# CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA

**Video 06 — Parte 3B: Sistemas de Segundo Orden, Medidas de Desempeño e Identificación — Frecuencia Natural ω_n, Factor de Amortiguamiento ζ, Sobreimpulso M_p, Tiempo Pico t_p e Identificación Experimental**

**Parte del Curso:** Parte 3B — Sistemas de Segundo Orden, Medidas de Desempeño e Identificación
**Planta / Sistema Modelado:** Sistema Subamortiguado de Segundo Orden e Identificación Experimental de Parámetros

$$y'' + 2\cdot\zeta\cdot\omega_n\cdot y' + \omega_n^2\cdot y = K\cdot\omega_n^2\cdot u(t) \Longrightarrow M_p = e^{-\pi\zeta/\sqrt{(1-\zeta^2)}},\ t_p = \pi/\omega_d,\ t_s = 4/(\zeta\cdot\omega_n)$$

## 1. Síntesis de Conceptos Clave y Fórmulas de Ingeniería

- Clasificación según el factor de amortiguamiento relativo ζ: No amortiguado (ζ = 0), Subamortiguado (0 < ζ < 1), Críticamente amortiguado (ζ = 1) y Sobreamortiguado (ζ > 1).
- Frecuencia natural amortiguada ω_d = ω_n·√(1 - ζ²) y tasa de atenuación exponencial σ = ζ·ω_n.
- Especificaciones exactas de desempeño: Tiempo de subida t_r = (π - arccos ζ)/ω_d, Tiempo pico t_p = π/ω_d, Sobreimpulso máximo M_p = exp(-πζ/√(1-ζ²)) y Tiempo de establecimiento t_s = 4/(ζ·ω_n).
- Identificación experimental: regresión logarítmica para τ en 1.er orden, y extracción de (K, ζ, ω_n) a partir de (y_ss, M_p, t_p) o del decremento logarítmico δ.

## 2. Parámetros Nominales del Modelo Matemático (Video 06)

| 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` | 8 | Elemento disipador de energía que gobierna la tasa de atenuación |
| Rigidez / Elasticidad Equivalente | `k_eq` / `1/C` | 100 | Almacenador de energía potencial restauradora |
| Frecuencia Natural No Amortiguada | `ω_n [rad/s]` | 10 rad/s | Distancia radial de los polos al origen en el plano complejo s |
| Factor de Amortiguamiento Relativo | `ζ [-]` | 0.4 | Determina el régimen dinámico y el sobreimpulso máximo M_p |
| Ganancia Estática / Constante de Tiempo | `K` / `τ [s]` | K = 1, τ = 0.25 s | Sensibilidad estacionaria y rapidez del modo exponencial e^{-t/τ} |

## 3. Desarrollo Académico Diapositiva por Diapositiva

### Diapositiva 01: Sistemas de Segundo Orden: Forma Estándar Canónica (ω_n, ζ y K)

Ecuación Diferencial $d^2y/dt^2 + 2\cdot\zeta\cdot\omega_n\cdot(dy/dt) + \omega_n^2\cdot y = K\cdot\omega_n^2\cdot u(t)$

- **Turno 1 (Masculina):** En la Parte tres B entramos al análisis de los sistemas dinámicos más importantes de toda la ingeniería mecatrónica: los sistemas de segundo orden.
- **Turno 2 (Femenina):** A diferencia de los sistemas de primer orden que solo almacenan un tipo de energía, un sistema de segundo orden posee dos almacenes complementarios, como masa y resorte en mecánica, o inductor y capacitor en electricidad.
- **Turno 3 (Masculina):** Al dividir la ecuación diferencial entre el coeficiente de la segunda derivada, todo sistema de segundo orden adopta la forma canónica: ye dos puntos, más dos por dseta por omega sub ene por ye punto, más omega sub ene al cuadrado por ye, igual a ka por omega sub ene al cuadrado por la entrada.
- **Turno 4 (Femenina):** Aquí ka es la ganancia estática, omega sub ene es la frecuencia natural no amortiguada en radianes por segundo, y la letra griega dseta es el factor de amortiguamiento relativo adimensional.

### Diapositiva 02: Frecuencia Natural No Amortiguada (ω_n) y Factor de Amortiguamiento (ζ)

Deducción en Masa-Resorte-Amortiguador ($\omega_n = \sqrt{(k/m)}$, $\zeta = c / 2\sqrt{(mk)}$) y Circuito RLC

- **Turno 1 (Masculina):** Veamos cómo se relacionan omega sub ene y dseta con los componentes reales de un mecanismo masa-resorte-amortiguador gobernado por masa por equis dos puntos, más ce por equis punto, más ka por equis igual a la fuerza.
- **Turno 2 (Femenina):** Al dividir entre la masa eme, el coeficiente de equis es ka sobre eme, por lo que la frecuencia natural no amortiguada omega sub ene es igual a la raíz cuadrada de la rigidez ka dividida entre la masa eme.
- **Turno 3 (Masculina):** El amortiguamiento crítico ce sub ce erre se define como dos veces la raíz cuadrada de masa por rigidez; y el factor de amortiguamiento relativo dseta es el cociente entre el amortiguamiento real ce y el amortiguamiento crítico.
- **Turno 4 (Femenina):** De manera idéntica, en un circuito eléctrico erre ele ce en serie, la frecuencia natural omega sub ene vale uno sobre la raíz cuadrada de ele por ce, y el factor de amortiguamiento dseta vale la mitad de la resistencia erre multiplicada por la raíz de ce sobre ele.

### Diapositiva 03: Ecuación Característica y los Cuatro Regímenes de Amortiguamiento

Raíces $s_{1,2} = -\zeta\cdot\omega_n \pm \omega_n\cdot\sqrt{(\zeta^2 - 1)}$: No Amortiguado, Subamortiguado, Crítico y Sobreamortiguado

- **Turno 1 (Masculina):** El comportamiento dinámico de cualquier sistema de segundo orden depende por completo de las raíces de su polinomio característico: ese al cuadrado, más dos por dseta por omega sub ene por ese, más omega sub ene al cuadrado igual a cero.
- **Turno 2 (Femenina):** Aplicando la fórmula cuadrática, las dos raíces características son: menos dseta por omega sub ene, más o menos omega sub ene por la raíz cuadrada de dseta al cuadrado menos uno.
- **Turno 3 (Masculina):** El signo del discriminante dseta al cuadrado menos uno clasifica la respuesta en cuatro casos físicos: si dseta es igual a cero el sistema es no amortiguado; si dseta está entre cero y uno es subamortiguado.
- **Turno 4 (Femenina):** Si dseta es exactamente igual a uno el sistema es críticamente amortiguado; y si dseta es mayor que uno el sistema es sobreamortiguado.

### Diapositiva 04: Régimen Subamortiguado (0 < ζ < 1) y Frecuencia Amortiguada (ω_d)

Raíces Complejas Conjugadas s = -σ ± j·ω_d y Envolvente Exponencial e^{-ζ·ω_n·t}

- **Turno 1 (Masculina):** El caso más importante en robótica y control de movimiento es el régimen subamortiguado, donde dseta es mayor que cero y menor que uno, por lo que el interior de la raíz cuadrada es negativo.
- **Turno 2 (Femenina):** Al extraer la unidad imaginaria jota, las raíces son complejas conjugadas: menos sigma más o menos jota por omega sub de, donde la tasa de atenuación real sigma es igual al producto de dseta por omega sub ene.
- **Turno 3 (Masculina):** La parte imaginaria omega sub de se denomina frecuencia natural amortiguada, y es igual a omega sub ene multiplicada por la raíz cuadrada de uno menos dseta al cuadrado: es la frecuencia real a la que oscila el sistema durante el transitorio.
- **Turno 4 (Femenina):** Ante un escalón unitario, la salida subamortiguada sube rápidamente, cruza el valor final, alcanza un pico de sobreimpulso y oscila con frecuencia omega sub de mientras su amplitud queda atrapada y extinguida por las envolventes exponenciales e elevado a menos dseta por omega sub ene por te.

### Diapositiva 05: Régimen Críticamente Amortiguado (ζ = 1) y Sobreamortiguado (ζ > 1)

Aproximación Más Rápida Sin Sobreimpulso vs. Respuesta Lenta de Dos Exponenciales Reales

- **Turno 1 (Masculina):** ¿Qué ocurre cuando aumentamos el amortiguamiento hasta alcanzar exactamente el valor umbral dseta igual a uno, conocido como amortiguamiento crítico?
- **Turno 2 (Femenina):** En dseta igual a uno el radical se anula y obtenemos una raíz real doble en menos omega sub ene; la respuesta al escalón contiene el término uno más omega sub ene por te multiplicado por e a la menos omega sub ene por te.
- **Turno 3 (Masculina):** Esta curva críticamente amortiguada tiene una propiedad de oro en ingeniería: proporciona el ascenso más rápido posible hacia la referencia sin presentar ni un solo milímetro de sobreimpulso ni oscilación, ideal para robots quirúrgicos o cabezales de corte láser.
- **Turno 4 (Femenina):** Si aumentamos aún más la fricción de modo que dseta sea mayor que uno, entramos al régimen sobreamortiguado con dos raíces reales negativas distintas: tampoco hay sobreimpulso, pero una de las exponenciales se acerca al origen y vuelve la respuesta lenta y perezosa.

### Diapositiva 06: Medidas de Desempeño de Segundo Orden I: Tiempo de Subida (t_r) y Tiempo Pico (t_p)

Deducción de t_r = (π - arccos(ζ)) / ω_d y del Instante de Máximo Sobreimpulso t_p = π / ω_d

- **Turno 1 (Masculina):** Para especificar con rigor la calidad de una respuesta transitoria subamortiguada utilizamos cuatro medidas cuantitativas de desempeño: el tiempo de subida, el tiempo pico, el sobreimpulso máximo y el tiempo de establecimiento.
- **Turno 2 (Femenina):** En sistemas subamortiguados, el tiempo de subida te sub erre se mide desde el cero hasta el cien por ciento, es decir, el primer instante en que la curva cruza el valor final estacionario: vale pi menos el arcocoseno de dseta, todo dividido entre la frecuencia amortiguada omega sub de.
- **Turno 3 (Masculina):** A continuación, la respuesta continúa subiendo hasta alcanzar su cresta máxima en el tiempo pico te sub pe, donde la derivada temporal de la salida se hace cero por primera vez.
- **Turno 4 (Femenina):** Al derivar la ecuación de respuesta e igualar a cero obtenemos que el seno de omega sub de por te sub pe debe anularse en pi radianes; por tanto, el tiempo pico te sub pe es simplemente pi dividido entre la frecuencia amortiguada omega sub de.

### Diapositiva 07: Medidas de Desempeño II: Sobreimpulso Máximo (M_p) y Tiempo de Establecimiento (t_s)

**Fórmula Exacta $M_p = e^{-\pi \cdot \zeta / \sqrt{(1 - \zeta^2)}}$ y Tiempo de Establecimiento al 2% $t_s = 4 / (\zeta \cdot \omega_n)$**

- **Turno 1 (Masculina):** La tercera especificación crítica es el sobreimpulso máximo eme sub pe, o porcentaje de overshoot, que mide cuánto excede el pico máximo al valor final estacionario expresado en tanto por uno o en porcentaje.
- **Turno 2 (Femenina):** Al evaluar la respuesta en el tiempo pico te sub pe descubrimos una ley fundamental: el sobreimpulso máximo depende única y exclusivamente del factor de amortiguamiento dseta, y vale e elevado a menos pi por dseta dividido entre la raíz cuadrada de uno menos dseta al cuadrado.
- **Turno 3 (Masculina):** Por ejemplo, un factor dseta de cero punto cinco produce siempre un dieciséis coma tres por ciento de sobreimpulso; y un factor dseta de cero punto setecientos siete produce un sobreimpulso mínimo de apenas cuatro coma tres por ciento.
- **Turno 4 (Femenina):** La cuarta medida es el tiempo de establecimiento te sub ese, determinado por el decaimiento de la envolvente exponencial e a la menos dseta por omega sub ene por te: para la banda del dos por ciento vale cuatro dividido entre dseta por omega sub ene, y para el cinco por ciento vale tres sobre dseta por omega sub ene.

### Diapositiva 08: Identificación Experimental de Sistemas: Concepto y Adquisición de Datos

**Modelado de Caja Gris a partir de Pruebas de Escalón en Banco Mecatrónico con DAQ / ADC**

- **Turno 1 (Masculina):** En la práctica industrial muchas veces nos enfrentamos a un motor, un horno o un actuador neumático del cual no tenemos planos de fabricante ni conocemos sus coeficientes internos exactos.
- **Turno 2 (Femenina):** La identificación experimental de sistemas consiste en inyectar una señal de excitación conocida a la planta real, registrar la respuesta temporal con nuestro sistema de adquisición de datos y deducir matemáticamente los parámetros del modelo.
- **Turno 3 (Masculina):** Al observar la forma de la curva adquirida ante un escalón, lo primero es clasificar su estructura: si la salida arranca con pendiente máxima y sube de forma monótona sin inflexión ni oscilación, estamos ante un sistema dominante de primer orden.
- **Turno 4 (Femenina):** Si por el contrario la curva arranca con pendiente horizontal nula en el origen y presenta una inflexión en forma de ese o bien oscila superando el valor final, el sistema debe modelarse como de segundo orden.

### Diapositiva 09: Identificación de Sistemas de Primer Orden y Método de Regresión Logarítmica

**Ganancia $K = \Delta y_{ss} / \Delta u$, Método de Dos Puntos (28.3% y 63.2%) y Ajuste por Mínimos Cuadrados**

- **Turno 1 (Masculina):** Para identificar un sistema de primer orden ante un escalón de amplitud delta u, primero medimos el cambio total en estado estacionario delta ye sub ese ese: la ganancia estática ka es el cociente delta ye sub ese ese dividido entre delta u.
- **Turno 2 (Femenina):** Luego buscamos en la tabla de datos el instante en que la respuesta alcanza el sesenta y tres coma dos por ciento de ese cambio total: ese tiempo es directamente la constante de tiempo tau.
- **Turno 3 (Masculina):** Si además existe un pequeño retardo o tiempo muerto inicial, aplicamos el método de dos puntos de Smith midiendo los instantes al veintiocho coma tres por ciento y al sesenta y tres coma dos por ciento: la constante de tiempo tau es uno coma cinco veces la resta de ambos tiempos.
- **Turno 4 (Femenina):** Y para máxima inmunidad al ruido electrónico del sensor, tomamos el logaritmo natural de uno menos la respuesta normalizada y ajustamos una recta por mínimos cuadrados en Python: la pendiente negativa de esa recta nos entrega exactamente menos uno sobre tau.

**Diapositiva 10: Identificación Experimental de Segundo Orden y Decremento Logarítmico (δ)**

Extracción de ζ desde M_p, Cálculo de ω_n desde t_p y Ensayo de Vibración Libre

- **Turno 1 (Masculina):** Finalmente, ¿cómo identificamos experimentalmente los tres parámetros ka, dseta y omega sub ene de un sistema subamortiguado de segundo orden a partir de su registro de prueba?
- **Turno 2 (Femenina):** En un ensayo de escalón, primero calculamos la ganancia ka dividiendo el valor final estacionario entre el escalón de entrada. Segundo, medimos el valor máximo del primer pico ye sub max y calculamos el sobreimpulso en tanto por uno eme sub pe.
- **Turno 3 (Masculina):** Despejando de la ecuación de sobreimpulso, el factor de amortiguamiento dseta es igual al valor absoluto del logaritmo natural de eme sub pe, dividido entre la raíz cuadrada de pi al cuadrado más el cuadrado del logaritmo natural de eme sub pe.
- **Turno 4 (Femenina):** Tercero, leemos en el eje de tiempo el instante del pico te sub pe y despejamos la frecuencia natural omega sub ene como pi dividido entre te sub pe por la raíz de uno menos dseta al cuadrado; o en vibración libre, usamos el decremento logarítmico delta entre picos consecutivos.

## 4. Firmware C++/Arduino: Integrador RK4 y Filtro de Tustin (simulacion_hil_y_adquisicion_video_06.ino)

```
/*
 * ============================================================================
 * CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
 * VIDEO 06: Video 06 — Parte 3B: Sistemas de Segundo Orden, Medidas de Desempeño e Identificación
 * Archivo: simulacion_hil_y_adquisicion_video_06.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 "Sistema Subamortiguado de Segundo Orden e Identificación Experimental de Parámetros",
 *   filtro discreto por Transformada Bilineal de Tustin y adquisición por ADC
 *   para identificación experimental de parámetros (K, tau, wn, zeta).
 *
 * Ecuación Modelo:
 *   y'' + 2·ζ·ω_n·y' + ω_n²·y = K·ω_n²·u(t)  ==>  M_p = e^{-πζ/√(1-ζ²)},  t_p = π/ω_d,  t_s = 4/(ζ·ω_n)
 * ============================================================================
 */

#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 06) ---
const float M_EQ       = 1.0000f;   // Masa / Inercia equivalente
const float C_EQ       = 8.0000f;   // Amortiguamiento / Resistencia equivalente
const float K_EQ       = 100.0000f;   // Rigidez / Elasticidad equivalente
const float WN_RAD_S   = 10.000f;  // Frecuencia natural no amortiguada [rad/s]
const float ZETA_REL   = 0.400f; // Factor de amortiguamiento relativo [-]
const float K_GAIN     = 1.000f;   // Ganancia estática de la planta
const float TAU_SEC    = 0.250f; // 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 06"));
  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_06.py)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
================================================================================
CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
VIDEO 06: Video 06 — Parte 3B: Sistemas de Segundo Orden, Medidas de Desempeño e Identificación
Archivo: modelado_laplace_y_bode_video_06.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 06: Sistema Subamortiguado de Segundo Orden e Identificación Experimental de
Parámetros
M_EQ     = 1
C_EQ     = 8
K_EQ     = 100
WN_RAD   = 10
ZETA     = 0.4
K_STATIC = 1
TAU_SEC  = 0.25


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 06")
    print(" Planta:", "Sistema Subamortiguado de Segundo Orden e Identificación Experimental de Parámetros")
    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_06.m)

```
% ========================================================================
% CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
% VIDEO 06: Video 06 — Parte 3B: Sistemas de Segundo Orden, Medidas de Desempeño e Identificación
% Archivo: analisis_control_mecatronico_video_06.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 06\n');
fprintf(' Planta: Sistema Subamortiguado de Segundo Orden e Identificación Experimental de Parámetros\n');
fprintf('========================================================================\n\n');

%% 1. Definicion de Parametros Fisicos y Canónicos
m_eq   = 1;
c_eq   = 8;
k_eq   = 100;
wn     = 10;
zeta   = 0.4;
K_gain = 1;
tau    = 0.25;

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 = [100.0];
den_G2 = [1.0, 8.0, 100.0];
G2 = tf(num_G2, den_G2);
fprintf('--- Sistema Principal del Video 06: 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 06 - 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 06 - 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 06) ===\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);
```