linealizada erre sub hache vale dos veces la caída de presión nominal dividida entre el caudal nominal cu sub cero.

Diapositiva 07: Sistemas Hidromecánicos: Servoválvula de Carrete y Cilindro de Doble Efecto

Arquitectura del Servoactuador Electrohidráulico de 4 Vías con Traslape Cero (Zero-Lapped Spool)

- **Turno 1 (Masculina):** Apliquemos ahora la linealización multivariable para modelar uno de los actuadores más potentes de la ingeniería mecatrónica: el sistema hidromecánico formado por una servoválvula de corredera de cuatro vías y un cilindro hidráulico de doble efecto.
- **Turno 2 (Femenina):** La servoválvula posee un carrete o corredera rectificada con traslape nulo, lo que significa que en la posición central equis sub ve igual a cero los bordes de los émbolos cubren exactamente las lumbreras de trabajo sin zona muerta.
- **Turno 3 (Masculina):** Al desplazar la corredera una distancia positiva equis sub ve hacia la derecha, se abre el paso desde la presión de suministro pe sub ese hacia la cámara uno del cilindro, mientras la cámara dos se conecta simultáneamente con el retorno a tanque.
- **Turno 4 (Femenina):** La diferencia de presiones entre ambas cámaras del cilindro, pe sub ele igual a pe uno menos pe dos, actúa sobre el área efectiva del pistón A sub pe para empujar el vástago y mover la masa de carga eme.

Diapositiva 08: Linealización de la Servoválvula: Ganancia de Flujo K_q y Coeficiente K_c

Deducción de la Ecuación Linealizada del Caudal de Carga $q_L = K_q \cdot x_v - K_c \cdot p_L$

- **Turno 1 (Masculina):** A través de los orificios variables de la corredera, el caudal de carga cu sub ele depende de forma no lineal del desplazamiento equis sub ve multiplicado por la raíz cuadrada de la presión de suministro menos la presión diferencial de carga pe sub ele.
- **Turno 2 (Femenina):** Aplicando la expansión en serie de Taylor de dos variables alrededor del punto de operación central, el caudal de carga linealizado se expresa como cu sub ele igual a ka sub cu por equis sub ve, menos ka sub ce por pe sub ele.
- **Turno 3 (Masculina):** El primer coeficiente, ka sub cu, es la derivada parcial del caudal respecto a la posición de la corredera a presión constante, y se denomina ganancia de caudal o sensibilidad de flujo de la válvula, en metros cuadrados por segundo.
- **Turno 4 (Femenina):** El segundo coeficiente, ka sub ce, es menos la derivada parcial del caudal respecto a la presión de carga a apertura constante, llamado coeficiente de flujo-presión: indica cuánto disminuye el caudal cuando la carga se opone al movimiento y amortigua naturalmente el sistema.

Diapositiva 09: Ecuación de Continuidad en el Cilindro: Fugas y Compresibilidad del Aceite

Balance Volumétrico $q_L = A_p \cdot (dy/dt) + C_{tp} \cdot p_L + (V_t / 4\beta) \cdot (dp_L/dt)$

- **Turno 1 (Masculina):** Una vez que el caudal cu sub ele sale de la servoválvula y entra a las cámaras del cilindro hidráulico, ¿en qué se emplea físicamente ese flujo de aceite?
- **Turno 2 (Femenina):** Por la ecuación de continuidad y conservación de masa, el caudal de carga se divide en tres componentes simultáneas: el caudal de desplazamiento del pistón, el caudal de fugas laminares y el caudal de compresibilidad elástica.
- **Turno 3 (Masculina):** La primera componente es el área efectiva del pistón A sub pe multiplicada por la velocidad del vástago, derivada de ye respecto al tiempo; la segunda componente son las fugas internas entre cámaras, proporcionales a un coeficiente de fuga total ce sub te pe por la presión de carga.
- **Turno 4 (Femenina):** Y la tercera componente, crucial en servocilindros de alta frecuencia, modela el resorte hidráulico del aceite atrapado: es igual al volumen total del cilindro ve sub te dividido entre cuatro veces el módulo de elasticidad volumétrico beta del aceite, multiplicado por la derivada de la presión de carga.

Diapositiva 10: Modelo Dinámico Completo e Integrador Hidromecánico Ideal

Acoplamiento con la Masa de Carga $A_p \cdot p_L = m \cdot d^2y/dt^2 + c \cdot dy/dt$ y Reducción a Integrador Puro

- **Turno 1 (Masculina):** Para cerrar el modelo hidromecánico acoplamos la segunda ley de Newton sobre el vástago del cilindro: la fuerza hidráulica neta, área del pistón A sub pe por la presión de carga pe sub ele, es igual a masa por aceleración ye dos puntos, más amortiguamiento viscoso ce por ye punto, más la fuerza externa.
- **Turno 2 (Femenina):** Al igualar la ecuación linealizada de la válvula con la ecuación de continuidad y sustituir la presión de carga desde la ecuación de Newton, obtenemos la ecuación diferencial lineal de tercer orden que gobierna la posición ye del servocilindro.
- **Turno 3 (Masculina):** En aplicaciones de velocidad moderada donde el aceite se puede considerar incompresible, las fugas son despreciables y la presión de suministro es mucho mayor que la caída en la carga, el término de caudal se simplifica a área del pistón por la derivada de ye igual a ka sub cu por equis sub ve.
- **Turno 4 (Femenina):** Esto revela una propiedad fundamental: la velocidad del cilindro es directamente proporcional a la apertura de la válvula, por lo que la posición del cilindro es la integral en el tiempo del desplazamiento de la corredera, comportándose como un integrador hidromecánico puro.

4. Firmware C++/Arduino: Integrador RK4 y Filtro de Tustin (simulacion_hil_y_adquisicion_video_04.ino)

```
/*
 * ========================================================================
 * CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
 * VIDEO 04: Video 04 — Parte 2B: Linealidad, Linealización por Series de Taylor y Sistemas Hidromecánicos
 * Archivo: simulacion_hil_y_adquisicion_video_04.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 "Linealización Jacobiana por Taylor y Actuador Hidromecánico Servoválvula-Cilindro",
 *   filtro discreto por Transformada Bilineal de Tustin y adquisición por ADC
 *   para identificación experimental de parámetros (K, tau, wn, zeta).
 *
 * Ecuación Modelo:
 *   Δy ≈ (∂f/∂x)|_0·Δx + (∂f/∂u)|_0·Δu   |   q_L = K_q·x_v - K_c·p_L = A_p·y' + C_tp·p_L + (V_t/4β)·p_L'
 * ========================================================================
 */

#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 04) ---
const float M_EQ       = 1.0000f;   // Masa / Inercia equivalente
const float C_EQ       = 14.0000f;   // Amortiguamiento / Resistencia equivalente
const float K_EQ       = 400.0000f;   // Rigidez / Elasticidad equivalente
const float WN_RAD_S   = 20.000f;  // Frecuencia natural no amortiguada [rad/s]
const float ZETA_REL   = 0.350f; // Factor de amortiguamiento relativo [-]
const float K_GAIN     = 0.300f;   // Ganancia estática de la planta
const float TAU_SEC    = 0.140f; // 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 04"));
  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_04.py)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
================================================================================
CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
VIDEO 04: Video 04 — Parte 2B: Linealidad, Linealización por Series de Taylor y Sistemas Hidromecánicos
Archivo: modelado_laplace_y_bode_video_04.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 04: Linealización Jacobiana por Taylor y Actuador Hidromecánico Servoválvula-Cilindro
M_EQ     = 1
C_EQ     = 14
K_EQ     = 400
WN_RAD   = 20
ZETA     = 0.35
K_STATIC = 0.3
TAU_SEC  = 0.14


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 04")
    print(" Planta:", "Linealización Jacobiana por Taylor y Actuador Hidromecánico Servoválvula-Cilindro")
    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_04.m)

```
% ========================================================================
% CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
% VIDEO 04: Video 04 — Parte 2B: Linealidad, Linealización por Series de Taylor y Sistemas Hidromecánicos
% Archivo: analisis_control_mecatronico_video_04.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 04\n');
fprintf(' Planta: Linealización Jacobiana por Taylor y Actuador Hidromecánico Servoválvula-Cilindro\n');
fprintf('========================================================================\n\n');

%% 1. Definicion de Parametros Fisicos y Canónicos
m_eq   = 1;
c_eq   = 14;
k_eq   = 400;
wn     = 20;
zeta   = 0.35;
K_gain = 0.3;
tau    = 0.14;

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 = [120.0];
den_G2 = [1.0, 14.0, 400.0, 0.0];
G2 = tf(num_G2, den_G2);
fprintf('--- Sistema Principal del Video 04: 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 04 - 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 04 - 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 04) ===\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);
```