

Implementación de Controlador PID de Temperatura Usando una SBC RedPitaya.

Sebastian Morelli,^{*} Daniel Ardila,^{**} Supervisora: Mayerlin Nuñez Portela,^{***}
and Analista de Laboratorio: Johnny Alberto Tenorio Albañil^{****}

Universidad de los Andes, Bogotá, Colombia.

(Dated: 2 de junio de 2021)

La plataforma Red Pitaya es un sistema integrado de código abierto diseñado para propósitos de instrumentación y medición. Para el siguiente experimento se utilizó esta plataforma con el fin de implementar un controlador PID de temperatura basándose en el mismo principio que un termostato. Con la ayuda de un relé, un sensor de temperatura LM35, una celda peltier, una fuente de voltaje de 12V y una Red Pitaya se logró realizar el controlador de temperatura. La precisión máxima alcanzada para este controlador fue de $\pm 0,5^{\circ}C$ y un tiempo de respuesta máximo de 0.1 s.

I. INTRODUCCIÓN

En los diferentes campos en los cuales la física experimental puede aplicarse, muchas veces es necesario realizar un control preciso de las condiciones del laboratorio, de forma que, los experimentos se desarrollen de manera óptima y se obtengan los resultados deseados. Dentro de dichas condiciones que pueden ser necesarias controlar, se encuentran variables físicas medibles como el voltaje, la presión, la temperatura, el ángulo de inclinación, entre otros. Estas variables, pueden influir en los resultados de un experimento, especialmente, si se buscan hacer mediciones en las que sea necesario mantener constantes dichas variables en el sistema, a lo largo de una toma de datos. En específico, una de las variables más importantes a controlar experimentalmente, es la temperatura, debido a que esta puede influir de gran manera en sistemas que impliquen ciencia de materiales, detectores de partículas o radiación y sistemas con procesos biofísicos [1].

Para lograr controlar la temperatura en ambientes experimentales existen múltiples métodos, entre los más usados, se encuentran los controladores de temperatura industriales que mantienen una temperatura estable de manera muy precisa; sin embargo, estos controladores son altamente costosos y, en muchas ocasiones, complicados de conseguir con las especificaciones necesarias para adecuarlos a un montaje de laboratorio específico. Con esto en mente, vale la pena preguntarse si existe la forma de implementar un controlador de temperatura con materiales menos costosos, de fácil implementación y, con una precisión similar a la de dichos controladores industriales [1]. Una posible respuesta a esta pregunta aparece cuando se observan los controladores Proporcionales Integrales Derivativos (PID), los cuales bajo una

implementación sencilla de código en una Single Board Computer (SBC), pueden controlar variables de laboratorio de manera precisa si son bien implementados [2].

De manera específica, se intentará evaluar si dicha pregunta se puede responder implementando un controlador PID de temperatura en un SBC RedPitaya como objetivo general. En donde los objetivos específicos para lograrlo, serán:

- Estudiar el funcionamiento del procesador RedPitaya y encontrar la forma óptima de programarlo.
- Implementar un controlador PID de temperatura con un precisión de al menos a $0.1^{\circ}C$ y un tiempo de respuesta de al menos 0.1 segundos.
- Utilizar el controlador de temperatura implementado en un horno peltier de laboratorio que permita evaluar su eficacia.

II. MARCO TEÓRICO

Inicialmente, para poder realizar este proyecto, existe la necesidad de comprender lo que significa un controlador PID. Básicamente, el significado de sus siglas: Proporcional, Integral y Derivativo, proporcionan la idea de sus funcionamiento, ya que estos son los tres procesos que este tipo de controladores realizan para mantener estable una variable determinada. Un esquema útil para entender el proceso de un controlador PID, se encuentra en la figura 1.

Este esquema muestra una señal o valor de referencia $r(t)$ la cual es comparada con un señal $h(t)$ que proporciona un sensor midiendo la variable de interés en el sistema. De esta comparación se proporciona una señal de error $e(t)$, que proporciona la información de la diferencia que existe entre el valor que se quiere $r(t)$ y el que hay en el sistema $h(t)$. Después, esta señal de error es corregida por los procesos proporcionales, integradores y derivadores (que serán explicados más adelante) y proporciona una señal de control $c(t)$ resultado de las

^{*} sm.morelli@uniandes.edu.co

^{**} da.ardila10@uniandes.edu.co

^{***} m.nunez@uniandes.edu.co

^{****} ja.tenorio@uniandes.edu.co

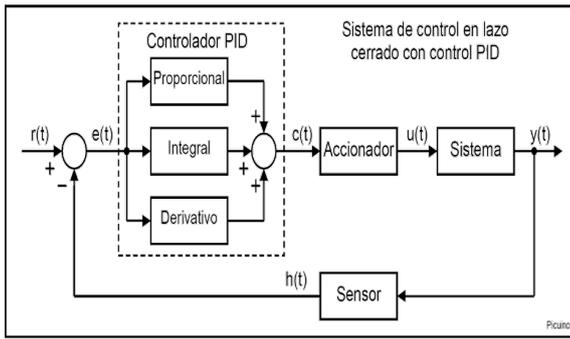


Figura 1. Esquema lógico de un controlador PID. Se reciben valores de referencia $r(t)$, de la variable medida $h(t)$, se proporciona el error $e(t)$, se hace el procedimiento PID y se envía un señal de control $c(t)$, que hace que el accionador afecte el sistema con una señal $u(t)$, para que finalmente el sistema cambie dicha variable y tenga un valor de $y(t)$, y el proceso se repite. Figura tomada de [2].

correcciones hechas por cada proceso. Finalmente, dicha señal de control $c(t)$ genera una señal en el accionador $u(t)$, la cual interviene en el sistema realizando la corrección necesaria para obtener nuestro valor de referencia $r(t)$ deseado. A partir de lo anterior, el sistema tendrá un valor $y(t)$ nuevo en nuestra variable controlada, que será nuevamente medida por el sensor, repitiendo todo el proceso con tal de mantener la variable estable para cualquier tiempo t [2].

Ahora, este tipo de controlador implementado a la variable de temperatura de un sistema, representa que la señal de referencia $r(t)$ sea el valor de temperatura al cual se quiere mantener constante el sistema, la señal $h(t)$ sería la temperatura del sistema medida por un sensor y la señal de error $e(t)$ sería la diferencia de temperatura al comparar estas dos últimas. Una vez hecha esta comparación, se realizan los tres procesos PID y se proporciona una señal de control $c(t)$ la cual hace que el accionador caliente o enfríe el sistema a través de una señal $u(t)$, y el sistema adquiere una temperatura $y(t)$ que será medida y el proceso se repetirá nuevamente [1, 2]. Teniendo en mente lo anterior, ahora es necesario especificar qué serían los procesos PID en este sistema, para ello es útil utilizar la figura 2. En esta figura, se muestra una gráfica de Temperatura en función del tiempo para un sistema al cual se trata de mantener a una temperatura estable.

Partiendo de la figura 2, podemos ver que se tiene un umbral o setting preestablecido, al cual se quiere mantener la temperatura constante, que siguiendo la notación anterior sería nuestra señal $r(t)$; a partir de este umbral, podemos ver cómo se apaga o se prende un calentador, que sería nuestro accionador, según sea necesario dejar enfriar o calentar el sistema. Este proceso, hace que el método de control proporcional, tome un dato $T = y(t)$

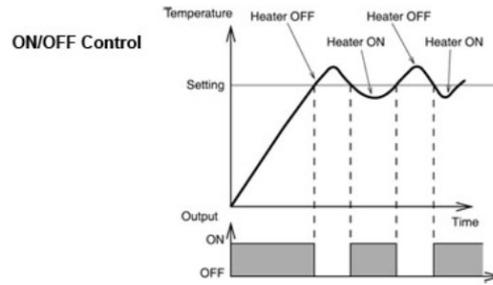


Figura 2. Esquema de un controlador PID implementado a la variable temperatura. Se muestra una gráfica de temperatura en función del tiempo. Se tiene un umbral al cual se quiere mantener constante la temperatura y se tiene cuando se prende o se apaga un calentador para mantener dicha temperatura cercana al umbral. Figura tomada de [3].

(siendo T temperatura y t tiempo) en un tiempo determinado y haga la operación

$$e(t) = y(t) - T_0; \quad (1)$$

siendo $e(t)$ la señal de error ya explicada y T_0 la temperatura umbral. A partir de esto, se genera una señal de control proporcional $c_p(t)$ que le dirá al accionador que varíe la temperatura una cantidad $e(t)$. De manera similar, el método de control integrador se encara de hallar el área bajo la curva hasta el umbral, de forma que lo que se busca es que dicha área sea mínima, es decir, el punto este más cerca del umbral; adicionalmente, esta medición da información de que tanto debe ser la corrección proporcional, pues entre mayor sea el área mayor debes ser esta y viceversa, donde medición genera una señal de control integral $c_i(t)$. Finalmente, el método de corrección derivativo se basa en realizar la derivada, es decir la recta tangente a un punto de la curva, por ende, se busca que dicha derivada sea lo más cercana a cero posible, brindando así una señal de control derivativa $c_d(t)$; adicionalmente, esta medición aporta información sobre la tendencia futura de la curva, lo cual a su vez permite regular el ajuste proporcional [1, 2, 4].

Teniendo todo lo anterior en mente, podemos decir que la ecuación que dicta el valor de la señal de control será:

$$c(t) = c_p(t) + c_i(t) + c_d(t); \quad (2)$$

o equivalentemente, haciendo explicitas sus funciones:

$$c(t) = e(t) + \int e(t)dt + \frac{\partial e(t)}{\partial t}. \quad (3)$$

Aún así, debido a que las señales experimentales no son funciones sino datos, tanto las derivadas como las

integrales deben encontrarse por definición, de manera que podemos definir:

$$\int e(t)dt = \frac{(y(t + \Delta t) - T_0) + (y(t) - T_0)}{2} \Delta t; \quad (4)$$

$$\frac{\partial e(t)}{\partial t} = \frac{(y(t + \Delta t) - T_0) - (y(t) - T_0)}{\Delta t}. \quad (5)$$

Con esto tenemos finalmente una ecuación la cual podemos implementar en código para nuestro controlador PID de temperatura [1, 2, 4].

III. MARCO EXPERIMENTAL

En el ámbito de la teoría necesaria para la experimentación, es importante entender qué es un SBC, y por qué se escogió un SBC RedPitaya para el presente proyecto. En principio, las siglas de SBC significan Computador de Placa Única por sus siglas en inglés, lo cual básicamente significa que este tipo de instrumentos son una unidad de procesamiento con todas sus componentes en una única placa o tarjeta. Este tipo de instrumentos, son de alta utilidad debido a su fácil programación y su versatilidad para cualquier tipo de tarea que requiera el procesamiento de datos o su computación [3]. Ahora, la tarjeta RedPitaya es una SBC de nueva generación que presenta amplias ventajas de procesamiento frente a sus semejantes Arduino y RaspberryPi, pero a su vez es mucho menos costosa que los controladores de temperatura industriales. Es válido a su vez hacer la aclaración, de que la RedPitaya también es más costosa que Arduino y RaspberryPi; sin embargo, estas dos últimas no poseen la capacidad de procesamiento necesaria para aportar la precisión de control requerida en un experimento de laboratorio avanzado. La tarjeta física, arquitectura y modelo de la RedPitaya es visible en la figura 3 [3].

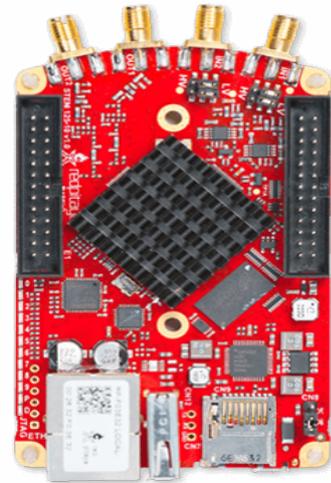


Figura 3. Foto real de una tarjeta RedPitaya, mostrando su arquitectura y componentes. Figura tomada de [3].

Teniendo este sistema de procesamiento, fue necesario entonces adaptar a este, en una de sus entradas de adquisición de datos (input), un sensor de temperatura, el sensor de temperatura utilizado fue un LM35, el cual provee una adquisición de datos suficiente para este experimento. Adicionalmente, en una de sus salidas de datos (output), se adecuó un relé RH-012C, el cual a su vez iba conectado a una celda peltier TEC1-12706. Las funciones de cada uno de estos componentes son entonces, en el caso de LM35, medir y enviar la información de la temperatura al RedPitaya; este sensor de temperatura funciona recibiendo un voltaje de parte del RedPitaya, en donde el sensor la deja pasar actuando como una resistencia con la equivalencia $R_{LM35} = \frac{1}{T}$ siendo R_{LM35} es el valor de la resistencia y T sigue siendo la temperatura. Con esto, cuando el voltaje vuelve a la RedPitaya, se puede saber cuanta resistencia tenía representaba el sensor y equivalentemente qué temperatura tenía, lo cual es la información de interés.

Después, el RedPitaya se encarga de realizar todo el procedimiento PID ya explicado, ejecutando un código escrito en Python 3.0 y corrido en un JupyterNotebook, para finalmente enviar una señal de voltaje hacia el relé a través de su output. El relé RH-012C, fue indispensable ponerlo debido a que el voltaje máximo que enviaba era de $V = \pm 6[V]$, lo cual no era suficiente para calentar ni enfriar la celda peltier como se requería, por ende el relé cumplía dos funciones, la primera, al estar conectado a una fuente de 12 voltios, era amplificar la señal según fuera necesario y segundo, apagaba y prendía el flujo de voltaje hacia el peltier, haciendo más eficaz el control al no permitir enviar el voltaje durante más tiempo del necesario, apoyado en la información provista por los modelos de control integral y derivativo.

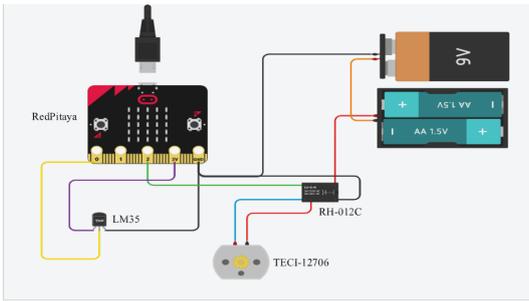


Figura 5. Esquema de la configuración de circuitos utilizados para un mejor entendimiento de la figura 4. Se utilizó la herramienta TinkerCad para realizar el esquema, el código representa que se tienen voltajes diferentes para cada color, como se explica en esta sección.

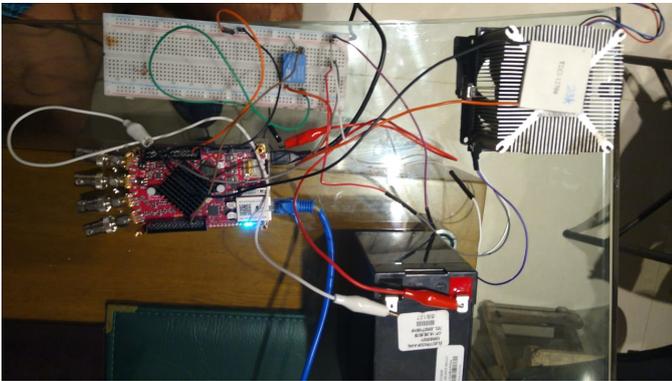


Figura 4. Montaje real del experimento realizado, se pueden observar todas las componentes necesarias y un protoboard que ayuda a conectar los circuitos.

Finalmente, el peltier recibe dicha señal de voltaje y la interpreta de modo que al recibir un voltaje de polaridad negativa tiene la capacidad de enfriarse, y uno positivo de calentarse, en ambos casos, proporcionalmente a la magnitud de dicho voltaje. Con esto en mente, se montó el circuito mostrado en la figura 4, que para entenderlo mejor, presentamos su esquema en la figura 5. Adicionalmente, las fichas técnicas de cada uno de los componentes mencionados se pueden ver en las referencias [5-8].

IV. RESULTADOS

Para comprobar que se podía mantener una temperatura estable para cualquier rango se utilizaron tres valores diferentes deseados con el fin de evaluar la precisión y la eficiencia del circuito. Se recuerda que la celda peltier tiene dos lados: uno sube la temperatura y el otro la disminuye. Para saber si se quiere calentar o enfriar un sistema se debe tener en cuenta

la temperatura ambiente. En el caso de los termistores se requiere una calibración que está dada por una relación entre resistencia y temperatura, para los chips de salida analógica esta calibración no es requerida. El sistema para el cual se quiso probar fue un disipador de calor, esto se debe a que su función principal es la conducción de calor y si se logra mantener este a una temperatura estable entonces cualquier otro sistema cuya conducción de calor no sea tan óptima puede manejarse de fácilmente.

Para cada valor deseado de temperatura se tomaron 60 muestras temperatura. Dependiendo si la temperatura deseada era mayor o menor a la temperatura ambiente entonces la celda peltier se colocaba ya fuese para calentar o enfriar el sistema. Todos los siguientes datos se tomaron en un lugar con temperatura ambiente aproximadamente de 26°C .

El primer valor de temperatura deseada en el que se probó fue de -2°C :

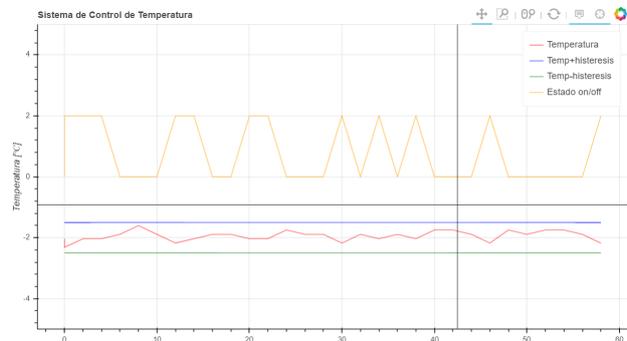


Figura 6. Gráfica de temperatura en función del tiempo en segundos para -2°C . La línea amarilla corresponde al estado del sistema de refrigeración, cuando se encuentra en un valor de 0 la celda se encuentra encendida y enfriando, cuando el valor cambia a 2 la celda se encuentra apagada y el sistema se comienza a calentar. La línea roja corresponde a la temperatura medida por el sensor LM35 y, la línea azul y verde corresponden al rango permisible de temperatura.

Luego, se probó para una temperatura igual a 15°C :

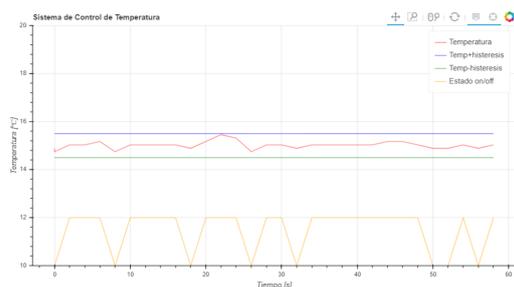


Figura 7. Gráfica de temperatura en función del tiempo para $15^{\circ}C$. La línea amarilla corresponde al estado del sistema de refrigeración, cuando se encuentra en un valor de 10 la celda se encuentra encendida y enfriando, cuando el valor cambia a 12 la celda se encuentra apagada y el sistema se comienza a calentar. La línea roja corresponde a la temperatura medida por el sensor LM35 y, la línea azul y verde corresponden al rango permisible de temperatura.

Por último se invirtió la polaridad de la conexión de la celda peltier para que la parte que está tocando el dissipador comenzará a calentar. La temperatura deseada en este caso fue de $40^{\circ}C$

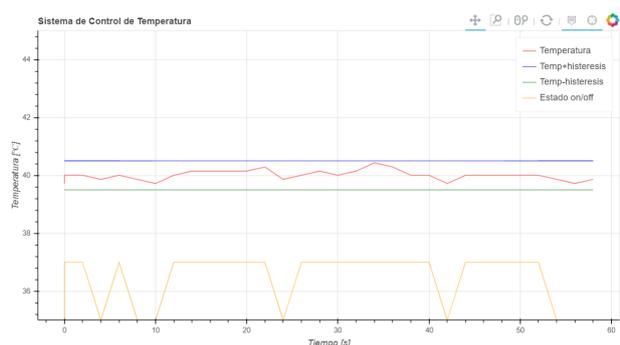


Figura 8. Gráfica de temperatura en función del tiempo para $40^{\circ}C$. La línea amarilla corresponde al estado del sistema de refrigeración, cuando se encuentra en un valor de 35 la celda se encuentra apagada y enfriando, cuando el valor cambia a 37 la celda se encuentra encendida y el sistema se comienza a calentar. La línea roja corresponde a la temperatura medida por el sensor LM35 y, la línea azul y verde corresponden al rango permisible de temperatura.

El promedio del tiempo de respuesta del circuito para cada uno de los casos se encontró que fue aproximadamente de $0,1 \pm 0,05s$. La histéresis térmica que se

logró alcanzar fue de $\pm 0,5^{\circ}C$, ésta nos define las cotas para la temperatura deseada.

V. DISCUSIÓN

La histéresis térmica se puede mejorar si se considera un sensor de temperatura con mayor precisión como los termistores. Debido a que el LM35 posee una precisión de calibración de $1^{\circ}C$ esto disminuye la precisión en las mediciones[6]. Otros factor que repercute en los errores de las mediciones incluye la variación de la temperatura ambiente. Para el caso de temperaturas tan bajas se debe tener en cuenta que el lugar en donde se realizaron las mediciones fue en la ciudad de Villavicencio y, por tal motivo, mantener una temperatura de $-2^{\circ}C$ es una tarea más difícil que si fuese en Bogotá. De igual manera, las otras dos temperaturas utilizadas, fueron todas con al menos $10^{\circ}C$ grados de diferencia de la temperatura ambiente.

Finalmente, resaltamos que el uso de un sensor de temperatura LM35 pudo afectar notoriamente la precisión obtenida, esto debido a que este es un sensor de uso cotidiano que si bien es muy útil y puede medir tanto altas como bajas temperaturas su precisión no es la mejor, por ende, si bien el RedPitaya tiene la capacidad de manejar aún más datos por segundo, adquirir mayor precisión también dependerá del sensor utilizado, por lo cual creemos que con instrumentos de laboratorio de medición de temperatura más avanzados, los objetivos se pueden alcanzar con facilidad.

VI. CONCLUSIONES

Al estudiar el funcionamiento del procesador de la Red Pitaya se encontró la forma de elaborar un controlador de temperatura con la ayuda del lenguaje de programación de Python. Esto se logró alcanzando una histéresis térmica de $0,5^{\circ}C$ y un tiempo de respuesta aproximadamente de $0,1 \pm 0,05s$. Por cuestiones actuales relacionadas con pandemia el controlador no se logró probar en un laboratorio de la Universidad de los Andes. A excepción de la Red Pitaya todos los equipos electrónicos usados fueron componentes caseros, así que hasta cierto grado permitido se logró cuantificar la eficacia del sistema.

[1] A Raghav, SureshKumar Rajendran, Prathap B Shetty, B R Kirana, Pinjala Damaruganath, D Shashanka, Shama Bhat, and Arun Chandrashekar. A simple and intelligent nonlinear pid temperature control with ambient

temperature feedback for nonlinear systems. In *2013 International conference on Circuits, Controls and Communications (CCUBE)*, pages 1–6, 2013.

- [2] Picuino. Controlador pid - control automático, 2021. url: <https://www.picuino.com/es/arduprog/control-pid.html>, Accedido: Abr. 23, 2021.
- [3] RedPitaya. Stemplab swiss army knife for engineers, 2021. url: <https://www.redpitaya.com/Index>, Accedido: Abr. 23, 2021.
- [4] J. Mozaryn, Jakub Petryszyn, and S. Ozana. Plc based fractional-order pid temperature control in pipeline: design procedure and experimental evaluation. *Meccanica*, 2020.
- [5] RedPitaya. Redpitaya documentation, 2021. url: <https://redpitaya.readthedocs.io/en/latest/>, Accedido: Jun. 1, 2021.
- [6] Texas Instruments. Lm35 precision centigrade temperature sensors, 2021. url: <https://www.ti.com/lit/ds/symlink/lm35.pdf/>, Accedido: Jun. 1, 2021.
- [7] Datasheet Archive. Rh -012c relay, 2021. url: <https://www.datasheetarchive.com/RH%20-012C%20RELAY-datasheet.html>, Accedido: Jun. 1, 2021.
- [8] All Data Sheet. Tec1-12706 datasheet (pdf) - hb electronic components, 2021. url: <https://www.alldatasheet.com/datasheet-pdf/pdf/313841/HB/TEC1-12706.html>, Accedido: Jun. 1, 2021.