CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA

Video 07 — Parte 4A: La Función de Transferencia en Laplace: Primer y Segundo Orden — Transformada de Laplace, Propiedad de Derivación, Definición de G(s), Impedancias Operacionales y Fracciones Parciales

Parte del Curso: Parte 4A — La Función de Transferencia en Laplace: Primer y Segundo Orden
Planta / Sistema Modelado: Modelado en el Dominio de Laplace G(s) = Y(s)/U(s) y Expansión en Fracciones Parciales

```
G_1(s) = K / (τ·s + 1) | G_2(s) = K·ω_n² / (s² + 2·ζ·ω_n·s + ω_n²) | y_ss = lim_{s->0} s·Y(s)
```

1. Síntesis de Conceptos Clave y Fórmulas de Ingeniería

- Transformada de Laplace y propiedad operacional: con condiciones iniciales nulas, d^n/dt^n se transforma en multiplicar por s^n y la integral en dividir por s.
- Teorema del Valor Inicial y(0+) = lim_{s->∞} s·Y(s) y Teorema del Valor Final y_ss = lim_{s->0} s·Y(s) (para polos en el semiplano izquierdo).
- Impedancias operacionales de Laplace (Z_R = R, Z_L = L·s, Z_C = 1/(C·s)) y funciones de transferencia de velocidad y posición en motores de CC.
- Transformada inversa mediante descomposición en fracciones parciales de Heaviside para polos reales simples, polos repetidos y pares complejos conjugados.

2. Parámetros Nominales del Modelo Matemático (Video 07)

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`	7.2	Elemento disipador de energía que gobierna la tasa de atenuación
Rigidez / Elasticidad Equivalente	`k_eq / 1/C`	36	Almacenador de energía potencial restauradora
Frecuencia Natural No Amortiguada	`ω_n [rad/s]`	6 rad/s	Distancia radial de los polos al origen en el plano complejo s
Factor de Amortiguamiento Relativo	`ζ [-]`	0.6	Determina el régimen dinámico y el sobreimpulso máximo M_p
Ganancia Estática / Constante de Tiempo	`K / τ [s]`	K = 1, τ = 0.278 s	Sensibilidad estacionaria y rapidez del modo exponencial e^{-t/τ}

3. Desarrollo Académico Diapositiva por Diapositiva

Diapositiva 01: La Transformada de Laplace en Ingeniería Mecatrónica

Definición F(s) = ∫ f(t)·e^{-st} dt y Conversión de Ecuaciones Diferenciales en Algebraicas

- **Turno 1 (Masculina):** Bienvenidos a la Parte cuatro del curso, donde introducimos la herramienta matemática más poderosa de toda la ingeniería de control y el modelado mecatrónico: la función de transferencia en el dominio de Laplace.
- **Turno 2 (Femenina):** Resolver ecuaciones diferenciales acopladas de tercer o cuarto orden directamente en el dominio del tiempo resulta largo y tedioso; por ello el matemático Pierre-Simon Laplace introdujo una transformación integral que cambia las reglas del juego.
- **Turno 3 (Masculina):** La transformada unilateral de Laplace toma una señal temporal efe de te definida desde te igual a cero hasta infinito, la multiplica por el núcleo exponencial e elevado a menos ese por te, e integra respecto al tiempo, convirtiéndola en una función efe mayúscula de la variable compleja ese.
- **Turno 4 (Femenina):** Aquí la variable de Laplace ese es un número complejo compuesto por una parte real sigma, asociada a la atenuación o crecimiento exponencial, más una parte imaginaria jota omega, asociada a la frecuencia de oscilación.

Diapositiva 02: Transformadas Elementales y Teoremas de Derivación e Integración

Correspondencia d/dt ↔ multiplicar por s, d^2/dt^2 ↔ s^2 e Integral ↔ dividir por s

- **Turno 1 (Masculina):** El motivo por el cual la transformada de Laplace simplifica de manera espectacular el modelado de sistemas radica en su propiedad de derivación e integración en el tiempo.
- **Turno 2 (Femenina):** Cuando las condiciones iniciales del sistema en te igual a cero son nulas, derivar una función respecto al tiempo equivale en el dominio de Laplace simplemente a multiplicar su transformada por la variable algebraica ese.
- **Turno 3 (Masculina):** De igual modo, la segunda derivada respecto al tiempo se convierte en multiplicar por ese al cuadrado; la derivada enésima en multiplicar por ese a la ene; y la integral respecto al tiempo se transforma en dividir entre ese.
- **Turno 4 (Femenina):** Además, las señales de prueba tienen transformadas elementales directas: el impulso unitario vale uno, el escalón unitario vale uno sobre ese, la rampa vale uno sobre ese al cuadrado, y la exponencial e a la menos a por te vale uno sobre ese más a.

Diapositiva 03: Teoremas del Valor Inicial y del Valor Final en el Dominio de Laplace

Cálculo Directo del Estado Estacionario y_ss = lim_{s→0} [s · Y(s)] sin Antitransformar

- **Turno 1 (Masculina):** Existen dos teoremas en el dominio de Laplace que nos ahorran tener que calcular la transformada inversa completa cuando solo queremos conocer cómo arranca o dónde termina la respuesta temporal.
- **Turno 2 (Femenina):** El primero es el teorema del valor inicial: el valor de la señal ye de te justo después del instante cero es igual al límite cuando la variable ese tiende a infinito del producto ese por ye mayúscula de ese.
- **Turno 3 (Masculina):** El segundo, aún más utilizado en diseño de controladores, es el teorema del valor final: si todos los polos de ese por ye de ese tienen parte real estrictamente negativa, el valor en estado estacionario cuando el tiempo tiende a infinito es igual al límite cuando ese tiende a cero de ese por ye de ese.
- **Turno 4 (Femenina):** Por ejemplo, si aplicamos un escalón unitario U de ese igual a uno sobre ese a un sistema estable con función de transferencia ge de ese, al multiplicar por ese se cancela el uno sobre ese y la salida estacionaria es simplemente ge evaluada en ese igual a cero.

Diapositiva 04: Definición Formal de la Función de Transferencia G(s) = Y(s) / U(s)

Condiciones Iniciales Nulas, Polinomio Numerador (Ceros) y Denominador Característico (Polos)

- **Turno 1 (Masculina):** Definamos formalmente la función de transferencia ge de ese de un sistema lineal e invariante en el tiempo: es el cociente entre la transformada de Laplace de la señal de salida ye de ese y la transformada de Laplace de la señal de entrada u de ese, asumiendo todas las condiciones iniciales iguales a cero.
- **Turno 2 (Femenina):** Al aplicar Laplace a la ecuación diferencial lineal del sistema y factorizar ye de ese y u de ese, la función de transferencia queda expresada como el cociente de dos polinomios en ese: un numerador ene de ese dividido entre un denominador de de ese.
- **Turno 3 (Masculina):** Las raíces del polinomio del numerador, aquellos valores de ese que anulan la transmisión, se denominan ceros de la función de transferencia; mientras que las raíces del polinomio del denominador se denominan polos del sistema.
- **Turno 4 (Femenina):** Observemos una propiedad fundamental: la función de transferencia ge de ese es una propiedad intrínseca de la planta física que depende exclusivamente de sus parámetros constructivos y es totalmente independiente de qué señal de entrada le apliquemos.

Diapositiva 05: Funciones de Transferencia de Sistemas de Primer Orden

Forma Canónica G(s) = K / (τ·s + 1) en Sistemas Mecánicos, Eléctricos, Hidráulicos y Térmicos

- **Turno 1 (Masculina):** Obtengamos la función de transferencia general de cualquier sistema de primer orden partiendo de su ecuación diferencial canónica: tau por la derivada de ye respecto al tiempo, más ye de te, es igual a ka por u de te.
- **Turno 2 (Femenina):** Aplicando la transformada de Laplace con condición inicial nula, la derivada se convierte en ese por ye de ese, resultando: tau por ese por ye de ese, más ye de ese, igual a ka por u de ese.
- **Turno 3 (Masculina):** Sacando factor común ye de ese que multiplica a tau por ese más uno, y pasando u de ese dividiendo, obtenemos la función de transferencia canónica de primer orden: ge de ese es igual a la ganancia ka dividida entre tau por ese más uno.
- **Turno 4 (Femenina):** Así, para un filtro erre ce la función de transferencia es uno sobre erre por ce por ese más uno; para un tanque hidráulico es erre sub hache sobre ro ge dividido entre erre sub hache por ce sub hache por ese más uno; y todos ellos poseen un único polo real ubicado en ese igual a menos uno sobre tau.

Diapositiva 06: Funciones de Transferencia de Sistemas de Segundo Orden e Impedancias Z(s)

Forma Canónica G(s) = K·ω_n² / (s² + 2·ζ·ω_n·s + ω_n²) y Método de Impedancias de Laplace

- **Turno 1 (Masculina):** Apliquemos el mismo procedimiento a la ecuación diferencial de segundo orden: segunda derivada de ye, más dos por dseta por omega sub ene por la primera derivada, más omega sub ene al cuadrado por ye, igual a ka por omega sub ene al cuadrado por la entrada.
- **Turno 2 (Femenina):** Al transformar a Laplace reemplazando la segunda derivada por ese al cuadrado y la primera por ese, despejamos el cociente ye de ese sobre u de ese y obtenemos: ka por omega sub ene al cuadrado, dividido entre ese al cuadrado más dos dseta omega sub ene por ese más omega sub ene al cuadrado.
- **Turno 3 (Masculina):** Para el sistema masa-resorte-amortiguador, la función de transferencia entre el desplazamiento equis de ese y la fuerza efe de ese es directamente uno dividido entre masa por ese al cuadrado, más amortiguamiento ce por ese, más rigidez ka.
- **Turno 4 (Femenina):** Y en circuitos eléctricos ni siquiera necesitamos plantear la ecuación diferencial: definiendo las impedancias operacionales de Laplace, erre para el resistor, ele por ese para el inductor y uno sobre ce por ese para el capacitor, aplicamos un simple divisor de tensión algebraico.

Diapositiva 07: Funciones de Transferencia de un Motor de Corriente Continua (CC)

Transferencia a Velocidad Ω(s)/V_a(s) (Tipo 0) y a Posición Angular Θ(s)/V_a(s) (Tipo 1 con Polo en s = 0)

- **Turno 1 (Masculina):** Deduzcamos ahora las dos funciones de transferencia más utilizadas en robótica y servocontrol: las del motor de corriente continua controlado por tensión de armadura ve sub a de ese.
- **Turno 2 (Femenina):** En Laplace, la ecuación eléctrica es ve sub a de ese menos ka sub e por omega de ese igual a ele sub a por ese más erre sub a multiplicado por la corriente i sub a de ese; y la ecuación mecánica sin par de carga es jota por ese más be multiplicado por omega de ese igual a ka sub te por la corriente.
- **Turno 3 (Masculina):** Eliminando la corriente obtenemos la función de transferencia desde el voltaje de armadura hasta la velocidad angular omega de ese: es igual a ka sub te dividido entre el producto de ele sub a ese más erre sub a por jota ese más be, más el producto ka sub te por ka sub e.
- **Turno 4 (Femenina):** Y si la salida que nos interesa controlar es la posición angular teta de ese del eje del motor, como la velocidad es la derivada de la posición, basta dividir la función anterior entre ese: aparece así un polo integrador puro en el origen ese igual a cero.

Diapositiva 08: Transformada Inversa de Laplace mediante Fracciones Parciales (Polos Reales)

Descomposición Y(s) = Σ r_k / (s - p_k) y Cálculo de Residuos de Heaviside

- **Turno 1 (Masculina):** Una vez que multiplicamos la función de transferencia ge de ese por la transformada de la entrada u de ese para obtener ye de ese, ¿cómo regresamos al dominio del tiempo para hallar la curva exacta ye de te?
- **Turno 2 (Femenina):** El método universal es la descomposición en fracciones parciales de Heaviside, que consiste en factorizar el denominador de ye de ese en sus polos simples ese menos pe sub uno, ese menos pe sub dos, hasta ese menos pe sub ene.
- **Turno 3 (Masculina):** Expresamos entonces ye de ese como una suma de fracciones simples: un residuo constante erre uno dividido entre ese menos pe uno, más erre dos dividido entre ese menos pe dos, y así sucesivamente para cada polo.
- **Turno 4 (Femenina):** Cada coeficiente o residuo erre sub ka se calcula tapando su propio factor ese menos pe sub ka y evaluando el resto de la expresión en ese igual a pe sub ka; al antitransformar término a término, cada fracción uno sobre ese menos pe sub ka se convierte directamente en la exponencial e elevado a pe sub ka por te.

Diapositiva 09: Fracciones Parciales con Polos Repetidos y Polos Complejos Conjugados

Completación de Cuadrados $(s + \sigma)^2 + \omega_d^2$ y Conversión a Cosenos y Senos Amortiguados

- **Turno 1 (Masculina):** ¿Cómo procedemos cuando el denominador de ye de ese posee un polo real repetido, como ocurre en el amortiguamiento crítico, o un par de polos complejos conjugados, como ocurre en un sistema subamortiguado?
- **Turno 2 (Femenina):** Si existe un polo real doble en ese igual a menos a, debemos incluir dos fracciones en la expansión: una constante ce uno sobre ese más a, y una segunda constante ce dos sobre ese más a elevado al cuadrado, cuya antitransformada es ce dos por te por e a la menos a por te.
- **Turno 3 (Masculina):** Cuando tenemos un factor cuadrático con raíces complejas conjugadas menos sigma más o menos jota omega sub de, en lugar de separar en números complejos completamos el cuadrado perfecto en el denominador: ese más sigma al cuadrado, más omega sub de al cuadrado.
- **Turno 4 (Femenina):** Separando el numerador en una parte proporcional a ese más sigma y otra proporcional a omega sub de, la transformada inversa se lee de forma inmediata como e a la menos sigma por te que multiplica a una combinación de coseno de omega sub de por te más seno de omega sub de por te.

Diapositiva 10: Sistemas con Múltiples Entradas y Perturbaciones (Superposición en Laplace)

Función de Transferencia de Mando G_1(s) y Función de Sensibilidad al Par de Carga G_2(s)

- **Turno 1 (Masculina):** En una máquina mecatrónica real casi nunca actúa una sola entrada: sobre el eje de un servomotor actúan al mismo tiempo el voltaje de control ve sub a de ese y el par perturbador de la carga mecánica te sub ele de ese.
- **Turno 2 (Femenina):** Gracias al principio de superposición lineal en el dominio de Laplace, podemos analizar el efecto de cada entrada por separado anulando temporalmente las demás, y luego sumar algebraicamente las salidas.
- **Turno 3 (Masculina):** Así, la velocidad del motor omega de ese es igual a una primera función de transferencia ge uno de ese multiplicada por el voltaje de armadura, más una segunda función de transferencia ge dos de ese multiplicada por el par de perturbación.
- **Turno 4 (Femenina):** En sistemas multivariables con múltiples entradas y múltiples salidas, conocidos como sistemas MIMO, esta colección ordenada de funciones de transferencia entre cada par de entrada y salida conforma la matriz de transferencia completa de la planta.

4. Firmware C++/Arduino: Integrador RK4 y Filtro de Tustin (simulacion_hil_y_adquisicion_video_07.ino)

```
/*
 * ============================================================================
 * CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
 * VIDEO 07: Video 07 — Parte 4A: La Función de Transferencia en Laplace: Primer y Segundo Orden
 * Archivo: simulacion_hil_y_adquisicion_video_07.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 "Modelado en el Dominio de Laplace G(s) = Y(s)/U(s) y Expansión en Fracciones Parciales",
 *   filtro discreto por Transformada Bilineal de Tustin y adquisición por ADC
 *   para identificación experimental de parámetros (K, tau, wn, zeta).
 *
 * Ecuación Modelo:
 *   G_1(s) = K / (τ·s + 1)   |   G_2(s) = K·ω_n² / (s² + 2·ζ·ω_n·s + ω_n²)   |   y_ss = lim_{s->0} s·Y(s)
 * ============================================================================
 */

#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 07) ---
const float M_EQ       = 1.0000f;   // Masa / Inercia equivalente
const float C_EQ       = 7.2000f;   // Amortiguamiento / Resistencia equivalente
const float K_EQ       = 36.0000f;   // Rigidez / Elasticidad equivalente
const float WN_RAD_S   = 6.000f;  // Frecuencia natural no amortiguada [rad/s]
const float ZETA_REL   = 0.600f; // Factor de amortiguamiento relativo [-]
const float K_GAIN     = 1.000f;   // Ganancia estática de la planta
const float TAU_SEC    = 0.278f; // 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 07"));
  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_07.py)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
================================================================================
CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
VIDEO 07: Video 07 — Parte 4A: La Función de Transferencia en Laplace: Primer y Segundo Orden
Archivo: modelado_laplace_y_bode_video_07.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 07: Modelado en el Dominio de Laplace G(s) = Y(s)/U(s) y Expansión en Fracciones
Parciales
M_EQ     = 1
C_EQ     = 7.2
K_EQ     = 36
WN_RAD   = 6
ZETA     = 0.6
K_STATIC = 1
TAU_SEC  = 0.278


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 07")
    print(" Planta:", "Modelado en el Dominio de Laplace G(s) = Y(s)/U(s) y Expansión en Fracciones Parciales")
    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_07.m)

```
% ========================================================================
% CURSO COMPLETO: MODELADO DE SISTEMAS EN MECATRÓNICA
% VIDEO 07: Video 07 – Parte 4A: La Función de Transferencia en Laplace: Primer y Segundo Orden
% Archivo: analisis_control_mecatronico_video_07.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 07\n');
fprintf(' Planta: Modelado en el Dominio de Laplace G(s) = Y(s)/U(s) y Expansión en Fracciones Parciales\n');
fprintf('========================================================================\n\n');

%% 1. Definicion de Parametros Fisicos y Canónicos
m_eq   = 1;
c_eq   = 7.2;
k_eq   = 36;
wn     = 6;
zeta   = 0.6;
K_gain = 1;
tau    = 0.278;

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 = [36.0];
den_G2 = [1.0, 7.2, 36.0];
G2 = tf(num_G2, den_G2);
fprintf('--- Sistema Principal del Video 07: 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 07 - 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 07 - 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 07) ===\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);
```