CURSO COMPLETO: FUNDAMENTOS DE ELECTRÓNICA

Video 02 — Parte 1B: Análisis en Corriente Alterna, Potencia, Transformador, Adaptación de Impedancias y Consideraciones Prácticas — Fasores e Impedancia Compleja, Potencia en CA, Transformadores, Máxima Transferencia, Capacitores, Protoboard, Soldadura, Osciloscopio, Tierra y Seguridad

Parte del Curso: Parte 1B — Análisis en Corriente Alterna, Potencia, Transformador, Adaptación de Impedancias y Práctica de Laboratorio
Arquitectura / Controlador Estudiado: Fasores, Impedancia Compleja RLC, Potencia Activa/Reactiva/Aparente, Transformador Ideal, Adaptación Z_L = Z_th* e Instrumentación

$$Z = R + j(\omega L - 1/(\omega C)) \mid S = P + jQ = V_{rms} \cdot I_{rms}^* \mid V_1/V_2 = N_1/N_2 = a \mid Z_{in} = a^2 \cdot Z_L \mid Z_L = Z_{th}^*$$

1. Síntesis de Conceptos Clave y Reglas de Ingeniería

- Análisis fasorial en régimen senoidal permanente: reactancia inductiva $X_L = \omega L$ (+90°), reactancia capacitiva $X_C = -1/(\omega C)$ (-90°) e impedancia compleja $Z = |Z| \angle \theta$.
- Triángulo de potencia en CA (potencia activa $P = V \cdot I \cdot \cos\theta$ en W, reactiva $Q = V \cdot I \cdot \text{sen}\theta$ en VAR, aparente $S = V \cdot I$ en VA y factor de potencia FP = $\cos\theta$), transformadores (escalado de tensión, corriente e impedancia reflejada $a^2 \cdot Z_L$) y máxima transferencia de potencia ($Z_L = Z_{th}^*$).
- Consideraciones prácticas de componentes y montaje: selección de capacitores (cerámicos de desacoplo, electrolíticos polarizados, film), capacitancias parásitas en protoboard, medición sin efecto de carga y soldadura electrónica libre de uniones frías.
- Uso profesional del osciloscopio (sondas compensadas 10x, disparo Trigger y acoplamiento DC/AC), prevención de lazos de tierra (ground loops), blindaje contra EMI y normas de seguridad eléctrica (umbrales de fibrilación ventricular e interruptor diferencial GFCI).

2. Parámetros Nominales de Configuración del Controlador (Video 02)

Parámetro del Controlador / Interfase	Símbolo	Valor Nominal	Función en el Lazo de Control / Automatización
Ganancia Proporcional	`Kp (PB = 100/Kp)`	3.5	Esfuerzo inmediato proporcional al error actual e(t)
Tiempo Integral	`Ti [s]`	0.45 s	Elimina el error estacionario (Offset) acumulando la integral del error
Tiempo Derivativo y Filtro Divisor	`Td [s] / N`	Td = 0.08 s, N = 12	Anticipa la tendencia sobre -PV y atenúa el ruido de alta frecuencia
Periodo de Muestreo en Tiempo Real	`Ts [ms]`	1 ms	Intervalo determinista de conversión ADC, cálculo discreto y actualización PWM
Límites de Saturación Anti-Windup	`[u_min, u_max]`	[-15, 15]	Cota física del actuador para integración condicional (Clamping)
Rango Escalado de Ingeniería (SCALE_X)	`[PV_min, PV_max]`	[0, 220]	Conversión lineal desde cuentas crudas del ADC (0..1023 o 0..27648)

3. Desarrollo Académico Diapositiva por Diapositiva

Diapositiva 01: Análisis de Circuitos en Corriente Alterna (CA) y Fasores

Ondas Senoidales, Frecuencia Angular ω = 2πf, Valor Eficaz (RMS), Representación Fasorial e Impedancia Z

- **Turno 1 (Masculina):** En el análisis de circuitos de corriente alterna excitamos la red con tensiones y corrientes senoidales caracterizadas por su amplitud pico, su frecuencia angular omega igual a dos pi por efe y su ángulo de fase inicial.
- **Turno 2 (Femenina):** Para cuantificar la capacidad energética real de una onda alterna utilizamos el valor eficaz o erre eme ese, que para una señal senoidal pura equivale a la amplitud pico dividida entre la raíz cuadrada de dos.
- **Turno 3 (Masculina):** Mediante la identidad de Euler transformamos ecuaciones diferenciales en el tiempo en ecuaciones algebraicas en el plano complejo utilizando fasores, los cuales capturan el módulo y el ángulo de fase de cada señal.
- **Turno 4 (Femenina):** En el dominio fasorial definimos la impedancia compleja zeta como el cociente entre el fasor de tensión y el fasor de corriente, cuya parte real es la resistencia erre y cuya parte imaginaria es la reactancia equis.

Diapositiva 02: Reactancia Inductiva, Reactancia Capacitiva y Redes RLC en CA

Impedancia del Inductor Z_L = jωL, Impedancia del Capacitor Z_C = -j/(ωC) y Desfase Tensión-Corriente

- **Turno 1 (Masculina):** Al aplicar una corriente senoidal a un inductor ideal, la derivada temporal multiplica por jota omega, produciendo una impedancia inductiva jota omega ele cuya magnitud crece linealmente con la frecuencia.
- **Turno 2 (Femenina):** El factor imaginario positivo jota significa que en un inductor puro la tensión adelanta exactamente noventa grados a la corriente eléctrica.
- **Turno 3 (Masculina):** En un capacitor ideal, la impedancia es uno sobre jota omega ce, o equivalentemente menos jota dividido entre omega ce, de modo que su reactancia disminuye al aumentar la frecuencia y la corriente adelanta noventa grados a la tensión.
- **Turno 4 (Femenina):** Gracias al concepto de impedancia compleja, todas las reglas de serie, paralelo, divisores, Thévenin y Norton estudiadas en corriente continua se aplican de forma idéntica en corriente alterna operando con números complejos.

Diapositiva 03: Potencia en Circuitos Eléctricos y Factor de Potencia

Potencia Instantánea p(t), Potencia Activa Media P [W], Reactiva Q [VAR], Aparente S [VA] y Corrección de Factor de Potencia

- **Turno 1 (Masculina):** En un circuito de corriente alterna cuya tensión y corriente presentan un ángulo de desfase teta, la potencia instantánea oscila al doble de la frecuencia de red e intercambia energía entre la fuente y los campos reactivos.
- **Turno 2 (Femenina):** La potencia activa o real pe, medida en vatios, es el promedio temporal que realiza trabajo útil o calor y equivale al voltaje eficaz por la corriente eficaz multiplicado por el coseno del ángulo de desfase.
- **Turno 3 (Masculina):** A ese coseno de teta lo denominamos factor de potencia, mientras que el producto de tensión eficaz por corriente eficaz por el seno de teta define la potencia reactiva cu en voltiamperios reactivos, asociada al trasiego de energía en bobinas y capacitores.
- **Turno 4 (Femenina):** El módulo de la potencia compleja es la potencia aparente ese en voltiamperios; en instalaciones con motores inductivos se conectan bancos de capacitores en paralelo para compensar la potencia reactiva y acercar el factor de potencia a la unidad.

Diapositiva 04: El Transformador Eléctrico: Acoplamiento Magnético y Relaciones

Inductancia Mutua, Relación de Transformación a = N_1/N_2, Escalado de Tensión y Corriente e Impedancia Reflejada

- **Turno 1 (Masculina):** El transformador es un dispositivo electromagnético pasivo formado por dos o más devanados acoplados a través de un núcleo ferromagnético común de alta permeabilidad, operando exclusivamente con flujos variables en el tiempo.
- **Turno 2 (Femenina):** En un transformador ideal, la relación entre el voltaje del primario ve uno y el voltaje del secundario ve dos es exactamente igual al cociente entre el número de espiras ene uno y ene dos.
- **Turno 3 (Masculina):** Por conservación de potencia instantánea, las corrientes se escalan en proporción inversa al número de espiras: elevar la tensión en el secundario reduce proporcionalmente la corriente disponible.
- **Turno 4 (Femenina):** Además de elevar o reducir tensiones y proveer aislamiento galvánico total entre circuitos, un transformador refleja una impedancia de carga zeta sub ele hacia el primario multiplicada por el cuadrado de la relación de espiras.

Diapositiva 05: Adaptación de Impedancias y Máxima Transferencia de Potencia

Condición Resistiva R_L = R_th, Condición Compleja Conjugada Z_L = Z_th* y Uso de Transformadores y Redes LC

- **Turno 1 (Masculina):** En sistemas de instrumentación, audio, transductores ultrasónicos y radiofrecuencia, un problema fundamental es lograr la adaptación de impedancias para transferir la máxima potencia posible desde una fuente hacia la carga.
- **Turno 2 (Femenina):** En circuitos puramente resistivos, derivando la potencia en la carga respecto a erre sub ele se demuestra que la transferencia de potencia alcanza su máximo absoluto cuando la resistencia de carga iguala exactamente a la resistencia de Thévenin de la fuente.
- **Turno 3 (Masculina):** Cuando la fuente y la carga poseen impedancias complejas en corriente alterna, la máxima transferencia de potencia exige que la impedancia de carga sea el complejo conjugado de la impedancia de Thévenin.
- **Turno 4 (Femenina):** Esto implica igualar las partes resistivas y hacer que la reactancia de la carga tenga igual magnitud pero signo opuesto a la reactancia de la fuente, cancelando la parte reactiva mediante redes LC o transformadores adaptadores.

Diapositiva 06: Consideraciones Prácticas I: Información de Capacitores y Prototipado en Protoboard

Capacitores Cerámicos, Electrolíticos, Tántalo y Film, Polaridad, ESR, Desacoplo y Reglas de Montaje en Protoboard

- **Turno 1 (Masculina):** Al llevar la teoría al banco de trabajo es esencial elegir el tipo físico de capacitor adecuado: los capacitores electrolíticos de aluminio y de tántalo ofrecen alta capacitancia para filtrado de fuentes, pero son polarizados y explotan si se invierte su polaridad.
- **Turno 2 (Femenina):** Por el contrario, los capacitores cerámicos multicapa y los de película plástica no tienen polaridad, presentan muy baja resistencia serie equivalente y se colocan de cien nanofaradios lo más cerca posible de los pines de alimentación de cada circuito integrado para desacoplo de ruido.
- **Turno 3 (Masculina):** Durante el prototipado en protoboard debemos recordar que cada hilera de cinco agujeros presenta unos dos a cinco picofaradios de capacitancia parásita y resistencia de contacto que degradan señales de alta frecuencia o alta corriente.
- **Turno 4 (Femenina):** Por ello, en el protoboard debemos mantener cables cortos y planos sobre el tablero, distribuir rieles de tierra en estrella y evitar inyectar corrientes superiores a medio amperio o frecuencias mayores a diez megahercios.

Diapositiva 07: Consideraciones Prácticas II: Medición de Tensión y Corriente y Soldadura

Conexión Segura del Multímetro, Efecto de Carga del Instrumento, Soldadura Eutéctica/Sin Plomo y Prevención de Soldaduras Frías

- **Turno 1 (Masculina):** Al medir en laboratorio, el error más destructivo consiste en colocar las puntas del multímetro en paralelo con una fuente mientras los cables están enchufados en la bornera de amperímetro, lo cual provoca un cortocircuito directo y funde el fusible interno.
- **Turno 2 (Femenina):** Asimismo, cuando medimos circuitos de muy alta impedancia, la resistencia interna de diez megaohmios del voltímetro forma un divisor resistivo que reduce la lectura real, fenómeno denominado efecto de carga del instrumento.
- **Turno 3 (Masculina):** Para ensamblar circuitos definitivos mediante soldadura electrónica, limpiamos e hidratamos la punta del cautín entre trescientos cincuenta y trescientos ochenta grados Celsius y calentamos simultáneamente el pad de cobre y el terminal del componente antes de aportar el estaño con alma de resina.
- **Turno 4 (Femenina):** Una unión correcta luce cóncava, brillante y uniforme en menos de tres segundos; mover la pieza antes de que solidifique o calentar solo el alambre produce una soldadura fría de alta resistencia y fallas intermitentes.

Diapositiva 08: Consideraciones Prácticas III: El Osciloscopio en el Laboratorio

Escalas Vertical (V/div) y Horizontal (s/div), Sistema de Disparo (Trigger), Acoplamiento DC/AC y Sondas Atenuadoras 10x

- **Turno 1 (Masculina):** Mientras el multímetro solo entrega un valor numérico promedio o eficaz, el osciloscopio grafica en tiempo real la evolución instantánea del voltaje en el eje vertical frente al tiempo en el eje horizontal.
- **Turno 2 (Femenina):** El circuito de disparo o trigger sincroniza el barrido horizontal cada vez que la señal cruza un umbral de tensión con pendiente ascendente o descendente, logrando estabilizar formas de onda periódicas o capturar transitorios únicos.
- **Turno 3 (Masculina):** El selector de acoplamiento de entrada en modo DC muestra la señal completa incluyendo su nivel continuo, mientras que el modo AC intercala un capacitor serie que bloquea la componente continua para ampliar pequeñas rizadas o señales alternas.
- **Turno 4 (Femenina):** Además, utilizar la sonda en posición diez equis eleva la impedancia de entrada a diez megaohmios con apenas diez picofaradios de carga capacitiva, siempre que hayamos ajustado su trimmer de compensación con la onda cuadrada de calibración.

Diapositiva 09: Consideraciones Prácticas IV: Puesta a Tierra e Interferencia Eléctrica

Tierra de Seguridad (PE) frente a Referencia 0V, Lazos de Tierra (Ground Loops), Acoplamiento EMI y Cable Par Trenzado Blindado

- **Turno 1 (Masculina):** En ingeniería electrónica es vital distinguir entre la tierra física de protección conectada al tercer conductor de la red eléctrica y el nodo común de referencia de cero voltios de nuestro circuito de señal.
- **Turno 2 (Femenina):** Recordemos que el clip de caimán de una sonda estándar de osciloscopio está internamente unido a la tierra física del enchufe; conectarlo a un nodo que no esté a potencial de tierra provocará un cortocircuito inmediato a través del chasis del instrumento.
- **Turno 3 (Masculina):** Cuando dos subsistemas comparten múltiples caminos de retorno de tierra se forma un lazo de tierra o ground loop, donde los campos magnéticos externos inducen corrientes parásitas de cincuenta o sesenta hercios que contaminan las lecturas analógicas.
- **Turno 4 (Femenina):** Para eliminar la interferencia electromagnética separamos los retornos de potencia de la tierra analógica uniéndolos en un único punto estrella, empleamos cables de par trenzado para cancelar campos magnéticos y conectamos la malla de blindaje a tierra en un solo extremo.

Diapositiva 10: Consideraciones Prácticas V: Seguridad Eléctrica y Cierre de Parte 1

Umbrales Fisiológicos de Corriente (mA), Fibrilación Ventricular, Interruptor Diferencial (GFCI) y Reglas de Trabajo Seguro

- **Turno 1 (Masculina):** Cerramos la Parte uno con el tema más importante de todo laboratorio: la seguridad eléctrica, recordando que el daño fisiológico en el cuerpo humano lo produce la corriente que atraviesa el tórax y no simplemente el valor nominal del voltaje.
- **Turno 2 (Femenina):** Apenas un miliamperio es perceptible, entre diez y veinte miliamperios causan tetanización muscular que impide soltar el conductor, y corrientes de treinta a cien miliamperios a través del corazón desencadenan fibrilación ventricular potencialmente letal.
- **Turno 3 (Masculina):** Por ello, todo banco de pruebas debe contar con interruptor diferencial de falla a tierra, transformador de aislamiento cuando se diagnostican equipos conectados a red, descarga previa de capacitores de alta tensión y la regla de trabajar con una sola mano ante tensiones peligrosas.
- **Turno 4 (Femenina):** Con estos fundamentos de circuitos, corriente alterna e instrumentación segura, estamos listos para adentrarnos en la Parte dos sobre electrónica de semiconductores.

4. Firmware C++/MCU: PID Discreto Tustin, FSM y CRC-16 (controlador_tiempo_real_video_02.ino)

```
/*
 * ============================================================================
 * CURSO COMPLETO: FUNDAMENTOS DE ELECTRÓNICA
 * VIDEO 02: Video 02 — Parte 1B: Análisis en Corriente Alterna, Potencia, Transformador, Adaptación de
Impedancias y Consideraciones Prácticas
 * Archivo: controlador_tiempo_real_video_02.ino
 * ----------------------------------------------------------------------------
 * Descripción Técnica:
 *   Firmware determinista en C/C++ Embebido para Microcontrolador (compatible
 *   con arquitectura AVR / ARM Cortex-M STM32 / ESP32 / RP2040) que implementa:
 *     1. Acceso a nivel de registros con punteros 'volatile' y máscaras de bits.
 *     2. Controlador PID Discreto por Transformación Bilineal de Tustin con
 *        derivada sobre la variable medida (-PV), filtro derivativo N y
 *        esquema Anti-Windup por retro-cálculo y Clamping.
 *     3. Máquina de Estados Finitos (FSM de Moore) no bloqueante para
 *        secuenciación segura e interbloqueo de emergencia.
 *     4. Trama de telemetría con verificación CRC-16-CCITT hacia SBC/PLC.
 *
 * Enfoque del Módulo:
 *   Fasores, Impedancia Compleja RLC, Potencia Activa/Reactiva/Aparente, Transformador Ideal, Adaptación Z_L =
Z_th* e Instrumentación
 * ============================================================================
 */

#include <Arduino.h>
#include <stdint.h>

// --- Definición de Pines de Interfase (Entradas Optoaisladas y Salidas PWM/MOSFET) ---
const uint8_t PIN_ADC_PV_SENSOR   = A0; // Entrada analógica PV (ej. Shunt 165 ohm de lazo 4-20 mA)
const uint8_t PIN_ADC_SP_CONSIGNA = A1; // Consigna local o referencia analógica (0..5V / 0..3.3V)
const uint8_t PIN_DI_MARCHA_NO    = 2;  // Entrada optoacoplada PC817: Pulsador de Marcha (NO)
const uint8_t PIN_DI_PARO_NC      = 3;  // Entrada optoacoplada PC817: Paro / Emergencia (Cableado NC)
const uint8_t PIN_PWM_ACTUADOR    = 9;  // Salida PWM hacia Driver MOSFET / Puente H
const uint8_t PIN_DO_SSR_ALARMA   = 8;  // Salida hacia Relé de Estado Sólido (SSR) / Piloto

// --- Parámetros Nominales del Controlador PID Discreto (Video 02) ---
const float KP_GAIN      = 3.500f;
const float TI_SEC       = 0.450f;
const float TD_SEC       = 0.080f;
const float N_FILT       = 12.0f;
const float TS_SEC       = 0.0010f; // Periodo de muestreo Ts = 1 ms
const float U_MIN_LIM    = -15.0f;
const float U_MAX_LIM    = 15.0f;
const float PV_MIN_ENG   = 0.0f;
const float PV_MAX_ENG   = 220.0f;

// --- Variables Volátiles Compartidas con Interrupción (ISR) ---
volatile bool     g_flagTickControl = false;
volatile uint32_t g_contadorPulsosEncoder = 0;

// --- Estructura de Estado del Controlador PID Discreto (Tustin + Anti-Windup) ---
struct ControladorPIDDiscreto {
  float kp;
  float ki;
  float kd;
  float n_divisor;
  float ts;
  float u_min;
  float u_max;
  float integral_acum;
  float termino_d_prev;
  float pv_prev;
  float error_prev;
  float u_salida;
};

ControladorPIDDiscreto g_pid;
```

```
// --- Máquina de Estados Finitos (FSM No Bloqueante de Moore) ---
enum EstadoControlador : uint8_t {
  ESTADO_REPOSO_SEGURO   = 0,
  ESTADO_PRECARGA_SUAVE  = 1,
  ESTADO_LAZO_PID_ACTIVO = 2,
  ESTADO_FALLA_PARO_NC   = 3
};

EstadoControlador g_estadoActual = ESTADO_REPOSO_SEGURO;
uint32_t g_prevMicrosLazo  = 0;
uint32_t g_timestampEstado = 0;
uint32_t g_pasoMuestreo    = 0;

// --- Rutina de Servicio de Interrupción Externa (Encoder / Sensor Rápido) ---
void isrPulsoEncoder() {
  g_contadorPulsosEncoder++;
}

// --- Inicialización de Coeficientes del PID Discreto ---
void inicializarPID(ControladorPIDDiscreto &c) {
  c.kp             = KP_GAIN;
  c.ki             = (TI_SEC > 1e-5f) ? (KP_GAIN / TI_SEC) : 0.0f;
  c.kd             = KP_GAIN * TD_SEC;
  c.n_divisor      = N_FILT;
  c.ts             = TS_SEC;
  c.u_min          = U_MIN_LIM;
  c.u_max          = U_MAX_LIM;
  c.integral_acum  = 0.0f;
  c.termino_d_prev = 0.0f;
  c.pv_prev        = 0.0f;
  c.error_prev     = 0.0f;
  c.u_salida       = 0.0f;
}

// --- Evaluación en Tiempo Real del PID por Regla Trapezoidal de Tustin ---
float evaluarPasoPID_Tustin(ControladorPIDDiscreto &c, float sp, float pv) {
  float error = sp - pv;

  // 1. Término Proporcional
  float p_term = c.kp * error;

  // 2. Término Derivativo sobre -PV filtrado por Tustin (Evita Derivative Kick)
  float tau_d  = (c.kp > 1e-6f) ? ((c.kd / c.kp) / c.n_divisor) : 0.0f;
  float denom  = 2.0f * tau_d + c.ts;
  float alpha1 = (2.0f * tau_d - c.ts) / denom;
  float beta0  = (2.0f * c.kd) / denom;
  float d_term = alpha1 * c.termino_d_prev - beta0 * (pv - c.pv_prev);

  // 3. Candidato de Integración Trapezoidal (Tustin)
  float delta_i = 0.5f * c.ki * c.ts * (error + c.error_prev);
  float i_cand  = c.integral_acum + delta_i;

  // 4. Suma Pre-Saturación y Esquema Anti-Windup Condicional (Clamping)
  float u_unsat = p_term + i_cand + d_term;
  float u_sat   = u_unsat;
  if (u_unsat > c.u_max) {
    u_sat = c.u_max;
    if (error < 0.0f) c.integral_acum = i_cand; // Solo integra si ayuda a desaturar
  } else if (u_unsat < c.u_min) {
    u_sat = c.u_min;
    if (error > 0.0f) c.integral_acum = i_cand;
  } else {
    c.integral_acum = i_cand;
  }

  // Actualizar memoria de estados discretos z^-1
  c.termino_d_prev = d_term;
```

```
  c.pv_prev        = pv;
  c.error_prev     = error;
  c.u_salida       = u_sat;
  return u_sat;
}

// --- Escalado Lineal Equivalente a NORM_X + SCALE_X ---
inline float escalarCuentasADC(int cuentasCrudas, int rawMin, int rawMax, float engMin, float engMax) {
  float u_norm = (float)(cuentasCrudas - rawMin) / (float)(rawMax - rawMin);
  if (u_norm < 0.0f) u_norm = 0.0f;
  if (u_norm > 1.0f) u_norm = 1.0f;
  return engMin + u_norm * (engMax - engMin);
}

// --- Cálculo de Verificación de Redundancia Cíclica CRC-16-CCITT (Polinomio 0x1021) ---
uint16_t calcularCRC16(const uint8_t *datos, uint8_t longitud) {
  uint16_t crc = 0xFFFFU;
  for (uint8_t i = 0; i < longitud; i++) {
    crc ^= ((uint16_t)datos[i] << 8);
    for (uint8_t b = 0; b < 8; b++) {
      if (crc & 0x8000U) {
        crc = (crc << 1) ^ 0x1021U;
      } else {
        crc <<= 1;
      }
    }
  }
  return crc;
}

void setup() {
  Serial.begin(115200);
  pinMode(PIN_DI_MARCHA_NO, INPUT_PULLUP);
  pinMode(PIN_DI_PARO_NC, INPUT_PULLUP);
  pinMode(PIN_PWM_ACTUADOR, OUTPUT);
  pinMode(PIN_DO_SSR_ALARMA, OUTPUT);

  attachInterrupt(digitalPinToInterrupt(PIN_DI_MARCHA_NO), isrPulsoEncoder, RISING);
  inicializarPID(g_pid);
  g_prevMicrosLazo  = micros();
  g_timestampEstado = millis();

  Serial.println(F("# ================================================================"));
  Serial.println(F("# CURSO CONTROLADORES EN MECATRONICA - VIDEO 02"));
  Serial.println(F("# Columnas: t_s,estado_fsm,SP_eng,PV_eng,u_PID,pulsos_enc,crc16"));
  Serial.println(F("# ================================================================"));
}

void loop() {
  uint32_t ahoraMicros = micros();
  uint32_t periodoMicros = (uint32_t)(TS_SEC * 1000000.0f);

  if (ahoraMicros - g_prevMicrosLazo >= periodoMicros) {
    g_prevMicrosLazo += periodoMicros;
    float t_sec = g_pasoMuestreo * TS_SEC;

    // 1. Lectura de Entradas Digitales (Seguridad de Paro NC y Marcha NO)
    bool paroCableadoOK = (digitalRead(PIN_DI_PARO_NC) == HIGH);
    bool ordenMarcha    = (digitalRead(PIN_DI_MARCHA_NO) == LOW);

    // 2. Lectura y Escalado de Entradas Analógicas (NORM_X -> SCALE_X)
    int rawPV = analogRead(PIN_ADC_PV_SENSOR);
    int rawSP = analogRead(PIN_ADC_SP_CONSIGNA);
    float pv_eng = escalarCuentasADC(rawPV, 0, 1023, PV_MIN_ENG, PV_MAX_ENG);
    float sp_eng = escalarCuentasADC(rawSP, 0, 1023, PV_MIN_ENG, PV_MAX_ENG);

    // 3. Evaluación de la Máquina de Estados Finitos (FSM de Moore No Bloqueante)
    if (!paroCableadoOK) {
```

```
    g_estadoActual = ESTADO_FALLA_PARO_NC;
  }

  float mandoActuador = 0.0f;
  switch (g_estadoActual) {
    case ESTADO_REPOSO_SEGURO:
      mandoActuador = 0.0f;
      digitalWrite(PIN_DO_SSR_ALARMA, LOW);
      if (ordenMarcha && paroCableadoOK) {
        g_timestampEstado = millis();
        g_estadoActual    = ESTADO_PRECARGA_SUAVE;
      }
      break;

    case ESTADO_PRECARGA_SUAVE:
      mandoActuador = 0.25f * U_MAX_LIM;
      if (millis() - g_timestampEstado >= 200UL) {
        // Transferencia sin saltos (Bumpless) al activar el lazo PID
        g_pid.integral_acum = mandoActuador;
        g_pid.pv_prev       = pv_eng;
        g_estadoActual      = ESTADO_LAZO_PID_ACTIVO;
      }
      break;

    case ESTADO_LAZO_PID_ACTIVO:
      mandoActuador = evaluarPasoPID_Tustin(g_pid, sp_eng, pv_eng);
      digitalWrite(PIN_DO_SSR_ALARMA, LOW);
      break;

    case ESTADO_FALLA_PARO_NC:
    default:
      mandoActuador = 0.0f;
      g_pid.integral_acum = 0.0f;
      digitalWrite(PIN_DO_SSR_ALARMA, HIGH);
      if (paroCableadoOK && ordenMarcha) {
        g_estadoActual = ESTADO_REPOSO_SEGURO;
      }
      break;
  }

  // 4. Escritura de Salida PWM hacia Etapa de Potencia (MOSFET / Puente H)
  float u_norm_pwm = (mandoActuador - U_MIN_LIM) / (U_MAX_LIM - U_MIN_LIM);
  int duty8bit     = constrain((int)(u_norm_pwm * 255.0f), 0, 255);
  analogWrite(PIN_PWM_ACTUADOR, duty8bit);

  // 5. Lectura Atómica de Contador de Interrupción y Telemetría con CRC-16
  if (g_pasoMuestreo % 10 == 0) {
    noInterrupts();
    uint32_t pulsosCopia = g_contadorPulsosEncoder;
    interrupts();

    uint8_t tramaBytes[6];
    tramaBytes[0] = (uint8_t)g_estadoActual;
    tramaBytes[1] = (uint8_t)duty8bit;
    tramaBytes[2] = (uint8_t)((rawPV >> 8) & 0xFF);
    tramaBytes[3] = (uint8_t)(rawPV & 0xFF);
    tramaBytes[4] = (uint8_t)((pulsosCopia >> 8) & 0xFF);
    tramaBytes[5] = (uint8_t)(pulsosCopia & 0xFF);
    uint16_t crc  = calcularCRC16(tramaBytes, 6);

    Serial.print(t_sec, 3);              Serial.print(',');
    Serial.print((int)g_estadoActual);   Serial.print(',');
    Serial.print(sp_eng, 2);             Serial.print(',');
    Serial.print(pv_eng, 2);             Serial.print(',');
    Serial.print(mandoActuador, 2);      Serial.print(',');
    Serial.print(pulsosCopia);           Serial.print(",0x");
    Serial.println(crc, HEX);
  }
```

```
    g_pasoMuestreo++;
  }
}
```

5. Script Python SBC: Sintonización, Índices IAE/ITAE y Tramas (supervision_sbc_y_sintonizacion_video_02.py)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
================================================================================
CURSO COMPLETO: FUNDAMENTOS DE ELECTRÓNICA
VIDEO 02: Video 02 — Parte 1B: Análisis en Corriente Alterna, Potencia, Transformador, Adaptación de Impedancias y
Consideraciones Prácticas
Archivo: supervision_sbc_y_sintonizacion_video_02.py
--------------------------------------------------------------------------------
Descripción Técnica:
  Script integral en Python para Computadora de Placa Única (Raspberry Pi /
  SBC) y estación de ingeniería que implementa:
    1. Simulación en lazo cerrado de un controlador PID discreto (Tustin +
       Filtro Derivativo N + Anti-Windup por Clamping y Back-Calculation).
    2. Cálculo cuantitativo de métricas transitorias (tr, tp, Mp, ts) y de
       los cuatro índices integrales de desempeño (IAE, ISE, ITAE, ITSE).
    3. Reglas automáticas de sintonización de Ziegler-Nichols (1.º y 2.º
       método), Cohen-Coon y sintonización robusta IMC / Lambda.
    4. Construcción y validación de tramas binarias Maestro-Coprocesador
       (SBC <-> MCU/PLC) protegidas con suma de verificación CRC-16-CCITT.
================================================================================
"""

import math
import struct
from typing import Dict, List, Tuple

# Parámetros nominales del Video 02
KP_NOMINAL = 3.5
TI_NOMINAL = 0.45
TD_NOMINAL = 0.08
N_FILTRO   = 12
TS_SEG     = 0.0010
U_MIN      = -15
U_MAX      = 15


def sintonizar_pid_ziegler_nichols_y_lambda(
    K_planta: float, L_retardo: float, T_constante: float, lambda_deseado: float
) -> Dict[str, Dict[str, float]]:
    """
    Calcula las ganancias PID (Kp, Ti, Td) mediante:
      - Ziegler-Nichols 1.er Método (Curva de Reacción FOPDT)
      - Método de Cohen-Coon (para retardo significativo L/T)
      - Método IMC / Regla Lambda (respuesta sin sobreimpulso)
    """
    r = L_retardo / max(T_constante, 1e-6)

    # 1. Ziegler-Nichols Lazo Abierto (PID Estándar ISA)
    kp_zn = (1.2 * T_constante) / (K_planta * max(L_retardo, 1e-6))
    ti_zn = 2.0 * L_retardo
    td_zn = 0.5 * L_retardo

    # 2. Cohen-Coon (PID)
    kp_cc = (1.0 / (K_planta * r)) * (4.0 / 3.0 + r / 4.0)
    ti_cc = L_retardo * (32.0 + 6.0 * r) / (13.0 + 8.0 * r)
    td_cc = L_retardo * 4.0 / (11.0 + 2.0 * r)

    # 3. IMC / Lambda (PI robusto sin sobreimpulso + término D para retardo Padé)
    kp_imc = T_constante / (K_planta * (lambda_deseado + L_retardo))
    ti_imc = T_constante
    td_imc = (T_constante * L_retardo) / (2.0 * T_constante + L_retardo)

    return {
        "Ziegler_Nichols_FOPDT": {
            "Kp": round(kp_zn, 4), "Ti_s": round(ti_zn, 4), "Td_s": round(td_zn, 4)
        },
```

```
        "Cohen_Coon": {
            "Kp": round(kp_cc, 4), "Ti_s": round(ti_cc, 4), "Td_s": round(td_cc, 4)
        },
        "IMC_Lambda_Robusto": {
            "Kp": round(kp_imc, 4), "Ti_s": round(ti_imc, 4), "Td_s": round(td_imc, 4)
        },
    }


def simular_lazo_pid_discreto_tustin(
    kp: float, ti: float, td: float, n_filt: float, ts: float, pasos: int = 800
) -> Dict[str, float]:
    """
    Simula una planta mecatrónica de 2.º orden controlada por un PID discreto
    discretizado por Tustin con Anti-Windup y calcula tr, Mp, ts, IAE, ISE, ITAE.
    """
    ki = kp / ti if ti > 1e-6 else 0.0
    kd = kp * td
    tau_d = (td / n_filt) if n_filt > 0 else 0.0
    denom_d = 2.0 * tau_d + ts
    alpha_d = (2.0 * tau_d - ts) / denom_d
    beta_d  = (2.0 * kd) / denom_d

    # Estado de la planta (wn = 8 rad/s, zeta = 0.45, K = 1.0)
    wn = 8.0
    zeta = 0.45
    y_pos = 0.0
    y_vel = 0.0
    sp = 1.0

    integral = 0.0
    d_prev = 0.0
    y_prev = 0.0
    e_prev = 0.0

    iae = 0.0
    ise = 0.0
    itae = 0.0
    itse = 0.0
    y_max = 0.0
    t_pico = 0.0
    t_10 = None
    t_90 = None
    t_estable = 0.0

    for k in range(pasos):
        t = k * ts
        error = sp - y_pos

        # Acumulación de índices integrales de desempeño
        abs_e = abs(error)
        iae  += abs_e * ts
        ise  += (error ** 2) * ts
        itae += (t * abs_e) * ts
        itse += (t * (error ** 2)) * ts

        # Algoritmo PID Discreto con filtro derivativo sobre -PV
        p_term = kp * error
        d_term = alpha_d * d_prev - beta_d * (y_pos - y_prev)
        i_cand = integral + 0.5 * ki * ts * (error + e_prev)

        u_unsat = p_term + i_cand + d_term
        u_cmd = max(U_MIN, min(U_MAX, u_unsat))

        # Anti-Windup condicional (Clamping)
        if u_unsat == u_cmd or (u_unsat > U_MAX and error < 0) or (u_unsat < U_MIN and error > 0):
            integral = i_cand
```

```
        d_prev = d_term
        y_prev = y_pos
        e_prev = error

        # Integración de la planta física (Euler semi-implícito)
        acel = (wn ** 2) * u_cmd - 2.0 * zeta * wn * y_vel - (wn ** 2) * y_pos
        y_vel += acel * ts
        y_pos += y_vel * ts

        if y_pos >= 0.10 * sp and t_10 is None:
            t_10 = t
        if y_pos >= 0.90 * sp and t_90 is None:
            t_90 = t
        if y_pos > y_max:
            y_max = y_pos
            t_pico = t
        if abs(sp - y_pos) > 0.02 * sp:
            t_estable = t

    tr = (t_90 - t_10) if (t_10 is not None and t_90 is not None) else 0.0
    mp_pct = max(0.0, (y_max - sp) / sp * 100.0)

    return {
        "tiempo_subida_tr_s": round(tr, 4),
        "tiempo_pico_tp_s": round(t_pico, 4),
        "sobreimpulso_Mp_pct": round(mp_pct, 2),
        "tiempo_establecimiento_ts_2pct_s": round(t_estable, 4),
        "IAE": round(iae, 4),
        "ISE": round(ise, 4),
        "ITAE": round(itae, 4),
        "ITSE": round(itse, 4),
    }


def crc16_ccitt(data: bytes) -> int:
    """Calcula CRC-16-CCITT (0x1021, init 0xFFFF) para enlace serie SBC <-> MCU."""
    crc = 0xFFFF
    for byte in data:
        crc ^= (byte << 8)
        for _ in range(8):
            if crc & 0x8000:
                crc = ((crc << 1) ^ 0x1021) & 0xFFFF
            else:
                crc = (crc << 1) & 0xFFFF
    return crc


def empaquetar_trama_consigna_sbc(comando: int, setpoint_float: float) -> bytes:
    """Construye trama binaria: [0xAA, 0x55, CMD, LEN=4, float32_LE, CRC16_BE]."""
    cabecera_y_payload = struct.pack("<BBBf", 0xAA, 0x55, comando, 4, setpoint_float)
    crc = crc16_ccitt(cabecera_y_payload)
    return cabecera_y_payload + struct.pack(">H", crc)


if __name__ == "__main__":
    print("=" * 78)
    print(" CURSO: FUNDAMENTOS DE ELECTRÓNICA — VIDEO 02")
    print(" Enfoque:", "Fasores, Impedancia Compleja RLC, Potencia Activa/Reactiva/Aparente, Transformador Ideal,
Adaptación Z_L = Z_th* e Instrumentación")
    print("=" * 78)

    tablas = sintonizar_pid_ziegler_nichols_y_lambda(
        K_planta=1.25, L_retardo=0.15, T_constante=0.85, lambda_deseado=0.35
    )
    print("\n[1] Sintonización Comparativa (Ziegler-Nichols, Cohen-Coon e IMC Lambda):")
    for metodo, gan in tablas.items():
        print(f"    {metodo:24s} -> Kp = {gan['Kp']:6.3f} | Ti = {gan['Ti_s']:6.3f} s | Td = {gan['Td_s']:6.3f}
s")
```

```
metricas = simular_lazo_pid_discreto_tustin(KP_NOMINAL, TI_NOMINAL, TD_NOMINAL, N_FILTRO, 0.005)
print("\n[2] Desempeño Temporal e Índices Integrales (IAE, ISE, ITAE, ITSE):")
for k_m, v_m in metricas.items():
    print(f"    {k_m:34s} : {v_m}")

trama = empaquetar_trama_consigna_sbc(comando=0x01, setpoint_float=110)
print("\n[3] Trama Binaria Híbrida SBC -> MCU con CRC-16-CCITT:", trama.hex(" ").upper())
```

6. Programa PLC IEC 61131-3 (ST + IL): Enclavamiento, GRAFCET y Escalado 4-20 mA (automatizacion_plc_iec61131_video_02.st)

```
(* ================================================================================
 * CURSO COMPLETO: FUNDAMENTOS DE ELECTRÓNICA
 * VIDEO 02: Video 02 — Parte 1B: Análisis en Corriente Alterna, Potencia, Transformador, Adaptación de
Impedancias y Consideraciones Prácticas
 * Archivo: automatizacion_plc_iec61131_video_02.st
 * --------------------------------------------------------------------------------
 * Estándar: IEC 61131-3 — Texto Estructurado (ST) + Equivalencia en Lista de
 *           Instrucciones (IL) y Peldaños de Escalera (Ladder Diagram)
 * Descripción Técnica:
 *   Bloque de Función Industrial (FB_ControladorCeldaMecatronica_V02) que
 *   integra el Ciclo de Barrido (Scan), circuito de enclavamiento prioritario
 *   al paro (cableado NC), secuenciador GRAFCET de 4 etapas, temporizador TON,
 *   contador de lotes CTU, registro de corrimiento de rechazo de calidad SHRB,
 *   escalado analógico 4-20 mA (NORM_X / SCALE_X con detección de rotura de
 *   hilo < 3.6 mA) y algoritmo PID discreto con Anti-Windup.
 * ================================================================================ *)

FUNCTION_BLOCK FB_ControladorCeldaMecatronica_V02
VAR_INPUT
    (* Entradas Digitales de la Tabla de Imagen de Proceso PII (%IX) *)
    xPulsadorMarcha_NO   : BOOL; (* %IX0.0 - Contacto Normalmente Abierto *)
    xPulsadorParo_NC     : BOOL; (* %IX0.1 - Cableado Físicamente NC en Campo *)
    xReleTermico_NC      : BOOL; (* %IX0.2 - Guardamotor NC (1 = Sin Falla) *)
    xSensorPiezaEntrada  : BOOL; (* %IX0.3 - Sensor Inductivo PNP 24V DC *)
    xCamaraPiezaDefecto  : BOOL; (* %IX0.4 - 1 = Envase Defectuoso Detectado *)
    xPulsoEncoderBanda   : BOOL; (* %IX0.5 - Reloj de Desplazamiento de Banda *)
    xResetContadores     : BOOL; (* %IX0.6 - Reinicio Manual de Lote *)

    (* Entradas Analógicas Crudas del Convertidor ADC (%IW) *)
    iEntradaAnalogicaRaw : INT;  (* %IW64  - Lazo 4-20 mA: 0 a 27648 cuentas *)
    rSetpointIngenieria  : REAL := 132.0;
END_VAR

VAR_OUTPUT
    (* Salidas Digitales hacia la Tabla de Imagen de Salidas PIQ (%QX) *)
    xMotorBandaContactor : BOOL; (* %QX0.0 - Contactor Principal Enclavado *)
    xValvulaDosificadora : BOOL; (* %QX0.1 - Solenoide 24V DC con Diodo Flyback *)
    xCilindroExpulsor    : BOOL; (* %QX0.2 - Rechazo en Posición Bit 5 del SHRB *)
    xAlarmaRoturaCable   : BOOL; (* %QX0.3 - Corriente < 3.6 mA o Paro Activo *)
    xLoteCompletado      : BOOL; (* %QX0.4 - Salida Q del Contador CTU *)

    (* Salida Analógica Escalada y Variables de Diagnóstico HMI *)
    iSalidaAnalogicaRaw  : INT;  (* %QW80  - Mando 0..27648 hacia Variador/SSR *)
    rVariableProcesoPV   : REAL; (* Variable física escalada en unidades de ing. *)
    rSalidaControlPID    : REAL; (* Esfuerzo de control 0.0 a 100.0 % *)
    iEstadoGrafcet       : INT;  (* Etapa activa del GRAFCET: 0, 10, 20, 30 *)
END_VAR

VAR
    (* Instancias de Temporizadores y Contadores Estándar IEC 61131-3 *)
    tonDosificado        : TON;
    ctuContadorLotes     : CTU;

    (* Variables Internas de Escalado NORM_X / SCALE_X *)
    rValorNormalizado    : REAL;
    rMinIngenieria       : REAL := 0.0;
    rMaxIngenieria       : REAL := 220.0;

    (* Registro de Corrimiento de 16 Bits (Shift Register para Rechazo) *)
    wRegistroCorrimiento : WORD := 16#0000;
    xPulsoEncoderPrev    : BOOL := FALSE;
    xFlancoEncoder       : BOOL := FALSE;

    (* Memoria Retentiva del Bloque PID Discreto *)
    rKp                  : REAL := 3.500;
    rTi_s                : REAL := 0.450;
    rTd_s                : REAL := 0.080;
```

```
    rTs_s                : REAL := 0.0010;
    rIntegralAcum        : REAL := 0.0;
    rErrorPrev           : REAL := 0.0;
    rPvPrev              : REAL := 0.0;
END_VAR

(* ================================================================================
 * 1. EQUIVALENCIA EN LISTA DE INSTRUCCIONES (IL) DEL PELDAÑO DE ENCLAVAMIENTO:
 *      LD     xPulsadorMarcha_NO
 *      OR     xMotorBandaContactor
 *      AND    xPulsadorParo_NC
 *      AND    xReleTermico_NC
 *      ANDN   xAlarmaRoturaCable
 *      ST     xMotorBandaContactor
 * ================================================================================ *)
xMotorBandaContactor := (xPulsadorMarcha_NO OR xMotorBandaContactor)
                        AND xPulsadorParo_NC
                        AND xReleTermico_NC
                        AND NOT xAlarmaRoturaCable;

(* ================================================================================
 * 2. NORMALIZACIÓN (NORM_X) Y ESCALADO LINEAL (SCALE_X) DE LAZO 4-20 mA
 *    Detección de rotura de hilo si iEntradaAnalogicaRaw < -1728 (< 3.0 mA)
 * ================================================================================ *)
IF (iEntradaAnalogicaRaw < -1728) OR (NOT xPulsadorParo_NC) OR (NOT xReleTermico_NC) THEN
    xAlarmaRoturaCable := TRUE;
    rValorNormalizado  := 0.0;
ELSE
    xAlarmaRoturaCable := FALSE;
    (* NORM_X: Convierte 0..27648 cuentas a rango flotante 0.0..1.0 *)
    rValorNormalizado  := INT_TO_REAL(iEntradaAnalogicaRaw) / 27648.0;
    IF rValorNormalizado < 0.0 THEN rValorNormalizado := 0.0; END_IF;
    IF rValorNormalizado > 1.0 THEN rValorNormalizado := 1.0; END_IF;
END_IF;

(* SCALE_X: Convierte 0.0..1.0 a Unidades de Ingeniería [rMinIngenieria .. rMaxIngenieria] *)
rVariableProcesoPV := rMinIngenieria + rValorNormalizado * (rMaxIngenieria - rMinIngenieria);

(* ================================================================================
 * 3. REGISTRO DE CORRIMIENTO DE BITS (SHRB / BSL) SINCRONIZADO CON ENCODER
 * ================================================================================ *)
xFlancoEncoder    := xPulsoEncoderBanda AND NOT xPulsoEncoderPrev;
xPulsoEncoderPrev := xPulsoEncoderBanda;

IF xFlancoEncoder AND xMotorBandaContactor THEN
    wRegistroCorrimiento := SHL(IN := wRegistroCorrimiento, N := 1);
    IF xCamaraPiezaDefecto THEN
        wRegistroCorrimiento := wRegistroCorrimiento OR 16#0001;
    END_IF;
END_IF;

(* Si el bit desplazado llega a la estación 5 (bit 4, máscara 16#0010), activa expulsor *)
xCilindroExpulsor := (wRegistroCorrimiento AND 16#0010) <> 16#0000;

(* ================================================================================
 * 4. SECUENCIADOR GRAFCET / SFC DE 4 ETAPAS CON TEMPORIZADOR TON Y CONTADOR CTU
 * ================================================================================ *)
IF NOT xMotorBandaContactor THEN
    iEstadoGrafcet := 0;
END_IF;

CASE iEstadoGrafcet OF
    0: (* Etapa 0: Reposo esperando pieza en estación *)
        xValvulaDosificadora := FALSE;
        IF xMotorBandaContactor AND xSensorPiezaEntrada THEN
            iEstadoGrafcet := 10;
        END_IF;
```

```
    10: (* Etapa 10: Dosificación temporizada con bloque TON *)
        xValvulaDosificadora := TRUE;
        IF tonDosificado.Q THEN
            xValvulaDosificadora := FALSE;
            iEstadoGrafcet       := 20;
        END_IF;

    20: (* Etapa 20: Incremento de contador de lote y espera de liberación *)
        xValvulaDosificadora := FALSE;
        IF NOT xSensorPiezaEntrada THEN
            iEstadoGrafcet := 0;
        END_IF;
END_CASE;

tonDosificado(IN := (iEstadoGrafcet = 10), PT := T#1500MS);
ctuContadorLotes(CU := (iEstadoGrafcet = 20), R := xResetContadores, PV := 50);
xLoteCompletado := ctuContadorLotes.Q;

(* ================================================================================
 * 5. BLOQUE TECNOLÓGICO PID DISCRETO CON ANTI-WINDUP Y DES-ESCALADO ANALÓGICO
 * ================================================================================ *)
IF xMotorBandaContactor AND NOT xAlarmaRoturaCable THEN
    VAR
        rErrorActual : REAL;
        rTerminoP    : REAL;
        rTerminoD    : REAL;
        rCandI       : REAL;
        rSumaUnsat   : REAL;
    END_VAR
    rErrorActual := rSetpointIngenieria - rVariableProcesoPV;
    rTerminoP    := rKp * rErrorActual;
    rTerminoD    := -rKp * rTd_s * (rVariableProcesoPV - rPvPrev) / rTs_s;
    rCandI       := rIntegralAcum + (rKp / rTi_s) * rTs_s * 0.5 * (rErrorActual + rErrorPrev);

    rSumaUnsat   := rTerminoP + rCandI + rTerminoD;
    IF rSumaUnsat > 100.0 THEN
        rSalidaControlPID := 100.0;
    ELSIF rSumaUnsat < 0.0 THEN
        rSalidaControlPID := 0.0;
    ELSE
        rSalidaControlPID := rSumaUnsat;
        rIntegralAcum     := rCandI;
    END_IF;

    rErrorPrev := rErrorActual;
    rPvPrev    := rVariableProcesoPV;
ELSE
    rSalidaControlPID := 0.0;
    rIntegralAcum     := 0.0;
END_IF;

(* Conversión de 0.0..100.0 % a 0..27648 cuentas crudas de salida analógica %QW80 *)
iSalidaAnalogicaRaw := REAL_TO_INT((rSalidaControlPID / 100.0) * 27648.0);

END_FUNCTION_BLOCK
```