CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA

Video 09 — Parte 5A: Respuesta en Frecuencia: Entrada Senoidal y Fasores — Régimen Permanente Senoidal, Fasores Complejos, Sustitución s = j·ω, Magnitud |G(jω)|, Fase ∠G(jω) y Resonancia

Parte del Curso: Parte 5A — Respuesta en Frecuencia: Entrada Senoidal y Fasores
Planta / Sistema Modelado: Función de Transferencia Senoidal G(jω), Álgebra Fasorial y Fenómeno de Resonancia

$$G(j\omega) = |G(j\omega)| \angle \varphi(\omega) \mid \omega_r = \omega_n \cdot \sqrt{(1 - 2\zeta^2)} \ (\text{para } \zeta < 0.707), \ M_r = 1 / (2\zeta \cdot \sqrt{(1 - \zeta^2)})$$

1. Síntesis de Conceptos Clave y Fórmulas de Ingeniería

- Respuesta en estado estacionario ante entrada armónica u(t) = A·sen(ω·t): salida senoidal de igual frecuencia y_ss(t) = A·|G(jω)|·sen(ω·t + φ(ω)).
- Representación por fasores complejos mediante la identidad de Euler y obtención directa de G(jω) sustituyendo s = j·ω en G(s).
- Sistema de 1.er orden en frecuencia: $|G(j\omega)| = K / \sqrt{(1 + (\omega\tau)^2)}$, φ(ω) = -arctan(ωτ); en la frecuencia de corte ω_c = 1/τ la ganancia cae a 0.707·K (-3 dB) y la fase es -45°.
- Sistema de 2.º orden en frecuencia: en ω = ω_n el desfase es siempre -90° exactos y $|G(j\omega_n)| = K/(2\zeta)$; para ζ < 0.707 aparece el pico de resonancia M_r en $\omega_r = \omega_n \cdot \sqrt{(1 - 2\zeta^2)}$.

2. Parámetros Nominales del Modelo Matemático (Video 09)

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`	6	Elemento disipador de energía que gobierna la tasa de atenuación
Rigidez / Elasticidad Equivalente	`k_eq / 1/C`	144	Almacenador de energía potencial restauradora
Frecuencia Natural No Amortiguada	`ω_n [rad/s]`	12 rad/s	Distancia radial de los polos al origen en el plano complejo s
Factor de Amortiguamiento Relativo	`ζ [-]`	0.25	Determina el régimen dinámico y el sobreimpulso máximo M_p
Ganancia Estática / Constante de Tiempo	`K / τ [s]`	K = 1, τ = 0.333 s	Sensibilidad estacionaria y rapidez del modo exponencial e^{-t/τ}

3. Desarrollo Académico Diapositiva por Diapositiva

Diapositiva 01: Concepto de Respuesta en Frecuencia en Sistemas Mecatrónicos

Excitación Senoidal u(t) = A·sen(ω·t) y Salida Estacionaria y_ss(t) = A·|G(jω)|·sen(ω·t + φ)

- **Turno 1 (Masculina):** Bienvenidos a la Parte cinco del curso, donde estudiaremos la respuesta en frecuencia: el método fundamental para analizar vibraciones, diseñar filtros, identificar plantas complejas y garantizar márgenes de estabilidad.
- **Turno 2 (Femenina):** Hasta ahora excitábamos al sistema con escalones o impulsos; en el análisis en frecuencia le inyectamos una señal senoidal pura de amplitud constante A y frecuencia angular omega en radianes por segundo: u de te igual a A por el seno de omega por te.
- **Turno 3 (Masculina):** En todo sistema lineal e invariante en el tiempo que sea estable, una vez que se extingue el breve transitorio inicial, la salida en estado estacionario es rigurosamente otra onda senoidal de exactamente la misma frecuencia omega.
- **Turno 4 (Femenina):** Lo único que cambia entre la onda que entra y la onda que sale son dos cantidades que dependen de la frecuencia omega: la relación de amplitudes eme de omega, que amplifica o atenúa la señal, y el ángulo de fase fi de omega, que adelanta o atrasa la onda en el tiempo.

Diapositiva 02: Representación de Señales Armónicas mediante Fasores Complejos

Fórmula de Euler e^{jθ} = cos(θ) + j·sen(θ), Magnitud, Argumento y Álgebra Fasorial

- **Turno 1 (Masculina):** Para no tener que operar con identidades trigonométricas engorrosas de senos y cosenos, representamos cada onda senoidal en régimen permanente mediante un vector rotatorio en el plano complejo llamado fasor.
- **Turno 2 (Femenina):** Gracias a la fórmula de Euler, e elevado a jota por teta es igual al coseno de teta más jota por el seno de teta; así, una onda senoidal A por el seno de omega te más fi queda representada por el fasor complejo de módulo A y ángulo fi.
- **Turno 3 (Masculina):** Cuando un fasor atraviesa un sistema lineal, el fasor de salida ye es simplemente el producto del fasor de entrada u por un número complejo propio del sistema a esa frecuencia.
- **Turno 4 (Femenina):** Por las reglas de los números complejos en forma polar, el módulo de la salida es el producto de los módulos, y el ángulo de fase de la salida es la suma del ángulo de entrada más el ángulo introducido por el sistema.

Diapositiva 03: La Función de Transferencia Senoidal G(jω) (Sustitución s = j·ω)

Evaluación de G(s) sobre el Eje Imaginario Positivo y Cociente de Fasores Y(jω) / U(jω)

- **Turno 1 (Masculina):** ¿Cómo obtenemos matemáticamente ese número complejo que caracteriza la respuesta en frecuencia de nuestra planta mecatrónica sin tener que resolver de nuevo la ecuación diferencial?
- **Turno 2 (Femenina):** Se demuestra mediante fracciones parciales que cuando el tiempo tiende a infinito y los polos del sistema están en el semiplano izquierdo, la respuesta estacionaria ante una entrada senoidal viene dada por la función de transferencia de Laplace evaluada en ese igual a jota por omega.
- **Turno 3 (Masculina):** Es decir, basta tomar nuestra función de transferencia conocida ge de ese y sustituir en todas partes la variable compleja ese por jota omega, obteniendo la llamada función de transferencia senoidal ge de jota omega.
- **Turno 4 (Femenina):** Geométricamente, esto equivale a recorrer con un instrumento de medición el eje imaginario vertical positivo del plano complejo desde frecuencia omega igual a cero hasta frecuencia infinita.

Diapositiva 04: Cálculo Analítico de Magnitud M(ω) = |G(jω)| y Fase φ(ω) = ∠G(jω)

Parte Real e Imaginaria, Módulo de Cocientes de Factores y Suma/Resta de Ángulos Arctan

- **Turno 1 (Masculina):** Para cada valor numérico de la frecuencia omega, la función ge de jota omega es un número complejo que posee una parte real y una parte imaginaria, o expresado en forma polar, una magnitud y un ángulo de fase.
- **Turno 2 (Femenina):** La magnitud eme de omega, o módulo de ge de jota omega, es la raíz cuadrada de la parte real al cuadrado más la parte imaginaria al cuadrado; si ge está factorizada, es el producto de los módulos del numerador dividido entre el producto de los módulos del denominador.
- **Turno 3 (Masculina):** Esto significa que si la entrada tiene amplitud A, la onda senoidal de salida tendrá en estado estacionario una amplitud igual a A multiplicada por el módulo de ge de jota omega.
- **Turno 4 (Femenina):** Por su parte, el ángulo de fase fi de omega es el arcotangente de la parte imaginaria sobre la parte real; en una función factorizada, es simplemente la suma de los ángulos de los factores del numerador menos la suma de los ángulos de los factores del denominador.

Diapositiva 05: Respuesta en Frecuencia de Bloques Elementales: Ganancia, Integrador y Derivador

Análisis de G(jω) = K, Integrador Puro 1/(jω) (-90° fijo) y Derivador Puro jω (+90° fijo)

- **Turno 1 (Masculina):** Analicemos primero la respuesta en frecuencia de los tres bloques más simples: una ganancia proporcional constante ka, un derivador ideal ese y un integrador ideal uno sobre ese.
- **Turno 2 (Femenina):** En un bloque de ganancia positiva ka no aparece la variable ese, por lo que su magnitud vale ka para todas las frecuencias desde cero hasta infinito, y su desfase es idénticamente cero grados.
- **Turno 3 (Masculina):** En un derivador ideal ge de ese igual a ese, al sustituir ese por jota omega obtenemos magnitud igual a omega y un factor jota positivo en el eje imaginario superior: el derivador amplifica linealmente las altas frecuencias y adelanta siempre la fase en más noventa grados exactos.
- **Turno 4 (Femenina):** En cambio, en un integrador ideal uno sobre ese, al sustituir uno sobre jota omega y multiplicar por jota arriba y abajo obtenemos menos jota por uno sobre omega: su magnitud cae en proporción inversa a la frecuencia y atrasa siempre la onda en menos noventa grados.

Diapositiva 06: Respuesta en Frecuencia de Sistemas de Primer Orden: G(jω) = K / (1 + j·ω·τ)

Magnitud K / √(1 + $\omega^2\tau^2$), Fase -arctan(ωτ) y Frecuencia de Corte o Esquina $\omega_c = 1/\tau$

- **Turno 1 (Masculina):** Estudiemos ahora la respuesta en frecuencia de un sistema de primer orden cuya función de transferencia es ka dividido entre tau por ese más uno; al hacer ese igual a jota omega obtenemos ka sobre uno más jota por omega por tau.
- **Turno 2 (Femenina):** El módulo del numerador es ka, y el módulo del denominador uno más jota omega tau es la raíz cuadrada de uno más el cuadrado del producto omega por tau; esa división nos da la relación de amplitudes exacta para cualquier frecuencia.
- **Turno 3 (Masculina):** El ángulo del numerador es cero grados y el ángulo del denominador es el arcotangente de la parte imaginaria omega tau dividida entre la parte real uno; como está en el denominador, pasa restando y el desfase vale menos el arcotangente de omega por tau.
- **Turno 4 (Femenina):** Observemos qué ocurre en la frecuencia clave omega igual a uno sobre tau, llamada frecuencia de esquina o de corte: allí omega por tau vale uno, la magnitud cae a ka sobre raíz de dos, es decir el setenta coma siete por ciento, y el atraso de fase es exactamente de menos cuarenta y cinco grados.

Diapositiva 07: Filtro Pasabajos de Primer Orden: Bajas, Medias y Altas Frecuencias

Transmisión Fiel para $\omega \ll 1/\tau$, Atenuación Progresiva y Límite de Fase de -90° para $\omega \gg 1/\tau$

- **Turno 1 (Masculina):** Interpretemos físicamente cómo se comporta este sistema de primer orden, ya sea un filtro erre ce, un motor o un sensor térmico, en tres zonas del espectro de frecuencias.
- **Turno 2 (Femenina):** En la zona de bajas frecuencias, cuando omega es mucho menor que uno sobre tau, el término omega por tau tiende a cero: la magnitud es igual a ka y el desfase es cero grados, por lo que la salida reproduce fielmente y sin retraso la onda de entrada.
- **Turno 3 (Masculina):** Al llegar a la frecuencia de corte omega sub ce igual a uno sobre tau, la potencia de la señal se reduce exactamente a la mitad, lo que en ingeniería electrónica y de control se conoce como el punto de caída de menos tres decibelios.
- **Turno 4 (Femenina):** Y en la zona de altas frecuencias, cuando omega es mucho mayor que uno sobre tau, la inercia o capacitancia del sistema le impide seguir cambios tan rápidos: la amplitud cae hacia cero como ka sobre omega tau, y el desfase se acerca asintóticamente a un tope máximo de menos noventa grados.

Diapositiva 08: Respuesta en Frecuencia de Sistemas de Segundo Orden

Deducción de $|G(j\omega)| = K / \sqrt{((1 - u^2)^2 + (2\zeta u)^2)}$ y Fase $\varphi(\omega)$ con $u = \omega / \omega_n$

- **Turno 1 (Masculina):** Pasemos al análisis en frecuencia de un sistema de segundo orden haciendo ese igual a jota omega en su función de transferencia canónica. Recordemos que jota al cuadrado vale menos uno, por lo que ese al cuadrado se convierte en menos omega al cuadrado.
- **Turno 2 (Femenina):** Si dividimos numerador y denominador por omega sub ene al cuadrado y definimos la razón de frecuencia adimensional u igual a omega sobre omega sub ene, la función toma la forma compacta: ka dividido entre uno menos u al cuadrado, más jota por dos dseta por u.
- **Turno 3 (Masculina):** El módulo de esta expresión es igual a la ganancia ka dividida entre la raíz cuadrada de uno menos u al cuadrado todo al cuadrado, más dos por dseta por u todo al cuadrado.
- **Turno 4 (Femenina):** Y el ángulo de fase fi de omega es menos el arcotangente de dos cuadrantes de la parte imaginaria, dos dseta por u, respecto a la parte real, uno menos u al cuadrado.

Diapositiva 09: Resonancia en Sistemas de Segundo Orden: Frecuencia ω_r y Pico M_r

Condición de Resonancia $\zeta < 0.707$, Frecuencia $\omega_r = \omega_n \cdot \sqrt{(1 - 2\zeta^2)}$ y Desfase de -90° en ω_n

- **Turno 1 (Masculina):** En los sistemas de segundo orden aparece un fenómeno físico fascinante y potencialmente destructivo si no se controla: la resonancia mecánica o eléctrica cuando la frecuencia de excitación se acerca a la frecuencia natural.
- **Turno 2 (Femenina):** Fijémonos primero qué ocurre exactamente cuando la frecuencia de entrada omega iguala a la frecuencia natural omega sub ene, es decir u igual a uno: la parte real uno menos u cuadrado se hace cero para cualquier amortiguamiento, el desfase pasa siempre por menos noventa grados exactos, y la magnitud vale ka dividido entre dos dseta.
- **Turno 3 (Masculina):** Al derivar el denominador respecto a la frecuencia encontramos que el pico máximo de amplitud ocurre en la frecuencia de resonancia omega sub erre, igual a omega sub ene por la raíz cuadrada de uno menos dos veces dseta al cuadrado.
- **Turno 4 (Femenina):** Como el interior de esa raíz solo es positivo cuando dos dseta al cuadrado es menor que uno, solo existe pico de resonancia si dseta es menor que uno sobre raíz de dos, es decir cero punto setecientos siete; y en ese pico la amplificación máxima eme sub erre vale uno dividido entre dos dseta por la raíz de uno menos dseta al cuadrado.

Diapositiva 10: Ensayo Experimental de Barrido Senoidal (Sine Sweep) en Laboratorio

Medición de Relación de Amplitudes y Desfase Temporal $\varphi = -360° \cdot f \cdot \Delta t$ en Osciloscopio / DAQ

- **Turno 1 (Masculina):** Cerramos la Parte cinco A explicando cómo se mide experimentalmente la respuesta en frecuencia de una máquina mecatrónica mediante un ensayo de barrido senoidal o sine sweep.
- **Turno 2 (Femenina):** Programamos el microcontrolador o generador de funciones para inyectar al actuador una sinusoide de frecuencia fija efe uno, esperamos unos segundos a que desaparezca el transitorio y registramos en estado estacionario las ondas de entrada y de salida.
- **Turno 3 (Masculina):** En esa frecuencia dividimos la amplitud pico de la salida entre la amplitud pico de la entrada para obtener el módulo; y medimos el retraso temporal delta te entre los cruces por cero de ambas ondas, multiplicándolo por menos trescientos sesenta grados por la frecuencia efe para hallar la fase.
- **Turno 4 (Femenina):** Al repetir automáticamente este proceso para decenas de frecuencias espaciadas logarítmicamente, construimos la firma frecuencial completa de la planta, lista para graficarse en los diagramas de Bode que estudiaremos en el próximo video.

4. Firmware C++/Arduino: Integrador RK4 y Filtro de Tustin (simulacion_hil_y_adquisicion_video_09.ino)

```
/*
 * ============================================================================
 * CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
 * VIDEO 09: Video 09 — Parte 5A: Respuesta en Frecuencia: Entrada Senoidal y Fasores
 * Archivo: simulacion_hil_y_adquisicion_video_09.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 "Función de Transferencia Senoidal G(jω), Álgebra Fasorial y Fenómeno de Resonancia",
 *   filtro discreto por Transformada Bilineal de Tustin y adquisición por ADC
 *   para identificación experimental de parámetros (K, tau, wn, zeta).
 *
 * Ecuación Modelo:
 *   G(jω) = |G(jω)| ∠ ϕ(ω)   |   ω_r = ω_n·√(1 - 2ζ²) (para ζ < 0.707),   M_r = 1 / (2ζ·√(1 - ζ²))
 * ============================================================================
 */

#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 09) ---
const float M_EQ       = 1.0000f;   // Masa / Inercia equivalente
const float C_EQ       = 6.0000f;   // Amortiguamiento / Resistencia equivalente
const float K_EQ       = 144.0000f;   // Rigidez / Elasticidad equivalente
const float WN_RAD_S   = 12.000f;  // Frecuencia natural no amortiguada [rad/s]
const float ZETA_REL   = 0.250f; // Factor de amortiguamiento relativo [-]
const float K_GAIN     = 1.000f;   // Ganancia estática de la planta
const float TAU_SEC    = 0.333f; // 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 09"));
  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_09.py)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
================================================================================
CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
VIDEO 09: Video 09 — Parte 5A: Respuesta en Frecuencia: Entrada Senoidal y Fasores
Archivo: modelado_laplace_y_bode_video_09.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 09: Función de Transferencia Senoidal G(jω), Álgebra Fasorial y Fenómeno de
Resonancia
M_EQ     = 1
C_EQ     = 6
K_EQ     = 144
WN_RAD   = 12
ZETA     = 0.25
K_STATIC = 1
TAU_SEC  = 0.333


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 09")
    print(" Planta:", "Función de Transferencia Senoidal G(jω), Álgebra Fasorial y Fenómeno de Resonancia")
    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_09.m)

```
% ========================================================================
% CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
% VIDEO 09: Video 09 — Parte 5A: Respuesta en Frecuencia: Entrada Senoidal y Fasores
% Archivo: analisis_control_mecatronico_video_09.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 09\n');
fprintf(' Planta: Función de Transferencia Senoidal G(jw), Álgebra Fasorial y Fenómeno de Resonancia\n');
fprintf('========================================================================\n\n');

%% 1. Definicion de Parametros Fisicos y Canónicos
m_eq   = 1;
c_eq   = 6;
k_eq   = 144;
wn     = 12;
zeta   = 0.25;
K_gain = 1;
tau    = 0.333;

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 = [144.0];
den_G2 = [1.0, 6.0, 144.0];
G2 = tf(num_G2, den_G2);
fprintf('--- Sistema Principal del Video 09: 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 09 - 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 09 - 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 09) ===\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);
```