

HORNO PID

HORNO PID

Electrónica para Ciencias

Juan Barbosa
Universidad de los Andes

Luisa Rodríguez
Universidad de los Andes



CONTENIDOS

1	Controlador PID	1
1.1	Calibración de la temperatura	2
1.2	Caracterización del actuador	2
1.3	Sistema de alimentación	3
1.4	Controlador	4
1.5	Digitalización	6
	Referencias	7
2	Algoritmo	9
	Referencias	10

CAPÍTULO 1

CONTROLADOR PID

Con el objetivo de controlar la temperatura sobre un actuador, se realiza un circuito con controlador PID. El circuito en su totalidad se compone de 5 partes principales:

1. **Sistema de alimentación:** tiene como objetivo generar los 5 V necesarios para encender la Raspberry Pi, así como fijar el valor máximo para la referencia.
2. **Sensor de temperatura:** transforma la temperatura del actuador en una variable eléctrica que es medida por el controlador y mostrada al usuario en tiempo real.
3. **Valor de referencia:** corresponde con el valor de temperatura al que se desea llevar el accionador.
4. **Controlador PID:** continuamente evalúa la diferencia entre el valor de referencia y la temperatura actual, además de aplicar corrección que corresponden con términos proporcionales, integradores y derivativos.
5. **Circuito de accionamiento:** tiene como objetivo proporcionar la potencia necesaria para que el actuador funcione de manera correcta.

Estas partes se pueden observar en la Figura 1.1, donde se dibuja el circuito usado para el horno PID. En ella cada parte se encuentra señalada con un color específico, siendo el sistema de alimentación, rojo; el sensor de temperatura, verde; el valor de referencia, azul; el controlador, naranja; y el circuito de accionamiento en violeta.

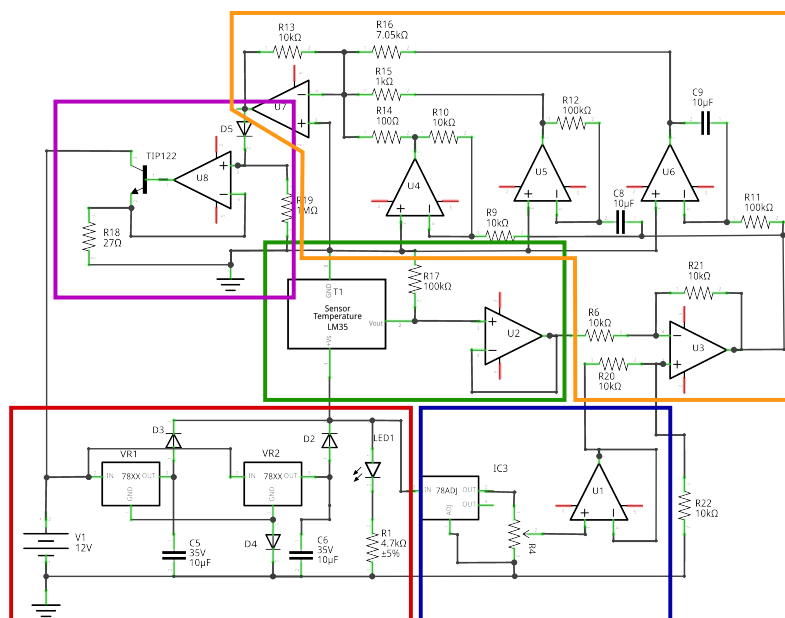


Figura 1.1 Circuito PID, implementado para el control de la temperatura.

1.1 Calibración de la temperatura

El sensor de temperatura corresponde con un LM35, circuito integrado con respuesta lineal a la temperatura, con un factor de escala de $10 \text{ mV}/^{\circ}\text{C}$ y un rango de operación de -55°C a 150°C [1]. A pesar de ser un sensor calibrado, se realiza una calibración *in situ* usando una termocupla de NiCr-Ni Phywe.

Los resultados se muestran en la Figura 1.2, de donde se observa la linealidad del sensor usado, tanto la pendiente como el intercepto muestran desfases del valor esperado (1 y 0, correspondientemente) sobre la segunda cifra decimal, discrepancias menores al 6 %.

1.2 Caracterización del actuador

Inicialmente la construcción del horno usaba una celda Peltier, no obstante esta dejó de funcionar. Por este motivo se modifica el actuador a una resistencia de 27Ω y 5

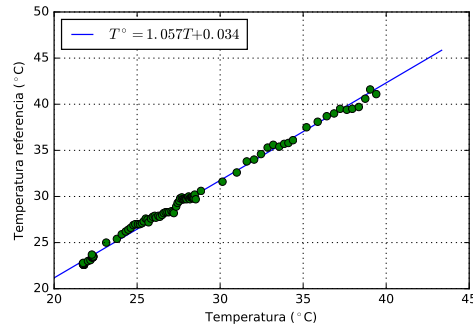


Figura 1.2 Calibración del sensor.

vatos. Para una resistencia la potencia es proporcional al cuadrado de la diferencia de potencial entre sus terminales:

$$P = VI = V \left(\frac{V}{R} \right) \quad (1.1)$$

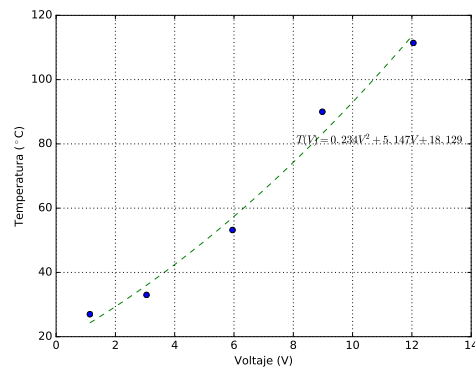


Figura 1.3 Temperaturas alcanzadas por la resistencia para distintos valores de voltaje.

Debido a que el sistema se encuentra restringido a voltajes de ± 12 y 0 V, la máxima potencia sobre la resistencia será 5.3 W, valor que se encuentra dentro de la capacidad de la resistencia.

1.3 Sistema de alimentación

El sistema de alimentación está compuesto por dos reguladores de voltaje de 5 V de la familia 7805 [2]. El circuito de alimentación se encuentra acompañado de dos

condensadores que actúan como filtros, además se dispone de tres diodos 1N4001 [3] como protección a corriente reversa. El circuito sin carga mantiene un potencial de 5.03 V, y con carga se alcanzan valores de 4.78 V.

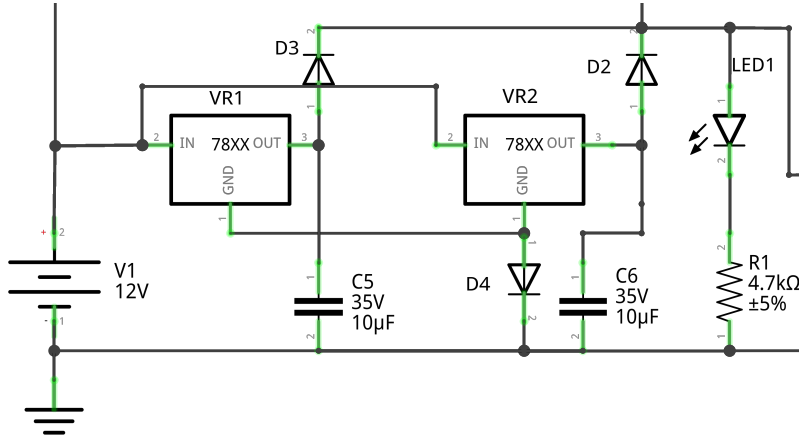


Figura 1.4 Circuito de alimentación del horno.

1.4 Controlador

El controlador está compuesto por tres etapas:

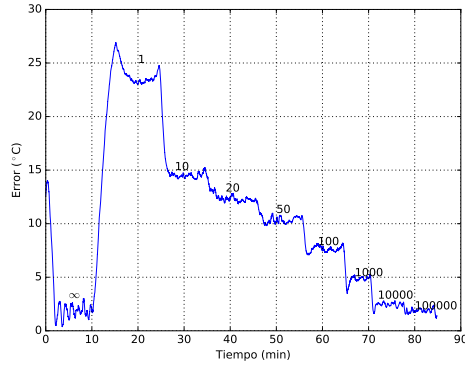
- **Restador:** calcula el error en la temperatura del horno.
- **PID:** todas las partes invierten la salida y mantienen la ganancia en 1.
- **Sumador:** se suma de forma ponderada las distintas componentes del PID. Las constantes usadas se muestran en la Tabla 1.1.

Tabla 1.1 Parámetros del controlador PID.

K_P	100
K_I	10
K_D	1.42

Los parámetros del controlador fueron determinados manualmente. En primer lugar se determinó K_P usando 8 valores distintos de R_{13} .

$$K_P = \frac{R_{14}}{R_{13}} \quad (1.2)$$



En segundo lugar se agrega la etapa derivativa, para la cual la ganancia K_D corresponde con;

$$K_D = \frac{R_{14}}{R_{15}} \quad (1.3)$$

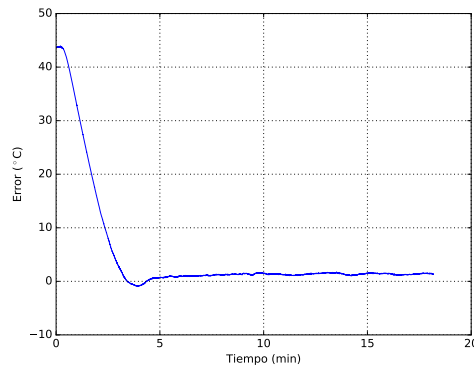


Figura 1.5 Resultado final del controlador PID.

Finalmente se agrega una etapa integradora para eliminar el error del estado estacionario, los valores de K_I con los que se probó el circuito fueron: 10, 1 y 1.42. El primero, satura permanentemente el actuador, manteniendo siempre al máximo la diferencia de potencial. Con el segundo se alcanza un error del estado estacionario para una temperatura de 67.6 °C en 3 °C, error que se reduce con la última constante a cerca de 1 °C.

1.5 Digitalización

El valor de referencia es fijado usando un potenciómetro digital con referencia X9C103P [4] con resistencia interna de 100 k Ω . Los tres pines lógicos son conectados a los GPIO, los puertos predeterminados son:

19 **IN** (INCREMENT)

20 **U/D** (UP/DOWN)

21 **C/S** (CHIP SELECT)

Sin embargo los mismos pueden ser modificados usando la interfáz gráfica. A continuación se muestran dos curvas del potencial en función de la posición de la aguja del potenciómetro.

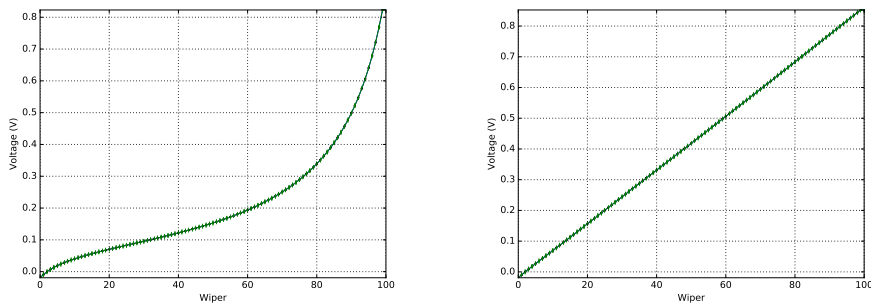


Figura 1.6 Voltaje de salida del potenciómetro acoplado al restador del PID. A la izquierda con carga, en el derecho con seguidor.

En las figuras anteriores se puede apreciar el efecto de la carga sobre el divisor de voltaje. Una modificación final en la etapa de digitalización permitió lograr un rango de potenciales de 0 a 3.3 V, conservando la linealidad, pero reduciendo la selección de temperaturas a cada 3.3 °C.

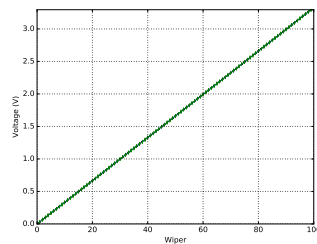


Figura 1.7 Calibración final del potenciómetro digital.

REFERENCIAS

1. Texas Instruments, “LM35 Precision Centigrade Temperature Sensors” LM35 datasheet, Aug. 1999 [Revised Aug. 2016].
2. Spark fun “Positive-Voltage Regulators” μ A7805C datasheet, May. 1976 [Revised May. 2003].
3. Diodes, “Diffused Junction 1N4001 - 1N4007” 1N4001 datasheet, Sep. 2004.
4. Digitally Controlled Potentiometer “X9C103” datasheet, Jul 2009.

CAPÍTULO 2

ALGORITMO

La parte computacional del proyecto está escrita en su totalidad en Python y usa una interfaz gráfica en Qt. Usando un paradigma de programación por objetos fueron creadas 5 clases que siguen la jerarquía que se muestra en la Figura 2.1. Las clases se encuentran divididas en tres archivos:

- `peltier.py`: Contiene la información relacionada con la interfaz gráfica, configura las señales y puertos necesarios por los objetos propios de Qt.
- `TCP.py`: Contiene las clases `client` y `server` que funcionan como cliente y servidor TCP (Protocolo de Control de Transmisión) los cuales permanentemente intercambian los datos de temperatura deseada, actual y referencia. El uso de un servidor TCP permite interactuar con el horno a través de una red LAN.
- `controller.py`: Contiene las clases `controller` y `potentiometer`, la primera determina los puertos lógicos con los que la Raspberry controla el potenciómetro digital, además, determina la posición de la aguja del potenciómetro para la cual se obtiene el voltaje más cercano a la temperatura deseada. La clase `potentiometer` está en capacidad de modificar la posición de la aguja del potenciómetro cada vez que `controller` lo considere necesario.

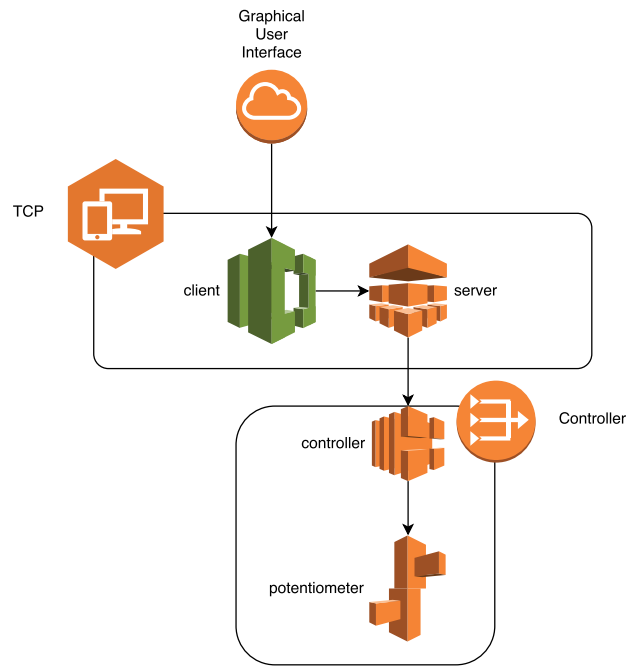


Figura 2.1 Jerarquía de las clases usadas por la interfaz gráfica.

Los códigos del proyecto así como los datos obtenidos para las curvas de calibración se encuentran publicados en GitHub, y hacen parte del repositorio `supreme-pi` como parte de un proyecto de *physical computing* [1].

REFERENCIAS

1. B. Juan, *supreme-pi*, GitHub repository, <https://github.com/jsbarbosa/supreme-pi>