

## IMPLEMENTACION DE CONTROLADOR PID EN UN INVERNADERO CASERO CON SISTEMA DE MONITOREO

Ernesto Corona Sánchez, Jesús Namigtle Jiménez, Jorge Edmundo Mastache Mastache, Mariela Enríquez Guadarrama, Alfredo Namigtle Jiménez, Oscar Bautista Merino.

<sup>a</sup> Universidad de Ixtlahuaca CUI, 50740, ernesto.corona@uicui.edu.mx, Zinacantepec, México, Estado de México, 51356.

<sup>b</sup> Jesús Namigtle Jiménez, Tecnológico de Estudios Superiores de Jocotitlán, Universidad de Ixtlahuaca CUI, México, namigtle@gmail.com <sup>c</sup> Jorge Edmundo Mastache Mastache, Tecnológico de Estudios Superiores de Jocotitlán, Universidad de Ixtlahuaca CUI, Jje\_masx\_masx@hotmail.com <sup>d</sup> Mariela Enríquez Guadarrama, Universidad de Ixtlahuaca CUI, mariela.enriquez@uicui.edu.mx <sup>e</sup> Alfredo Namigtle Jiménez, Universidad Tecnológica de Tehuacán, Instituto Nacional de Astrofísica, Óptica y Electrónica, Alfredo.namigtle@inaoep.mx <sup>f</sup> Oscar Bautista Merino, Universidad Tecnológica de Tehuacán.

### Resumen

En los últimos años se ha logrado implementar controladores PID de temperatura con sistemas PLC, tarjetas programables como por ejemplo Arduino e incluso utilizando otros microcontroladores con la ayuda de distintos lenguajes de programación. En el presente trabajo se propone un controlador PID programado en Python montado en *Raspberry Pi*, el cual es complementado con el desarrollo de interfaces gráficas formando así un sistema de monitoreo aprovechando las ventajas que este lenguaje ofrece. *GrovePi+* tiene un gran peso en esta propuesta ya que, al tener disponibles puertos analógicos, digitales, de comunicación serial y otros más, es una gran extensión de *Raspberry Pi* que permite agregar módulos y circuitos de control que favorezcan al funcionamiento del controlador. En dicho sistema de monitoreo se agregan herramientas tales como gráficas 2D que permiten la visualización de otras variables físicas como la humedad de suelo y el nivel de luz dentro del invernadero, grabación de datos de temperatura y una interfaz del control *PID* con el que el usuario tenga la posibilidad de manipular la temperatura dependiendo de las necesidades de la planta. Teniendo en cuenta que para el controlador *PID* se requiere de una etapa en la que un actuador este ligado al controlador se proponen circuitos adaptados tanto a *Raspberry Pi* como a *GrovePi+* y que de esa manera se tenga una mayor manipulación del controlador de la mano con el sistema de monitoreo.

**Palabras clave**— *GrovePi+*, Interfaces, PID, Python, *Raspberry Pi*.

### Abstract

*In recent years it has been possible to implement PID temperature controllers with PLC systems, programmable cards such as Arduino and even using other microcontrollers with the help of different programming languages. In this work we propose a PID controller programmed in Python*

*mounted on Raspberry Pi, which is complemented with the development of graphical interfaces thus forming a monitoring system taking advantage of the benefits offered by this language. Grove Pi+ has a great weight in this proposal because, having available analog, digital, serial communication ports and others, it is a great extension of Raspberry Pi that allows adding modules and control circuits that favor the operation of the controller. This monitoring system adds tools such as 2D graphs that allow the visualization of other physical variables such as soil moisture and light level inside the greenhouse, temperature data recording and a PID control interface with which the user has the ability to manipulate the temperature depending on the needs of the plant. Taking into account that the PID controller requires a stage in which an actuator is linked to the controller, we propose circuits adapted to both Raspberry Pi and GrovePi+ and thus have a greater manipulation of the controller hand in hand with the monitoring system.*

**Keywords**— *GrovePi+*, Interfaces, PID, Python, *Raspberry Pi*.

## 1. INTRODUCCIÓN

Durante los últimos años se han propuesto sistemas de monitoreo en invernaderos pequeños y medianos con distintos objetivos: supervisión de plagas y enfermedades, crecimiento óptimo de una planta mediante el seguimiento de la temperatura, la humedad de suelo, el riego, implementación de lógica difusa y IoT a través del uso de plataformas como entornos de desarrollo de aplicaciones Android, Arduino, *Raspberry Pi*, que han representado beneficios en el campo de la agricultura tradicional por la eficiencia y la optimización que llegan a alcanzar. Por lo que el tema de los invernaderos caseros ha tenido una repercusión positiva en el campo agrícola gracias a trabajos que se enfocan en la supervisión y el control en el desarrollo de [cultivos [10].

La evolución de la tecnología ha ofrecido una gran variedad de sensores, con mayor precisión a la hora de medir variables. En invernaderos se siguen conservando 3 en particular; la temperatura, la luz y la humedad de suelo. Para este proyecto se utilizarán los módulos *seed* (de la familia *Grove*) compatibles con el armazón *GrovePi+* disponible para *Raspberry Pi 3*. Estos módulos se han caracterizado por la facilidad de uso y es en mayor medida gracias a Python, ya que no es necesario de librerías particulares para cada sensor.

De las aportaciones que se han logrado implementar en el tiempo reciente, *Raspberry Pi* ha logrado adaptarse a la adición de módulos permitiendo el control de motores [2], propuestas para el control eficiente de riego [3] e incluso abarcando el área de la inteligencia artificial [4]. En otros casos, algunos sistemas incluyen características para detección de fallas, control de temperatura y monitoreo de enfermedades, lo que demuestra la efectividad de las reglas de producción como una tecnología compartida [5].

Por otro lado, hoy en día el lenguaje de programación Python ha tendido un auge creciente en el desarrollo de proyectos, es por ello que se tomará en cuenta para que trabaje en conjunto con *Raspberry Pi*. Al usar este lenguaje se abren las posibilidades de usar distintas herramientas aprovechando la variedad de módulos y por consecuencia un mejor tratamiento de los datos, *GrovePi+* al ser un complemento de *Raspberry Pi* ofrece un mejor manejo de los mismos con programas de apoyo básicos y una sintaxis que ahorra mucho el tiempo de desarrollo de programas (y el tiempo de ejecución también). Sin embargo, este armazón no es comercialmente explotado a pesar de las ventajas que ofrece. El compilador que posee se presta para poder realizar varias funciones desde la gestión de paquetes y librerías, visualización de herramientas y/o variables, etc.

Los filtros son una parte importante en sistemas de control con alta eficacia, en este proyecto se toma en cuenta y es por esta razón que se implementa un filtro de Kalman, este tipo de filtro digital ha repercutido en trabajos como; estimación de estados desconocidos [6], elaboración de interfaces en sensores resistivos inteligentes dirigidos a detectores de temperatura resistivos [8], etc. Además, se han realizado estudios acerca del impacto de estos filtros en controladores PID que demuestran ser de gran ayuda en sistemas y procesos de control robustos [7].

Otro de los elementos fundamentales es el controlador PID, el cual llega a ser útil en un sistema de control y es aplicable conociéndose o no el modelo matemático de planta. Este tipo de controlador es ampliamente utilizado en procesos industriales por el comportamiento que adopta dependiendo de las condiciones de operación [19]. Conformado por 3 elementos; ganancia proporcional, ganancia integral y ganancia derivativa es un tipo de controlador robusto que puede implementarse en la temperatura. Al igual que los trabajos citados anteriormente, la implementación de controladores PID y la implementación del lenguaje Python a demostrado ser posible bajo ciertas condiciones [18].

#### Planteamiento del problema

Los invernaderos caseros que no cuentan con supervisión y control adecuado, por lo general presentan deficiencias en el uso de recursos tanto de energía eléctrica como de agua, además, las posibles fluctuaciones de temperatura ponen en riesgo el crecimiento y salud de la planta y por consecuencia el cultivo corre el riesgo de perderse. Actualmente se pueden encontrar distintas formas para llevar un control automático en este tipo de invernaderos, tales como los sistemas scada, mediante análisis de imágenes, por el uso de plataformas inteligentes basadas en IoT, etc. Es importante decir que llevar a cabo estos sistemas de control resultan tener un alto costo por lo que se necesita una alternativa de menor costo que permita ser accesible para todo público y sobre todo que esté basado en tecnología open source.

#### Justificación

De acuerdo a los últimos avances en el diseño y automatización de invernaderos caseros se ha utilizado como herramienta principal alguna tarjeta de desarrollo, sin

embargo, *Raspberry Pi* gracias a su versatilidad con los sensores no se ha explorado en su totalidad en el campo del control y automatización. *Raspberry Pi* por si sola tiene limitantes que dificulta integrar sensores y la extensión *GrovePi+* sirve de cubrir esa desventaja al poseer puertos analógicos, digitales y de comunicación serial (I2C). Además, el sistema Operativo con el que trabaja es de código abierto haciéndolo utilizable para el público en general.

*Raspbian*, en su versión *Raspbian for Robots* forma parte de los sistemas operativos de código abierto presentando una de las ventajas más interesantes. Su robustez, estabilidad y rapidez, son ideales para servidores y aplicaciones distribuidas.

Otra de las virtudes de *Raspberry Pi* radica en su arquitectura; siendo un mini ordenador tiene la posibilidad de agregar programas o compiladores que hacen posible el uso del lenguaje de programación Python. Lenguaje que por su parte no se ha explotado en su totalidad y pese a eso ha colaborado en el desarrollo de proyectos tales como los sistemas de monitoreo, en el campo de la robótica, etc. Cabe recalcar que el sistema operativo *Raspbian for Robots* ya incluye un compilador (Thonny) por lo que solo es necesario instalar las librerías necesarias y comenzar a desarrollar.

#### Hipótesis.

A través del sistema de monitoreo implementado en *Raspberry Pi* en conjunto con *GrovePi+* se espera mantener la temperatura a un nivel deseado por medio del controlador PID propuesto, monitoreando en todo momento la humedad de suelo y la luminosidad, variables que influyen directamente en la salud de la planta.

#### Metodología

Se plantea utilizar la extensión *GrovePi+* disponible para *Raspberry Pi 3 B+* para desarrollar un sistema de monitoreo. Se utilizará el lenguaje de programación Python para realizar interfaces graficas además de herramientas que contribuyan a la implementación del controlador *PID* de temperatura.

El prototipo de prueba será construido con materiales de bajo costo con el propósito de brindar accesibilidad a la hora de integrar elementos de control.

El sistema de monitoreo estará conformado por una serie de interfaces graficas que provean al usuario información proveniente de los sensores, gráficos 2D para una mejor visualización, además de una herramienta que logre capturar información en hojas de cálculo para el procesamiento de los datos y posteriormente lograr encontrar los valores adecuados para el controlador de temperatura.

Los módulos y sensores que se utilizarán a lo largo del proyecto serán de la familia Seeed más que nada por la compatibilidad que tiene cada uno con la extensión *GrovePi+* (*Grove Temperature sensor* y *Grove Humidity sensor* por mencionar algunos ejemplos).

Se considera para la etapa de potencia un circuito detector de cruce por cero y otro que tenga la función de llevar el control de energía con la cual el actuador (foco cerámico) estará constantemente trabajando. Los circuitos serán desarrollados con el uso de Proteus por la versatilidad que tiene a la hora de armar la PCB.

## 2. CONTENIDO

### Diseño y construcción del invernadero.

Se diseña y construye un invernadero de 100x42.1x81.5 cm, la base estará conformada por madera, la estructura se conforma principalmente por tubos PVC de 19mm de diámetro y lona calibre 800. Dicho invernadero se construye con el fin de tener el espacio suficiente para adicionar los elementos de control que conforman el sistema de monitoreo. En la figura 1 se puede observar de manera general la estructura del invernadero.

Fig. 1. Prototipo.

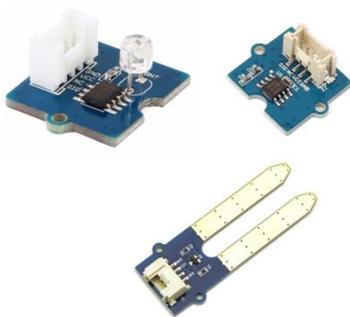


Fuente: Elaboración propia.

### Implementación de sensores de instrumentación.

Se implementarán sensores para el monitoreo constante de la temperatura, luz y humedad de suelo. La familia *Grove* provee sensores analógicos tales como; *Grove Light Sensor*, *Grove Moisture Sensor* y *Grove Temperature Sensor V1.2*. Cada uno de ellos tiene un rango de 0 a 1024, además, para poder obtener las lecturas se hará uso de instrucciones que a primera vista son muy similares a Arduino.

Fig. 2. Sensores de la familia *Grove*



Fuente: Elaboración propia.

### Adquisición de datos.

Es durante esta etapa que se utilizara la extensión *GrovePi+* utilizando los puertos analógicos. En el entorno de Thonny se llevará a cabo el procesamiento de datos de cada uno de ellos en conjunto de las interfaces graficas propuestas.

### Procesamiento de las lecturas.

El programa estará dividido en subprogramas; el script principal mostrara las lecturas que se están obtenido además de proveer al usuario un menú para poder acceder a las funciones del sistema de monitoreo.

La forma en que se podrá acceder a las funciones se representará mediante interfaces que dependen de la acción de los botones, tal como se puede observar en la figura 3. En la interfaz mostrada en dicha figura también se encuentran presentes medidas de luz y humedad de suelo, en este proyecto únicamente se estará presentando un controlador *PID* en relación a la temperatura, aunque también es necesario tener en cuenta otras variables para el sistema de monitoreo que influyen directamente en el desarrollo del cultivo.

Fig. 3. Interfaz principal del sistema de monitoreo.



Fuente: Elaboración propia.

En el sistema de monitoreo se contempla una herramienta desarrollada con Python cuya función es la de capturar datos de temperatura en las celdas en una hoja de cálculo de Excel. Ya que el usuario debe inicializar esta función requiere crear el archivo y para ello se contempla una interfaz gráfica.

Es gracias a esta función del sistema que se logra obtener información del estado de la temperatura que se encuentra dentro del invernadero. En la interfaz se encuentra un campo en el cual el usuario cada vez que deba iniciar una grabación de datos tiene que registrar el nombre del archivo, crearlo con el botón "Nombrar" e iniciar la captura.

Además, se pueden encontrar dos etiquetas; la de "temperatura" comenzara a actualizarse conforme se ejecute la herramienta. Mientras que la etiqueta "muestras" lleva el conteo del numero de datos que se guardan con respecto al tiempo.

Otra de las funciones que contiene el sistema de monitoreo son las graficas 2D que se revisara a continuación.

Para esta acción se requiere de la librería Matplotlib, ya que contiene instancias, métodos y elementos que agilizan la

creación de gráficos de distintos tipos según las necesidades del programador.

El muestreo es de 5 segundos, aunque fácilmente el tiempo podría ser mas corto, sin embargo, se debe tener en cuenta que el abrir otras funciones del sistema de monitoreo la actualización de los datos será más lenta.

La interfaz del controlador *PID* está diseñada para que funcione desde el proceso de sintonización de valores (valor proporcional, integral y derivativo) así como también de el accionamiento de la etapa de potencia para el actuador.

En la interfaz se puede visualizar los distintos campos que el usuario debe llenar para comenzar con la acción del controlador, primeramente, el valor de referencia (o valor deseado), valor proporcional, seguido por el integral y finalizando por el derivativo. Cada campo admite valores de tipo flotante de máximo 2 decimales y con restricciones para caracteres o de tipo especial.

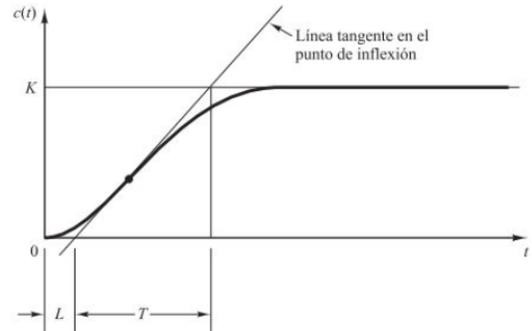
El apartado de parámetros se comenzará a actualizar una vez se guarden los valores haciendo clic en sus respectivos botones para poder guardarlos. Los datos mas relevantes que se pueden obtener son la temperatura actual, el valor deseado, valor de la suma del controlador (rango de 0 1000), el valor de PWM y el error actual. Esta información le servira al usuario a entender el estado del controlador o entendido de otra forma ayudan a verificar el correcto funcionamiento del PID.

#### Diseño de un controlador PID.

Para poder llevar a cabo el desarrollo del controlador, se determinó usar el método de sintonización de señales de Ziegler Nichols con el fin de hallar los parámetros adecuados (control proporcional, integral y derivativo).

Se realizará el método 1 del procedimiento experimental de Ziegler Nichols al no conocer el modelo de la planta (Temperatura del invernadero). Dicho método consiste en obtener la respuesta del proceso mediante datos experimentales, en donde se dejará trabajando una lampará cerámica emisora de calor en un tiempo de 8 horas, el sensor *Grove Temperature Sensor V1.2* captura valores cada 2 segundos y al finalizar la grabación por medio de la herramienta de la interfaz encontrar el valor estable del sistema logrando obtener la curva de respuesta (sigmoideal). Posteriormente se traza una recta tangente en el punto de inflexión tal como se puede observar en la figura 4.

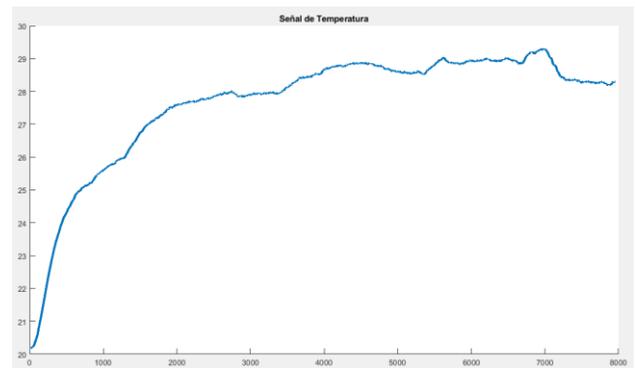
Fig. 4. Curva de respuesta del método 1.



Fuente: Ogata. K. 2003.

Conociendo el procedimiento, se analiza la grafica resultante de los datos capturados. La figura 5 muestra el comportamiento de una de las pruebas experimentales que se estuvieron realizando, cabe destacar que se uso Matlab para poder encontrar los puntos de la grafica que determinara la recta tangente.

Fig. 5. Señal resultante de la temperatura en el invernadero.



Fuente: Elaboración Propia

Al obtener la recta tangente a la grafica se realizaron los siguientes cálculos basándose en las expresiones matemáticas tal como lo describe el autor en [1].

$$K_p = 1.2 \frac{T}{L} \quad (1)$$

$$K_i = 2L \quad (2)$$

$$K_d = 0.5L \quad (3)$$

Por tanto, de la ecuación (1) se obtiene:

$$K_p = 1.2 \left( \frac{0.09996}{0.02926} \right) = 4.0995$$

De la ecuación (2):

$$K_i = 2L = 0.05852$$

Y finalmente de la ecuación (3):

$$K_d = 0.5L = 0.01465$$

Es de esta manera que se logran encontrar los valores de cada ganancia, una vez terminada la sintonización del controlador queda por definir la etapa de potencia para lograr un control completo de la temperatura.

En cuanto a la programación del controlador, se implementará la fórmula teórica del *PID* en una clase aparte para que, al cargarla al programa principal acepte los valores P, I y D determinados analíticamente, acote el valor de la suma y pueda generar un pulso PWM que ponga en funcionamiento el actuador (Generador de calor)

#### Diseño y Elaboración de la Etapa de Potencia.

En esta etapa se hará uso de un optoacoplador triac para que la señal PWM digital maneje el funcionamiento del actuador analógico (Fuente de calor).

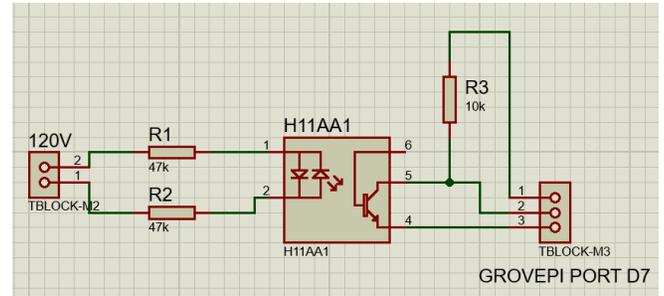
En la imagen de la figura 7 se puede observar el diseño del circuito el cual consiste en utilizar un optoacoplador moc3021 y un triac BT139 principalmente. Este permitirá utilizar la potencia del voltaje de una red de 120v de acuerdo con el funcionamiento del PWM.

Ya que *Raspberry Pi* en su modelo 3 B+ es incapaz de arrojar un pulso a una frecuencia de 120Hz por tanto se opta por utilizar una señal de reloj de 3.4V desde el pin 18 o también conocido como reloj PCM. Este pin permitira tener el control del circuito de potencia a una frecuencia de 120Hz mediante un pulso modulado, sin embargo, el circuito integrado MOC3021 funciona con tensiones de 3.3V en adelante y para valores bajos que provee la señal de reloj no es posible accionar el actuador, es por esta razón que se hará uso del amplificador operacional TL084. La configuración de sumador no inversor es adecuada para el amplificador y arrojará una tensión máxima de 7.02V.

Otro de los circuitos que formaran parte del sistema es el integrado H11AA1, que tendrá la función de sincronizar el pulso de reloj con la onda senoidal de la red de alimentación de 120V, es por esta razón que es tan importante que la señal de reloj funcione a 120Hz. La señal de reloj se verá afectada por el ancho de pulso determinado por el programa, por lo que, en cuanto el pulso llegue a 0V habrá un corte de energía en la señal de 120V.

En dicho circuito de cruce por cero se hace uso de resistencias de 47KΩ ya que el C.I contiene en su interior diodos y es fundamental cuidar que no excedan la corriente que circule por ellos, la salida se verá reflejada en forma de pulsos de 5V dirigidos a la *GrovePi+* en el puerto digital 7.

Fig. 6. Detector de cruce por cero.

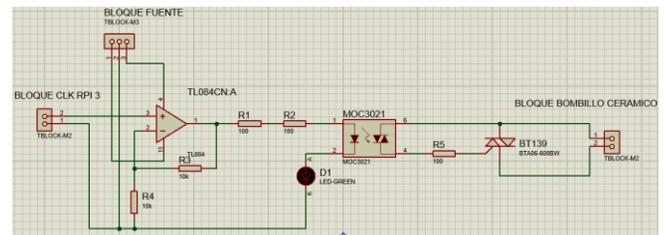


Fuente: Elaboración propia.

Al poner en funcionamiento el circuito de cruce por cero (Figura 6) se corroboran algunos de los datos mediante el uso del osciloscopio.

Con la descripción de los circuitos y los valores encontrados para el controlador, el sistema de monitoreo estará listo para realizar pruebas de control de la temperatura por medio del PID.

Fig. 7. Diseño de PCB para etapa de potencia.



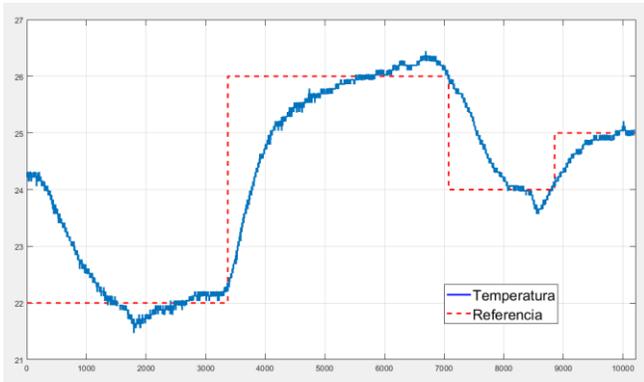
Fuente: Elaboración propia.

#### Resultados

Al realizar las interfaces gráficas en Python se pudo observar que, al ejecutar el sistema de monitoreo el rendimiento de *Raspberry Pi* no sobrepasa el 10% en el uso de los recursos, por lo que utilizar *GrovePi+* como extensión representa una gran ventaja.

En cuanto a la respuesta del controlador de temperatura se muestran las gráficas en las siguientes imágenes. Para la primera prueba se determinó llegar a 4 valores deseados; en un principio a 22° C, después a 26° C, pasar a 24° C y finalmente subir hasta los 25° C. Gracias a la figura 8 se observo que al momento en que el controlador debía llegar al valor deseado logro retener la temperatura de manera constante con un error máximo de 0.1 a 0.3 por encima del esperado (en el caso del valor de referencia de 25° C, se observó una temperatura real de 25.1° C por mencionar un ejemplo).

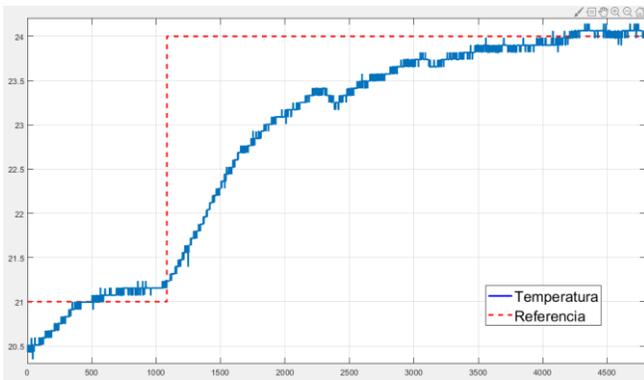
Fig. 8. Respuesta del controlador PID prueba 1.



Fuente: Elaboración propia.

Se realizó una segunda prueba en la que se debía mantener la temperatura a 21° C y después incrementarla a 24° C, la figura 9 muestra el comportamiento del controlador.

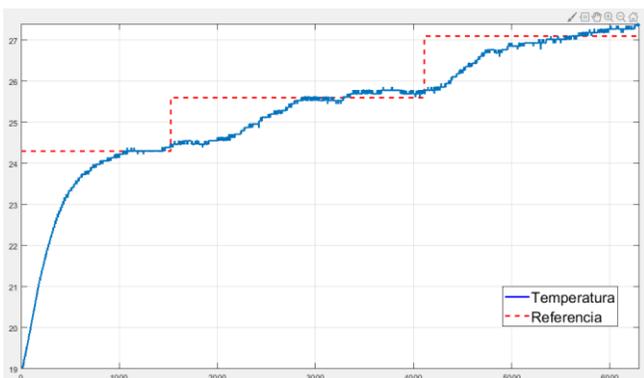
Fig. 9. Respuesta del controlador PID prueba 2.



Fuente: Elaboración propia.

Por último, se llevó a cabo una última prueba, en la que los valores de referencia determinados fueron; en primera instancia 24.3° C, seguido de elevar hasta los 25.6° C y terminar con una temperatura de 27.1° C. Tal como se observa en la figura 10.

Fig. 10. Comportamiento del controlador prueba 3.



Fuente: Elaboración propia.

Es por este método experimental que se corrobora el funcionamiento del procedimiento propuesto por Ziegler-Nichols.

### 3. CONCLUSIONES Y RECOMENDACIONES

La temperatura máxima que alcanzó el invernadero fue de 29° C, en base a este dato es posible albergar cultivos hortícolas en un rango aceptable para su crecimiento.

Es posible desarrollar un sistema de monitoreo en *Raspberry Pi* en conjunto con el complemento *GrovePi+* ya que en base con los resultados arrojados por las interfaces el rendimiento de la tarjeta programable no sobrepasa el 10% aun cuando el controlador está activo. Adicionar sensores al sistema sin que las lecturas arrojadas por cada uno alteren la ejecución del sistema resulta ser sencillo, esto si se emplea el uso de la misma familia de módulos y sensores, esto llega a ocurrir con sensores digitales de temperatura como lo es el DHT11.

En general, el error mostrado en cada una de las pruebas se encuentra en el rango de 0.1 y 0.3 grados centígrados, de acuerdo a esta estadística se puede decir que el controlador PID de temperatura logra alcanzar el valor deseado con muy poco margen de error.

#### 3.1 Observaciones generales

Especial agradecimiento a las instituciones que prestaron su apoyo y espacio en la elaboración de este proyecto; a la Universidad Tecnológica de Tehuacán, al TECNM/Tecnológico de estudios Superiores de Jocotitlán y al CUI de Ixtlahuaca.

### 4. REFERENCIAS

- [1] K. Ogata, *Ingeniería de control moderna*. Pearson Educación, 2003.
- [2] T. J. E. Andrés, "Sistema de control de software e interfaz de usuario para control de microclimas de cultivo con gateway *iot*," 2016. <http://repositorio.unab.cl/xmlui/handle/ria/3726>
- [3] V. Del Rocío Ochoa López *et al.*, "Propuesta de un sistema para optimizar el riego en invernaderos de plantas heterogéneas usando WNS y algoritmos evolutivos," *Research in Computing Science*, vol. 113, no. 1, pp. 131–145, Dec. 2016, doi: 10.13053/rcs-113-1-10.
- [4] Cañadas, J., Sánchez-Molina, J., Rodríguez, F., & Del Águila Cano, I. M. (2017). Improving automatic climate control with decision support techniques to minimize disease effects in greenhouse tomatoes. *Information Processing in Agriculture*, 4(1), 50-63. <https://doi.org/10.1016/j.inpa.2016.12.002>
- [5] J. Cañadas, J. A. Sánchez-Molina, F. Rodríguez, and J. Del Sagrado, "Improving automatic climate control with

decision support techniques to minimize disease effects in greenhouse tomatoes,” *Information Processing in Agriculture*, vol. 4, no. 1, pp. 50–63, Mar. 2017, doi: 10.1016/j.inpa.2016.12.002.

[6] M. Valipour, K. M. Toffolo, and L. Ricardez–Sandoval, “State estimation and sensor location for Entrained-Flow Gasification Systems using Kalman Filter,” *Control Engineering Practice*, vol. 108, p. 104702, Mar. 2021, doi: 10.1016/j.conengprac.2020.104702.

[7] S. Wakitani, H. Nakanishi, Y. Ashida, and T. Yamamoto, “Study on a Kalman Filter based PID Controller,” *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 422–425, Jan. 2018, doi: 10.1016/j.ifacol.2018.06.131.

[8] R. Anandanatarajan, M. Umopathy, and G. Uma, “Performance enhancement and fault identification using Kalman filter in a resistive temperature sensor interface,” *Measurement*, vol. 183, p. 109836, Oct. 2021, doi: 10.1016/j.measurement.2021.109836.

[9] A. E. Abioye *et al.*, “A data-driven Kalman filter-PID controller for fibrous capillary irrigation,” *Smart Agricultural Technology*, vol. 3, p. 100085, Feb. 2023, doi: 10.1016/j.atech.2022.100085.

[10] F. Jiang and X. Hu, “An IoT-based hierarchical control method for greenhouse seedling production,” *Procedia Computer Science*, vol. 192, pp. 1954–1963, Jan. 2021, doi: 10.1016/j.procs.2021.08.201.

[11] A. Kumar, V. Singh, S. D. Kumar, S. P. Jaiswal, and V. S. Bhadoria, “IoT enabled system to monitor and control greenhouse,” *Materials Today: Proceedings*, vol. 49, pp. 3137–3141, Jan. 2022, doi: 10.1016/j.matpr.2020.11.040.

[12] A. De Castro *et al.*, “BerryIP embedded: An embedded vision system for strawberry crop,” *Computers and Electronics in Agriculture*, vol. 173, p. 105354, Jun. 2020, doi: 10.1016/j.compag.2020.105354.

[13] M. Cobos, P. Otiñar, P. Magaña, A. Lira-Loarca, and A. Baquerizo, “MarineTools.temporal: A Python package to simulate Earth and environmental time series,” *Environmental Modelling and Software*, vol. 150, p. 105359, Apr. 2022, doi: 10.1016/j.envsoft.2022.105359.

[14] S. H. Park, H. S. Kim, C. Jhin, H.-J. Kim, and S. H. Park, “Time-serial analysis of deep neural network models for prediction of climatic conditions inside a greenhouse,” *Computers and Electronics in Agriculture*, vol. 173, p. 105402, Jun. 2020, doi: 10.1016/j.compag.2020.105402.

[15] P. Bhandari and P. Z. Csurcsia, “Digital implementation of the PID controller,” *Software Impacts*, vol. 13, p. 100306, Aug. 2022, doi: 10.1016/j.simpa.2022.100306.

[16] R. Bucher, “Practical experiences with Python and Linux RT at the SUPSI Laboratory,” *IFAC-PapersOnLine*, vol. 52, no. 9, pp. 133–138, Jan. 2019, doi: 10.1016/j.ifacol.2019.08.137.

[17] A. Vergnaud, J.-B. Fasquel, and L. Autrique, “Python based internet tools in control education,” *IFAC-PapersOnLine*, vol. 48, no. 29, pp. 43–48, Jan. 2015, doi: 10.1016/j.ifacol.2015.11.211.

[18] P. Gopi, M. Ramesh, and M. P. Lalitha, “Evaluation of Automatic Voltage Regulator’s PID Controller coefficients using Python,” *IEEE*, Aug. 2021, doi: 10.1109/mascon51689.2021.9563458.

[19] R. C. Dorf and R. H. Bishop, *Sistemas de control moderno*. PRENTICE HALL, 2005.