# CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA

**Video 05 — Parte 3A: Respuestas Dinámicas: Terminología y Sistemas de Primer Orden — Entradas de Prueba (Escalón, Rampa, Impulso), Respuesta Transitoria vs. Estacionaria, Constante de Tiempo τ y Ganancia K**

**Parte del Curso:** Parte 3A — Respuestas Dinámicas: Terminología y Sistemas de Primer Orden
**Planta / Sistema Modelado:** Planta Canónica de Primer Orden (Térmica / RC / Velocidad de Motor)

$$\tau \cdot (dy/dt) + y = K \cdot u(t) \Longrightarrow y(t) = K \cdot A \cdot (1 - e^{-t/\tau}),\ t_r = 2.2 \cdot \tau,\ t_s(2\%) = 4 \cdot \tau$$

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

- Señales de prueba estándar en control: Impulso de Dirac δ(t), Escalón unitario u_s(t), Rampa r(t) = t y Parábola p(t) = ½t$^2$.
- Descomposición de la respuesta temporal: Respuesta Transitoria y_tr(t) (modos propios exponenciales) + Respuesta en Estado Estacionario y_ss(t).
- Sistema canónico de primer orden τ·y' + y = K·u(t): en t = 1τ alcanza el 63.2% del cambio final; en t = 3τ el 95.0%; en t = 4τ el 98.2%.
- Medidas de desempeño de primer orden: Tiempo de retardo t_d = 0.693·τ, Tiempo de subida (10% a 90%) t_r = 2.2·τ, Tiempo de establecimiento t_s = 4·τ y error en rampa e_ss = v_0·τ.

## 2. Parámetros Nominales del Modelo Matemático (Video 05)

| Parámetro Físico / Canónico | Símbolo | Valor Nominal | Significado en el Modelado Dinámico |
|---|---|---|---|
| Masa / Momento de Inercia Equivalente | `m_eq / J_eq` | 0.5 | Almacenador de energía cinética opositor a cambios de velocidad |
| Amortiguamiento / Resistencia Equivalente | `c_eq / R_eq` | 1 | Elemento disipador de energía que gobierna la tasa de atenuación |
| Rigidez / Elasticidad Equivalente | `k_eq / 1/C` | 2 | Almacenador de energía potencial restauradora |
| Frecuencia Natural No Amortiguada | `ω_n [rad/s]` | 2 rad/s | Distancia radial de los polos al origen en el plano complejo s |
| Factor de Amortiguamiento Relativo | `ζ [-]` | 1 | Determina el régimen dinámico y el sobreimpulso máximo M_p |
| Ganancia Estática / Constante de Tiempo | `K / τ [s]` | K = 2, τ = 0.5 s | Sensibilidad estacionaria y rapidez del modo exponencial e^{-t/τ} |

## 3. Desarrollo Académico Diapositiva por Diapositiva

### Diapositiva 01: Modelado de Sistemas Dinámicos mediante Ecuaciones Diferenciales

**Concepto de Memoria Dinámica, Orden del Sistema y Número de Almacenes de Energía**

- **Turno 1 (Masculina):** Bienvenidos a la Parte tres del curso, donde estudiaremos cómo evolucionan en el tiempo las respuestas dinámicas de los modelos matemáticos que hemos formulado en los módulos anteriores.
- **Turno 2 (Femenina):** A diferencia de un sistema estático o algebraico, cuya salida depende únicamente del valor instantáneo de la entrada en ese mismo microsegundo, un sistema dinámico posee memoria porque almacena energía en sus masas, resortes, capacitores, inductores o tanques.
- **Turno 3 (Masculina):** Como la energía no puede cambiar de forma instantánea sin requerir potencia infinita, la salida de un sistema dinámico evoluciona gradualmente siguiendo una ecuación diferencial ordinaria de orden ene.
- **Turno 4 (Femenina):** El orden ene de la ecuación diferencial, determinado por la derivada más alta de la variable de salida, coincide siempre con el número de elementos almacenadores de energía independientes que posee la planta mecatrónica.

### Diapositiva 02: Terminología de Señales de Prueba Estándar en Ingeniería

**Impulso Unitario $\delta(t)$, Escalón Unitario $u_s(t)$, Rampa $r(t) = t$ y Parábola $p(t) = \frac{1}{2}t^2$**

- **Turno 1 (Masculina):** Para evaluar y comparar objetivamente distintos sistemas mecatrónicos en el laboratorio, la ingeniería de control utiliza un conjunto universal de señales matemáticas de prueba en la entrada.
- **Turno 2 (Femenina):** La primera es la entrada escalón: un cambio instantáneo de un nivel constante a otro en el instante te igual a cero, que pone a prueba la rapidez de reacción del sistema ante un nuevo comando de consigna o setpoint.
- **Turno 3 (Masculina):** La segunda es la función impulso de Dirac: un pulso de duración infinitesimal pero de área unitaria, equivalente a golpear una estructura mecánica con un martillo de impacto para observar su vibración natural libre.
- **Turno 4 (Femenina):** La tercera es la entrada rampa, que crece linealmente con el tiempo a velocidad constante, como una antena que rastrea un satélite; y la cuarta es la entrada parabólica, que evalúa la capacidad de seguir un objetivo con aceleración constante.

### Diapositiva 03: Respuesta Transitoria frente a Respuesta en Estado Estacionario

**Descomposición $y(t) = y_{tr}(t) + y_{ss}(t)$, Estabilidad y Error de Régimen Permanente**

- **Turno 1 (Masculina):** Cuando aplicamos cualquiera de estas señales de prueba a un sistema dinámico, la respuesta temporal total ye de te se compone siempre de la suma de dos partes con significado físico muy distinto.
- **Turno 2 (Femenina):** La primera parte es la respuesta transitoria ye sub te erre: es la etapa de adaptación que va desde el estado inicial hasta las cercanías del estado final, gobernada por las raíces características o modos propios de la ecuación homogénea.
- **Turno 3 (Masculina):** Si el sistema es estable, todos los términos exponenciales de la respuesta transitoria decaen hasta extinguirse por completo cuando el tiempo tiende a infinito; si en cambio crecen sin límite, el sistema es inestable.
- **Turno 4 (Femenina):** La segunda parte es la respuesta en estado estacionario o régimen permanente ye sub ese ese: es el comportamiento que permanece una vez desaparecido el transitorio, y su diferencia respecto al valor deseado define el error de estado estacionario.

### Diapositiva 04: Sistemas de Primer Orden: Forma Estándar Canónica (τ y K)

**Normalización de la EDO τ·(dy/dt) + y = K·u(t), Ganancia Estática y Constante de Tiempo**

- **Turno 1 (Masculina):** Estudiemos a fondo los sistemas de primer orden, aquellos que contienen un único almacén de energía, como un circuito erre ce, la bobina erre ele de un electroimán, el nivel de un tanque o la temperatura de un horno.
- **Turno 2 (Femenina):** Aunque provengan de dominios físicos completamente distintos, al dividir toda la ecuación diferencial entre el coeficiente que acompaña a la variable de salida ye, todos ellos adoptan una única forma estándar canónica.
- **Turno 3 (Masculina):** Esa forma canónica es: tau multiplicado por la derivada de ye respecto al tiempo, más ye de te, es igual a ka multiplicado por la entrada u de te, quedando el sistema caracterizado por tan solo dos parámetros.
- **Turno 4 (Femenina):** El parámetro ka es la ganancia estática o sensibilidad de estado estacionario, que nos dice cuánto cambia la salida final por cada unidad de entrada cuando la derivada se hace cero; y el parámetro tau, medido en segundos, es la constante de tiempo del sistema.

### Diapositiva 05: Respuesta de un Sistema de Primer Orden a una Entrada Escalón

**Deducción de la Ley Exponencial Universal y(t) = K·A·(1 - e^{-t/τ})**

- **Turno 1 (Masculina):** Resolvamos la ecuación diferencial de primer orden cuando en el instante te igual a cero aplicamos un escalón de amplitud constante A, partiendo del reposo con condición inicial ye de cero igual a cero.
- **Turno 2 (Femenina):** La solución particular de estado estacionario, cuando la derivada de ye se anula, es simplemente el valor final constante ye sub ese ese igual al producto de la ganancia ka por la amplitud del escalón A.
- **Turno 3 (Masculina):** La solución homogénea o transitoria proviene de la ecuación característica tau por ese más uno igual a cero, cuya raíz es ese igual a menos uno sobre tau, generando un modo exponencial decreciente elevado a menos te sobre tau.
- **Turno 4 (Femenina):** Imponiendo que en te igual a cero la salida vale cero, obtenemos la célebre respuesta al escalón de todo sistema de primer orden: ye de te es igual a ka por A multiplicado por uno menos e elevado a la menos te sobre tau.

### Diapositiva 06: Interpretación Física y Geométrica de la Constante de Tiempo (τ)

**Regla del 63.2% en t = 1τ, Pendiente Inicial en el Origen y Asentamiento en 4τ y 5τ**

- **Turno 1 (Masculina):** ¿Qué representa exactamente la constante de tiempo tau en la curva de respuesta al escalón y cómo podemos medirla a simple vista sobre la pantalla de un osciloscopio o en un registro de datos?
- **Turno 2 (Femenina):** Primera propiedad clave: si evaluamos la fórmula en el instante te exactamente igual a una constante de tiempo tau, el término uno menos e a la menos uno vale cero punto seis tres dos; es decir, en te igual a tau la salida alcanza el sesenta y tres coma dos por ciento de su valor final.
- **Turno 3 (Masculina):** Segunda propiedad geométrica: la velocidad inicial de subida en te igual a cero es máxima y vale ka por A sobre tau; si trazamos la recta tangente a la curva en el origen, esa recta corta la línea horizontal del valor final exactamente en te igual a tau.
- **Turno 4 (Femenina):** Tercera propiedad práctica: en dos constantes de tiempo la respuesta llega al ochenta y seis coma cinco por ciento, en tres tau al noventa y cinco por ciento, en cuatro tau al noventa y ocho coma dos por ciento, y en cinco tau al noventa y nueve coma tres por ciento, considerándose totalmente establecida.

### Diapositiva 07: Especificaciones de Desempeño de Primer Orden: t_d, t_r y t_s

Tiempo de Retardo 0.69τ, Tiempo de Subida (10% a 90%) t_r = 2.2τ y Establecimiento t_s = 4τ

- **Turno 1 (Masculina):** A partir de la exponencial uno menos e a la menos te sobre tau, podemos deducir analíticamente las tres medidas estándar de velocidad de respuesta para cualquier sistema de primer orden.
- **Turno 2 (Femenina):** El tiempo de retardo te sub de es el tiempo necesario para alcanzar por primera vez el cincuenta por ciento del valor final: despejando e a la menos te sobre tau igual a cero punto cinco, obtenemos logaritmo natural de dos por tau, que es cero punto sesenta y nueve veces tau.
- **Turno 3 (Masculina):** El tiempo de subida te sub erre se define en sistemas sin sobreimpulso como el tiempo que tarda la respuesta en pasar del diez por ciento al noventa por ciento de su valor final: restando te noventa menos te diez obtenemos logaritmo natural de nueve por tau, igual a dos punto dos veces tau.
- **Turno 4 (Femenina):** Y el tiempo de establecimiento te sub ese es el tiempo requerido para que la curva entre y permanezca dentro de una banda de tolerancia alrededor del valor final: vale tres veces tau para el criterio del cinco por ciento, y cuatro veces tau para el criterio industrial estándar del dos por ciento.

### Diapositiva 08: Respuesta de Primer Orden ante Impulso y ante Entrada Rampa

Decaimiento Exponencial al Impulso y Retraso de Seguimiento Estacionario e_ss = v_0·τ en Rampa

- **Turno 1 (Masculina):** Veamos ahora cómo responde el mismo sistema de primer orden cuando en lugar de un escalón le aplicamos un impulso unitario o una entrada en rampa de pendiente constante ve sub cero por te.
- **Turno 2 (Femenina):** Dado que el impulso es la derivada temporal del escalón, por linealidad la respuesta al impulso es la derivada de la respuesta al escalón: salta instantáneamente en te igual a cero hasta el valor ka sobre tau, y desde allí decae exponencialmente hacia cero.
- **Turno 3 (Masculina):** Por el contrario, cuando aplicamos una rampa lineal u de te igual a ve sub cero por te con ganancia unitaria, la solución es ve sub cero multiplicado por te menos tau por uno menos e a la menos te sobre tau.
- **Turno 4 (Femenina):** Cuando transcurren más de cuatro constantes de tiempo y el término exponencial desaparece, la salida sigue a la rampa con exactamente la misma pendiente, pero atrasada en el tiempo por una cantidad constante igual a tau, produciendo un error de seguimiento estacionario igual a la velocidad por tau.

### Diapositiva 09: Constantes de Tiempo Físicas en los Cuatro Dominios Mecatrónicos

Síntesis de τ = R·C, τ = L/R, τ = m/c, τ = J/b, τ = R_h·C_h y τ = R_T·C_T

- **Turno 1 (Masculina):** Comprender cómo depende la constante de tiempo tau de los parámetros físicos nos permite rediseñar cualquier componente mecatrónico para hacerlo más rápido o para filtrar mejor el ruido.
- **Turno 2 (Femenina):** En un filtro eléctrico erre ce, tau es el producto resistencia por capacitancia; mientras que en el devanado inductivo erre ele de un solenoide o motor, tau es el cociente inductancia sobre resistencia: añadir resistencia en serie acelera la subida de la corriente.
- **Turno 3 (Masculina):** En un carro mecánico de masa eme con fricción viscosa ce, tau vale masa sobre amortiguamiento; y en el eje de un motor rotativo de inercia jota y fricción efectiva be, la constante mecánica vale inercia jota dividida entre be.
- **Turno 4 (Femenina):** Finalmente, en un tanque hidráulico tau es resistencia de la válvula por capacitancia del área del tanque, y en un sensor térmico tau es masa por calor específico dividido entre el coeficiente de convección hache por el área superficial.

**Diapositiva 10: Estudio de Caso Práctico: Dinámica de un Sensor RTD y de un Motor CC**

Cálculo de Tiempos de Respuesta, Error Dinámico en Rampa y Criterio de Muestreo Digital

- **Turno 1 (Masculina):** Resolvamos un caso práctico de ingeniería: un sensor de temperatura erre te de con constante de tiempo tau igual a dos segundos se sumerge súbitamente desde veinticinco grados hasta un baño térmico a ciento veinticinco grados Celsius.
- **Turno 2 (Femenina):** El salto total es de cien grados. Al cabo de dos segundos, una constante de tiempo, el sensor marcará veinticinco más sesenta y tres coma dos, es decir, ochenta y ocho coma dos grados; y recién a los ocho segundos, cuatro veces tau, alcanzará ciento veintitrés grados, dentro del dos por ciento.
- **Turno 3 (Masculina):** Si en cambio la temperatura del baño sube en rampa a razón de tres grados por segundo, el sensor irá siempre dos segundos atrasado, leyendo en todo momento seis grados menos que la temperatura real del líquido.
- **Turno 4 (Femenina):** Además, para digitalizar o controlar en lazo cerrado cualquier planta de primer orden con un microcontrolador, la regla práctica de ingeniería exige elegir un periodo de muestreo te sub ese al menos diez a veinte veces menor que la constante de tiempo tau.

## 4. Firmware C++/Arduino: Integrador RK4 y Filtro de Tustin (simulacion_hil_y_adquisicion_video_05.ino)

```
/*
 * ============================================================================
 * CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
 * VIDEO 05: Video 05 — Parte 3A: Respuestas Dinámicas: Terminología y Sistemas de Primer Orden
 * Archivo: simulacion_hil_y_adquisicion_video_05.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 "Planta Canónica de Primer Orden (Térmica / RC / Velocidad de Motor)",
 *   filtro discreto por Transformada Bilineal de Tustin y adquisición por ADC
 *   para identificación experimental de parámetros (K, tau, wn, zeta).
 *
 * Ecuación Modelo:
 *   τ·(dy/dt) + y = K·u(t)  ==>  y(t) = K·A·(1 - e^{-t/τ}),   t_r = 2.2·τ,   t_s(2%) = 4·τ
 * ============================================================================
 */

#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 05) ---
const float M_EQ       = 0.5000f;   // Masa / Inercia equivalente
const float C_EQ       = 1.0000f;   // Amortiguamiento / Resistencia equivalente
const float K_EQ       = 2.0000f;   // Rigidez / Elasticidad equivalente
const float WN_RAD_S   = 2.000f;  // Frecuencia natural no amortiguada [rad/s]
const float ZETA_REL   = 1.000f; // Factor de amortiguamiento relativo [-]
const float K_GAIN     = 2.000f;   // Ganancia estática de la planta
const float TAU_SEC    = 0.500f; // 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 05"));
  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_05.py)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
================================================================================
CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
VIDEO 05: Video 05 — Parte 3A: Respuestas Dinámicas: Terminología y Sistemas de Primer Orden
Archivo: modelado_laplace_y_bode_video_05.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 05: Planta Canónica de Primer Orden (Térmica / RC / Velocidad de Motor)
M_EQ     = 0.5
C_EQ     = 1
K_EQ     = 2
WN_RAD   = 2
ZETA     = 1
K_STATIC = 2
TAU_SEC  = 0.5


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 05")
    print(" Planta:", "Planta Canónica de Primer Orden (Térmica / RC / Velocidad de Motor)")
    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_05.m)

```
% ========================================================================
% CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
% VIDEO 05: Video 05 — Parte 3A: Respuestas Dinámicas: Terminología y Sistemas de Primer Orden
% Archivo: analisis_control_mecatronico_video_05.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 05\n');
fprintf(' Planta: Planta Canónica de Primer Orden (Térmica / RC / Velocidad de Motor)\n');
fprintf('========================================================================\n\n');

%% 1. Definicion de Parametros Fisicos y Canónicos
m_eq   = 0.5;
c_eq   = 1;
k_eq   = 2;
wn     = 2;
zeta   = 1;
K_gain = 2;
tau    = 0.5;

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 = [2.0];
den_G2 = [0.5, 1.0];
G2 = tf(num_G2, den_G2);
fprintf('--- Sistema Principal del Video 05: 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 05 - 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 05 - 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 05) ===\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);
```