INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA Y TECNOLOGÍAS AVANZADAS

TÓPICOS AVANZADOS DE AUTOMATIZACIÓN

15 de diciembre de 2023

Alumnos:
Buenabad García Alberto
Garibay Molina Carlos
Zambrano Ramírez Diego

CONTROL DE TEMPERATURA

Profesor: David Abraham Morales Enrique

Grupo:4MM9

OBJETIVO.

- Diseño y sintonización de un control PID con Predictor de Smith para el sistema de control de temperatura.

INTRODUCCIÓN.

El Predictor de Smith, también conocido como Smith Predictor en inglés, es una herramienta fundamental en el ámbito del control automático. Su estructura se basa en un modelo interno que permite la predicción de la salida de un proceso en tiempo real, incluso en presencia de retardos. Este predictor resulta especialmente valioso en situaciones donde la anticipación precisa del comportamiento del sistema es crucial.

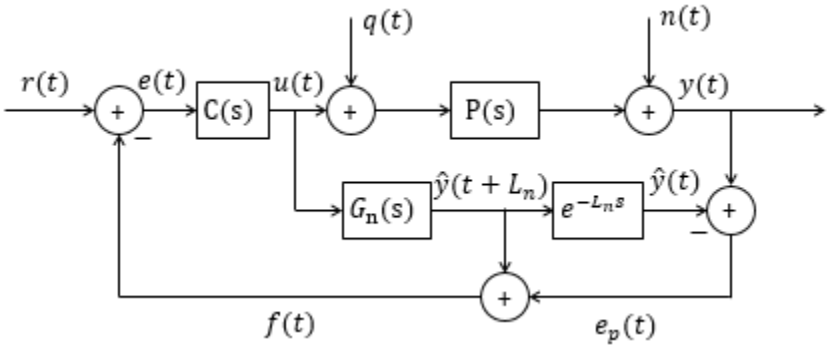
MATERIALES.

- Placa arduino
- Sensor de temperatura LM35
- Bombilla.
- Dimmer JTDB.
- Cables variados.

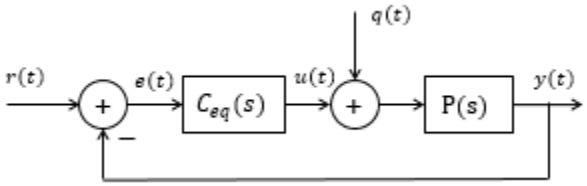
DESARRROLLO.

Estructura del Predictor de Smith.

La estructura del predictor de Smith (Smith predictor en ingles) viene dado por la siguiente representación de modelo interno:



En la estructura el control primario es C(s), el proceso real es P(s), el modelo rápido o modelo sin retardo es Gn(s) y el retardo viene dado por la expresión: e^-Lns
Note además que si el modelo es igual a el proceso real implica que P_n(s)=G_n(s)e^{-L_ns}=P(s).
Este compensador de tiempo muerto es capaz de predecir la salida del proceso real, y(t), por medio de la dinámica sin retardo G_n, o sea que con esa estructura el control es capaz de predecir el comportamiento del proceso real, P(s), en un tiempo igual al retardo Ln.
El método de Smith en Control, se puede reducir utilizando el algebra de bloques a la siguiente representación equivalente:



donde el control esquivalente viene dado por:

$$C_{eq} = \frac{C(s)}{1 + C(s)G_n(s) - C(s)P_n(s)}$$

Así las relaciones de la entrada r(t) y la salida y(t) suponiendo P(s)=P_n(s) viene dado por:

$$H_{yr}(s) = \frac{y(t)}{r(t)} = \frac{C(s)P_n(s)}{1 + C(s)G_n(s)}$$

Notamos que gracias a la estructura del predictor de Smith, la ecuación característica del proceso queda libre de la influencia del retardo.

Por otro lado, la relación de la entrada q(t) y la salida y(t) viene dado por:

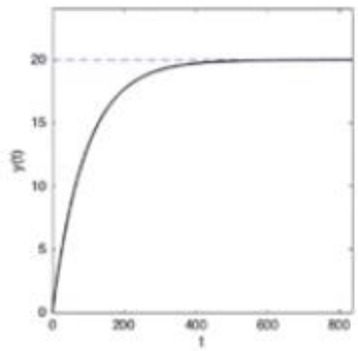
$$H_{yq}(s) = \frac{y(t)}{q(t)} = P_n(s)\left[1 - \frac{C(s)P_n(s)F_r(s)}{1 + C(s)G_n(s)}\right]$$

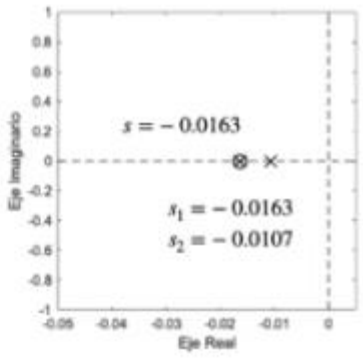
El código proporcionado implementa un sistema de control de temperatura mediante un controlador PID y un Predictor de Smith. A continuación, se presenta un desarrollo detallado de las secciones clave del código.

- En esta sección, se definen variables que determinan el modo de operación, los parámetros para análisis a lazo abierto y cerrado, así como las constantes del controlador PID.
- En la función setup(), se inicializan los pines de entrada y salida, y se configuran las interrupciones para detectar el cruce por cero en la línea de alimentación.
- En el bucle principal (loop()), se realiza el control del dimmer, la detección del cruce por cero, la lectura de la temperatura, y la implementación del control PID y el Predictor de Smith. Los resultados se envían al Serial Plotter.
- La función smithP() implementa el Predictor de Smith, calculando la salida estimada con retardo (yE), el error de predicción (ep), la salida del modelo rápido sin retardo (yR), y la salida final de predicción (fk). También se actualiza el vector de la ley de control.

Considerando lo siguiente:

$$G_p(s) = \frac{0.66}{61.4s + 1} \qquad K_C = 1 \qquad \tau_I = 61.4 \qquad G_{CL}(s) = \frac{61.4\ s + 1}{5712\ s^2 + 154.43\ s + 1}$$





El código demuestra la integración de un controlador PID y un Predictor de Smith en un sistema de control de temperatura, proporcionando un enfoque versátil para la regulación térmica en función de las necesidades específicas del usuario.

ANÁLISIS DE RESULTADOS.

Ilustración 1 - Prueba de potencia del foco

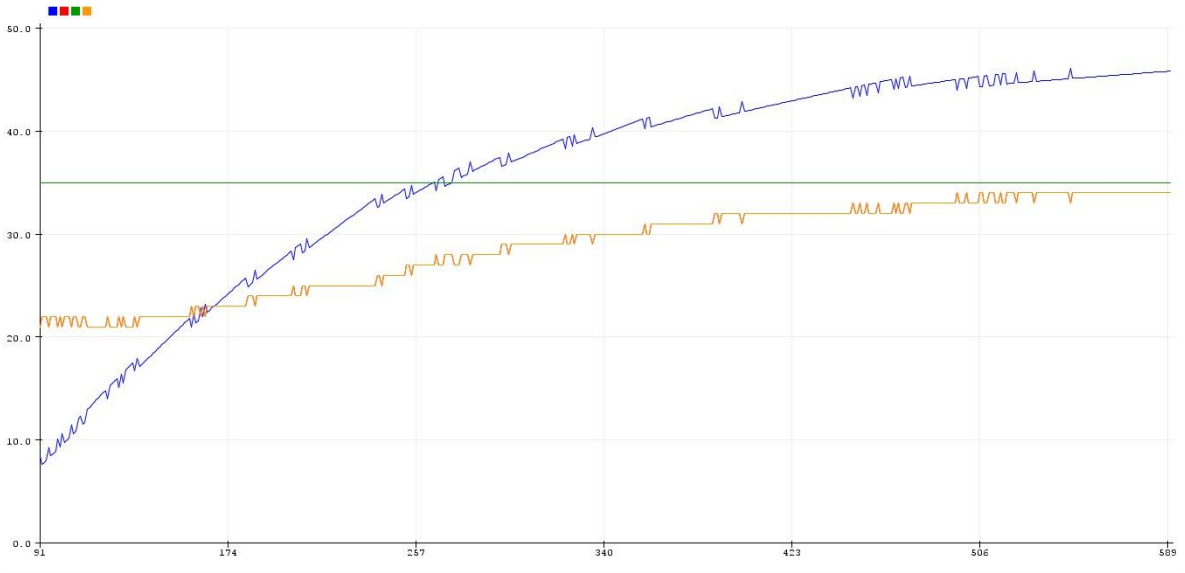


Ilustración 2 - Prueba de predicción

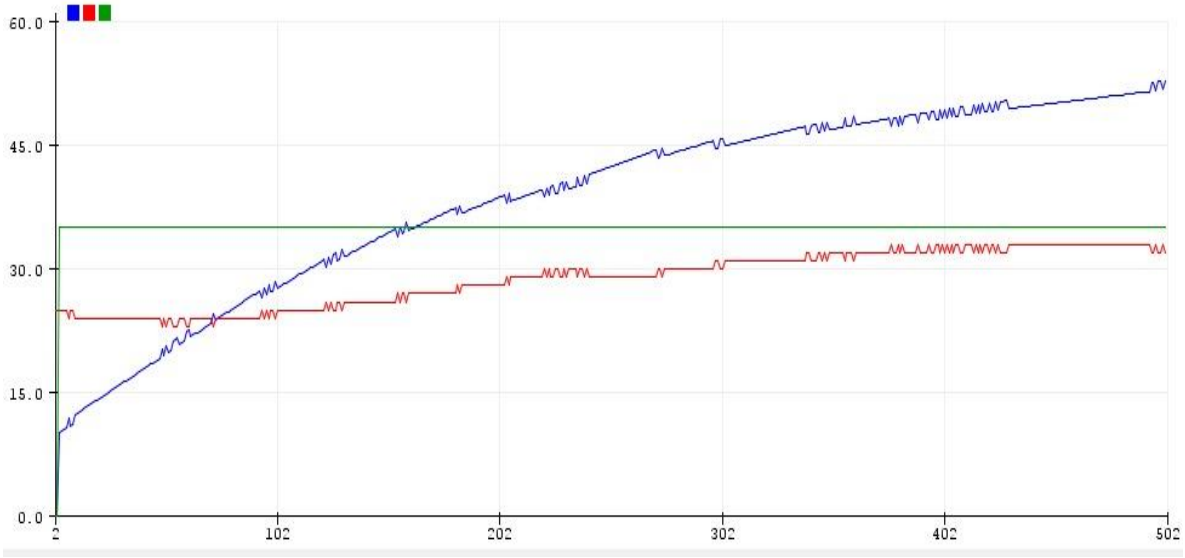


Ilustración 3 – Setpoint

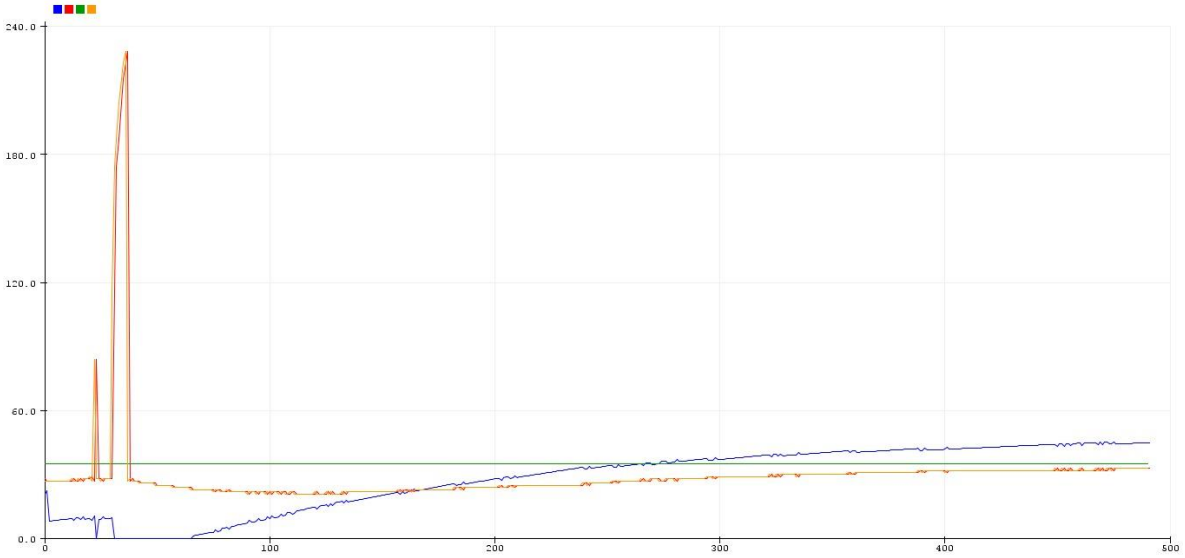


Ilustración 4 - Movimiento de potencial y control de temperatura

CONCLUSIONES INDIVIDUALES.

La inclusión de dos modos de operación, manual y automático, permite una amplia gama de estrategias de control. En el modo manual, el análisis a lazo abierto facilita la comprensión y ajuste del sistema. Por otro lado, el modo automático, basado en un control PID y un Predictor de Smith, ofrece una capacidad adaptativa avanzada. Esta versatilidad brinda a los operadores la opción de seleccionar la estrategia más

adecuada según los requisitos específicos del proceso térmico.
Zambrano Ramírez Diego

La inclusión del Predictor de Smith demuestra su valía al abordar los desafíos asociados con retardos en el sistema de control térmico. Este componente anticipa la salida del proceso incluso en presencia de retardos conocidos, mitigando los efectos adversos en la respuesta del sistema. La capacidad de previsión del Predictor de Smith se traduce en una mayor eficiencia operativa y una mejor capacidad de respuesta, especialmente en situaciones donde el tiempo de retardo puede afectar significativamente el rendimiento del sistema.
Buenabad García Carlos.

La implementación del control PID en el modo automático proporciona una respuesta más precisa y rápida al error de temperatura. El ajuste proporcional, integral y derivativo del PID permite mantener la temperatura deseada con mayor precisión, minimizando la oscilación y mejorando la estabilidad del sistema. Esto resulta crucial en aplicaciones donde la precisión térmica es crítica, como en procesos industriales o experimentos de laboratorio.
Garibay Molina Carlos.

ANEXO 1

Código de programación del control de temperatura:

control_temp §

```
int Modo = 2;  // Modo de operación 1: Modo manual (lazo abierto)
               //                    2: Modo automático (lazo cerrado)

// Parámetros para análisis a lazo abierto
float Potencia_1 = 0;     // Valor inicial del cambio escalón. (min = 0)
float Potencia_2 = 0;     // Valor final del cambio escalón. (max = 100)

// Parámetros para análisis a lazo cerrado
float Setpoint = 35;       // Celsius

// Constantes de PID
float Kc = 2; float Tao_I = 32;

int Tiempo0 = 5000;     // Retardo (en milisegundos) para ejecutar cambio escalón cuando se encuentra

int A = 0;           // Pin A0 de entrada analógica para sensor LM35 (Variable de salida)
float Potencia = 0;  // Potencia inicial enviada al dimmer en rango de 0 a 100 (Variable de entrada)

//  Declaración de variables
int pin_disparo = 3;
int pin_cruce_cero = 8;
int last_CH1_state = 0;
int detectado = 0;
int valor = 0;

unsigned long Tiempo_previo = 0;
unsigned long Tiempo_actual = 0;
int Read_Delay = 1000;     // Periodo de muestreo en milisegundos
int Temperatura = 0;       // Celsius
float sp = 0;

// Variables para PID
float PID_error = 0;
float previous_error = 0;
float PID_value = 0;
float Error_INT = 0;

// Variables para el predictor de Smith
float b0 = 0, b1 = 0.5423, a1 = -0.999;
int d = 10 - 1;  // Retardo
float up[15];    // Vector de entradas con retardo
int kT = 14;     // Tiempo discreto actual
float yE = 0, yE_1 = 0;  // Salida estimada con retardo
float ep = 0;             // Error de predicción
float yR = 0, yR_1 = 0;    // Salida del modelo rápido (sin retardo)
```

```
void loop() {
  Tiempo_actual = millis(); // Tiempo Actual
  valor = map(Potencia, 0, 100, 7600, 10);

  if (detectado) {
    delayMicroseconds(valor);
    digitalWrite(3, HIGH);
    delayMicroseconds(100);
    digitalWrite(3, LOW);
    detectado = 0;

    if (Tiempo_actual - Tiempo_previo >= Read_Delay) {
      Tiempo_previo += Read_Delay;

      Temperatura = 5.0 * 100.0 * analogRead(A) / 1024.0;  // Lectura del sensor LM35

      // Predictor de Smith
      smithP();

      // PID
      if (Modo == 1) {
        // Modo manual (lazo abierto)
        if (Tiempo_actual <= Tiempo0) {
          Potencia = Potencia_1;
        } else if (Tiempo_actual >= Tiempo0) {
          Potencia = Potencia_2;
        }
      } else if (Modo == 2) {
        // Modo automático (lazo cerrado)
        PID_error = Setpoint - Temperatura;                    // Cálculo del error
        Error_INT = Error_INT + PID_error * (1000 / Read_Delay);  // Cálculo de la integral del error
        PID_value = Kc * (PID_error + (1 / Tao_I) * Error_INT);     // Cálculo de la salida del controlador PI

        sp = Setpoint;

        // Límite de salida del controlador
        if (PID_value < 0) {
          PID_value = 0;
        }
        if (PID_value > 100) {
          PID_value = 100;
        }

        Potencia = PID_value;   // Asignación a la entrada de la planta.
      }
```

```
      // Mostrar en Serial Plotter
      Serial.print(Potencia);
      Serial.print(" ");
      Serial.print(Temperatura);
      Serial.print(" ");
      Serial.print(sp);
      Serial.print(" ");
      Serial.println(fk);
    }
  }
}

// Rutina de interrupción
ISR(PCINT0_vect) {
  if (PINB & B00000001) {  // Entrada desde el optoacoplador
    if (last_CH1_state == 0) {
      detectado = 1;
    }
  } else if (last_CH1_state == 1) {
    detectado = 1;
    last_CH1_state = 0;
  }
}

// Predictor de Smith
void smithP() {
  // Ecuación en diferencias de la planta discreta con retardo (Y estimada)
  yE = b0 * up[kT - d] + b1 * up[kT - 1 - d] - a1 * yE_1;D
  // Error de predicción (Temperatura del proceso - Y estimada)
  ep = Temperatura - yE;
  // Salida del modelo rápido (sin retardo) Y rápida
  yR = b0 * up[kT] + b1 * up[kT - 1] - a1 * yR_1;
  // Suma entre error de predicción y la salida rápida
  fk = yR + ep;

 // Desplazar el vector de la ley de control
  for (int i = 1; i <= kT; i++) {
    up[i - 1] = up[i];
  }
 // Actualizar los valores pasados del predictor
  yE_1 = yE;
  yR_1 = yR;
}
```