

CONESCAPANHONDURAS2025paper2.pdf

 Institute of Electrical and Electronics Engineers (IEEE)

Document Details

Submission ID

trn:oid:::14348:477779240

Submission Date

Jul 31, 2025, 11:58 PM CST

Download Date

Aug 1, 2025, 1:05 PM CST

File Name

CONESCAPANHONDURAS2025paper2.pdf

File Size

667.5 KB

6 Pages




3,500 Words

18,206 Characters

14% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Top Sources

- 14%  Internet sources
- 4%  Publications
- 0%  Submitted works (Student Papers)

Integrity Flags




0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Top Sources

- 14%  Internet sources
- 4%  Publications
- 0%  Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	es.scribd.com	2%
2	Internet	www.coursehero.com	1%
3	Internet	mafiadoc.com	1%
4	Internet	eprints.umm.ac.id	<1%
5	Internet	pdfs.semanticscholar.org	<1%
6	Internet	dspace.ups.edu.ec	<1%
7	Internet	dokumen.tips	<1%
8	Internet	core.ac.uk	<1%
9	Internet	repository.pedagogica.edu.co	<1%
10	Internet	vdocuments.com.br	<1%
11	Internet	dc.uwm.edu	<1%

12	Internet	ia800804.us.archive.org	<1%
13	Internet	repobib.ubiobio.cl	<1%
14	Internet	tesis.usat.edu.pe	<1%
15	Internet	autodocbox.com	<1%
16	Internet	edoc.pub	<1%
17	Internet	www.iq.coord.usb.ve	<1%
18	Internet	b-ok.org	<1%
19	Internet	dergipark.org.tr	<1%
20	Internet	engineering.stackexchange.com	<1%
21	Internet	www.agilent.com	<1%
22	Internet	ddd.uab.cat	<1%
23	Internet	investigacionescun.weebly.com	<1%
24	Internet	isa.uniovi.es	<1%
25	Internet	artistic.contitouch.co.zw	<1%

26	Internet	laexpediciondeportiva.blogspot.com	<1%
27	Internet	prezi.com	<1%
28	Internet	repository.udistrital.edu.co	<1%
29	Internet	www.jove.com	<1%
30	Publication	Mendez Perez, Juan Albino. "Desarrollo de Estrategias de Control Predictivas de A...	<1%
31	Internet	academica-e.unavarra.es	<1%

Diseño, implementación y validación de un módulo didáctico para enseñanza de control PID en tiempo discreto con plataforma embebida

Abstract—This document presents the design and construction concepts of a didactic and robust discrete-time PID module implemented in a development platform. The system aims to be an alternative to existing PID controllers, being easy to maintain and use for teachers and students in engineering-related areas.

Keywords—Proportional, integral, derivative, PID, discrete, time, didactic, arduino, module, development platform, *setpoint*.

INTRODUCCIÓN

El estudio de los controladores PID es fundamental en ingeniería electrónica y de automatización para el control de variables dinámicas con comportamientos temporales complejos y parámetros no estacionarios. Los estudiantes deben validar su comprensión mediante experimentación práctica, interactuando con sistemas reales para ajustar parámetros, analizar respuestas transitorias y evaluar estabilidad, para poder determinar los diferentes efectos que tiene modificar los valores de las constantes (proporcional, derivativa e integral); sin embargo, las limitaciones en los materiales de estudio hacen que no todas las instituciones educativas cuenten con sistemas PID lo suficientemente robustos y didácticos para que los estudiantes puedan interactuar. Por lo tanto, se plantea la construcción de un Módulo Didáctico de PID en Tiempo Discreto con materiales y diseño asequibles para cualquier institución, sin comprometer la calidad de las mediciones ni la robustez del sistema.

El controlador Proporcional-Integral-Derivativo (PID) es un sistema de retroalimentación utilizado en ingeniería de control para regular variables dinámicas. Su diseño se basa en la combinación de tres acciones elementales que corrigen el error ($e(t)$) entre una señal deseada (*setpoint*) y la salida medida del sistema, dichas acciones elementales se definen como:

Proporcional (P) [1]

Esta acción ajusta la salida del regulador de forma que sea proporcional al error. Por tanto, si el error o la perturbación es grande, la corrección también lo será, y viceversa, lo que resulta en un margen de error menor y una mayor estabilidad.

Integral (I)

Esta acción reduce progresivamente el error ajustando la salida en función de su integral. Esto minimiza los picos al implementar un sistema, logrando así un sistema más estable y preciso.

Derivativo (D)

Esta acción predictiva permite al sistema reaccionar ante futuros eventos basándose en las tendencias observadas, lo que hace que el sistema sea más preciso y pueda corregirse ante perturbaciones durante su funcionamiento.

La representación del PID en tiempo continuo es [1]:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

donde:

- $u(t)$ es la señal de control continua,
- $e(t)$ es el error continuo,
- K_p , K_i , K_d son las constantes proporcional, integral y derivativa, respectivamente.

Para implementarlo digitalmente se utiliza la ecuación en diferencias del PID [2]:

$$u(k) = K_p e(k) + K_i T \sum e(k) + K_d \left(\frac{e(k) - e(k-1)}{T} \right) \quad (2)$$

donde:

- $u(k)$ es la señal de control discreta,
- $e(k)$ es el error discreta,
- K_p , K_i , K_d son las constantes proporcional, integral y derivativa, respectivamente,
- T es el período de muestreo.

En un sistema digital, cada componente del PID se describe de manera diferente mediante aproximaciones numéricas [3]:

- **Proporcional:** Su implementación matemática permanece igual, aunque al implementarse digitalmente, el valor de la constante está limitado por el tamaño de la palabra en bits.
- **Integral:** Se realizan aproximaciones matemáticas usando métodos como las Sumas de Riemann para obtener un valor cercano al área real bajo la curva. La precisión depende del tamaño de la ventana de integración, determinado por la velocidad de muestreo.
- **Derivativo:** La derivación numérica se realiza mediante diferencias hacia atrás, calculando la derivada de la señal en un punto como la diferencia entre valores actuales y anteriores. [4].

PROCESO DE DISEÑO

Se establece que la planta debe tener una interfaz gráfica que permita al usuario ingresar los parámetros K_p , T_i y T_d del sistema, y así observar las diferentes respuestas del PID Discreto mediante una gráfica de la salida y el *setpoint* definido en Python.

La planta calefactora está conformada por un sistema de potencia a pequeña escala que incluirá un transistor Darlington y una resistencia calefactora de baja resistividad, controlado por una plataforma de desarrollo que leerá los valores mediante un sensor. El diseño de la planta se divide en dos partes fundamentales: la parte física (circuitos) y la parte digital (*software*), que en conjunto permitirán el correcto funcionamiento del PID.

DISEÑO DE LA PLANTA FÍSICA

Se estableció que la planta física estará conformada por un pequeño sistema de potencia, utilizando un transistor tipo Darlington. Para observar el comportamiento del sistema, se incluirá una pantalla LCD 16x2 que mostrará si está esperando valores del PID o el valor de *setpoint* asignado y la temperatura de la planta.

Dado que la plataforma de desarrollo utilizada es un Arduino UNO, el cual no posee salidas analógicas capaces de generar un voltaje proporcional al error que se desea corregir, se utilizará modulación por ancho de pulso (*PWM*). Así, dependiendo del valor de salida calculado por el PID, se obtendrá una señal *PWM* de ancho variable en la salida.

Se utiliza un transistor tipo Darlington, capaz de manejar la corriente máxima que circulará a través de la resistencia calefactora seleccionada, de 5 Ω .

Se eligió como elemento calefactor una resistencia de 5 Ω considerando que la fuente utilizada, un módulo de fuente de poder de 12V DC y 3A (entrada: 100V-220V AC), puede entregar suficiente corriente para variar la temperatura.

Corriente máxima con una resistencia de 5 Ω [5]:

$$I = \frac{V}{R} = \frac{12\text{ V}}{5\ \Omega} = 2.4\text{ A} \quad (3)$$

Por la capacidad máxima de corriente capaz de brindar la fuente se usará una resistencia de 5 Ω .

En cuanto a la conexión, el transistor tendrá la base conectada a través de una resistencia limitadora a la plataforma de desarrollo, el emisor conectado a tierra, y el colector conectado a la resistencia calefactora, la cual a su vez se alimentará con los 12V de la fuente.

Debemos asegurarnos que cuando la salida del Arduino esté en *HIGH* (5V) toda la corriente que puede brindarnos de forma segura la fuente pase por la resistencia. [6]:

$$I_b = \frac{V - V_{BE}}{R} \quad (4)$$

$$I_b = \frac{5\text{ V} - 0.7\text{ V}}{1000\ \Omega} = 4.3\text{ mA}$$

$$I_c = \beta I_b \quad (5)$$

$$I_c = 1000 \times 4.3\text{ mA} = 4.3\text{ A}$$

Este dato es superior a la corriente máxima que puede pasar por la resistencia calefactora, por lo que usaremos la fuente de forma eficiente para tiempos de funcionamiento y estabilización más cortos.

DISEÑO DEL SOFTWARE

La planta consiste en dos programas separados, uno en la plataforma de desarrollo y otro en Python. El código en Python incluye la Interfaz Gráfica de Usuario (IGU), a través de la cual es posible observar los cambios de temperatura en la resistencia calefactora y compararlos con el *setpoint*. Además, permite ingresar los valores de K_p , T_i y T_d , siendo la constante proporcional, tiempo integral y tiempo derivativo respectivamente. El eje x se expandirá a medida que se ingresen más datos hasta que se presione "Guardar Datos", lo que almacena los datos recopilados en un documento de Microsoft Excel. El programa debe cerrarse con el botón "Terminar" para liberar el puerto COM utilizado por la plataforma de desarrollo.

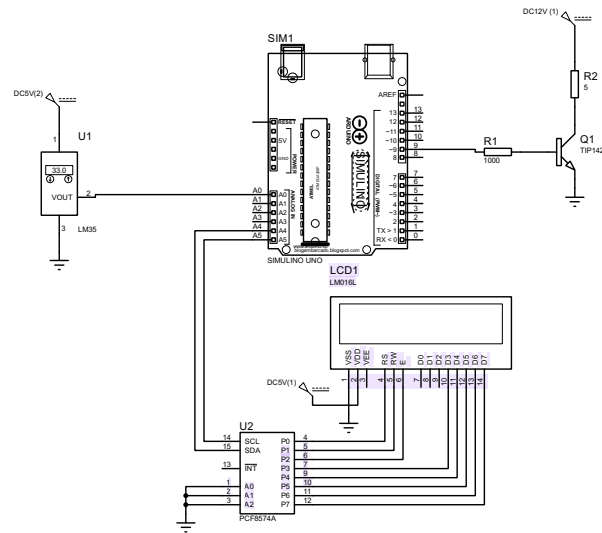


Fig. 1. Diagrama esquemático de la planta.

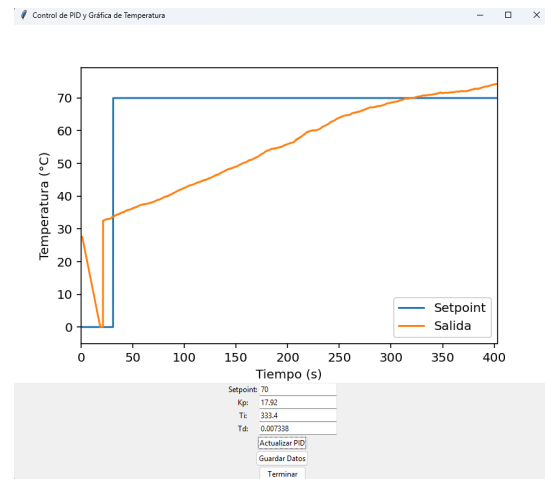


Fig. 2. Interfaz Gráfica de Usuario (IGU).

Es posible actualizar los valores del PID y del *setpoint* durante el funcionamiento. La transferencia de datos hacia la plataforma de desarrollo se realiza mediante comunicación serial.

FUNCIÓN DE TRANSFERENCIA, ESTABILIDAD Y SINTONIZACIÓN

Se creó un pequeño programa en la plataforma de desarrollo que simula una planta sin retroalimentación con un valor de K_p fijo y sin valores para K_i y K_d , con el objetivo de simular una planta en lazo abierto. Utilizando el script de Python, se guardaron los datos en un documento de Microsoft Excel para analizarlos posteriormente en MATLAB.

Estos datos se llevaron a MATLAB donde, mediante *System Identification*, se realizaron iteraciones con distintos polos y ceros en tiempo discreto para encontrar la función de transferencia que tuviera mayor porcentaje de similitud con la planta utilizada, como se observa en la Fig. 5 y Fig. 6.

En la Fig. 7 se observa que la función de transferencia en

	A	B	C
1	Time (s)	Setpoint(°C)	Salida (°C)
2	0.147	70	34.65
3	0.294	70	34.65
4	0.441	70	34.65
5	0.588	70	34.65
6	0.735	70	34.65
7	0.882	70	34.65
8	1.029	70	34.65
9	1.176	70	34.65
10	1.323	70	34.65
11	1.47	70	34.65
12	1.617	70	34.65
13	1.764	70	34.65
14	1.911	70	34.65
15	2.058	70	34.65

Fig. 3. Datos guardados

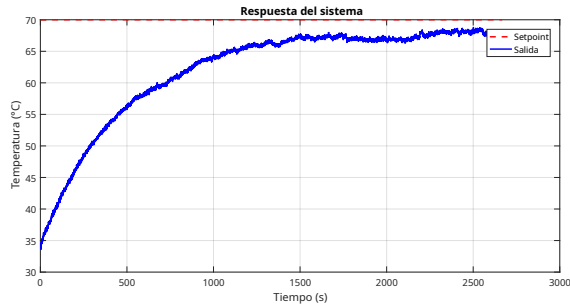


Fig. 4. Gráfica en lazo abierto

tiempo discreto con el mayor porcentaje de similitud (94.8 %) es la siguiente:

$$H(z) = \frac{0.00053155 \cdot z^{-1}}{1 - 0.2139z^{-1} - 0.7856z^{-2}} \quad (6)$$

Es importante recordar que esta función de transferencia es en lazo abierto. Con el comando "feedback" se obtiene fácilmente la función de transferencia en lazo cerrado.

$$H_{cl}(z) = \frac{0.00053155 \cdot z^{-1}}{1 - 0.2134z^{-1} - 0.7856z^{-2}} \quad (7)$$

Con la función de transferencia determinada, podemos encontrar los polos y ceros y graficarlos en la circunferencia unidad para determinar si la planta es estable. Con MATLAB se determina que la función de transferencia tiene ceros en $z=0$ y polos en $p=0.9994$ y $p=-0.7860$. En la Fig. 8 se aprecia

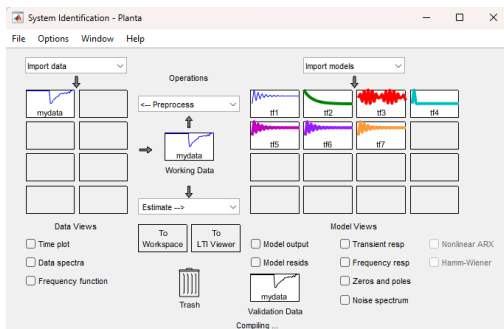


Fig. 5. Diferentes iteraciones con distinto número de polos y ceros

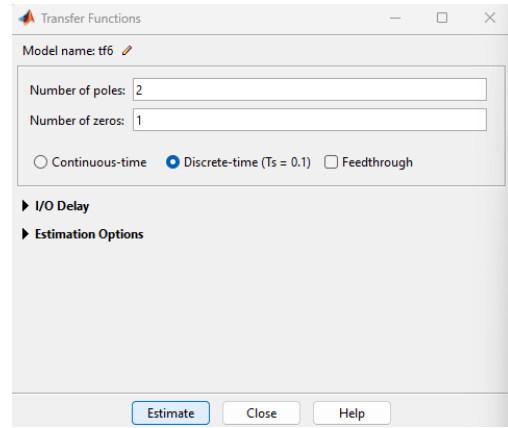


Fig. 6. Configuración de polos y ceros en tiempo discreto

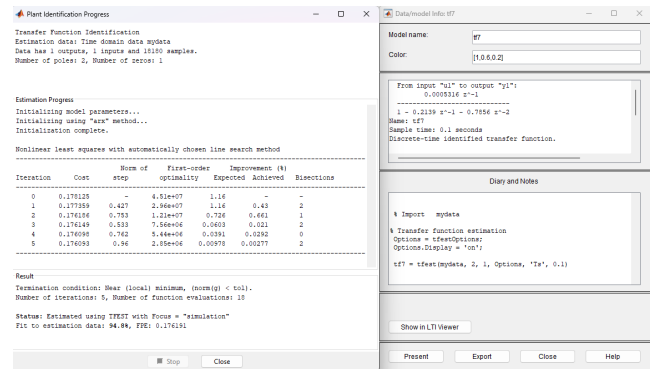


Fig. 7. Función de transferencia con mayor porcentaje de similitud

que los polos están dentro de la circunferencia unidad, por lo tanto el sistema es estable.

Una vez diseñado el sistema PID, se procedió a su sintonización utilizando las herramientas de MATLAB, específicamente *PID Tuner*, como se observa en la Fig. 9. Con la función de transferencia determinada, se pueden obtener los valores de K_p , T_i , T_d [7] con los cuales la planta se estabiliza de manera más óptima.

Se obtuvo que la planta se estabiliza de forma más eficiente alrededor de los valores $K_p = 17.92$, $T_i = 333.4$ y $T_d = 0.007338$.

En la plataforma de desarrollo se realiza la lectura del sensor de temperatura, el cálculo de la salida del PID y su transformación en una señal PWM. La ecuación en diferencias del PID es:

$$u(k) = K_p e(k) + K_i T \sum e(k) + K_d \left(\frac{e(k) - e(k-1)}{T} \right) \quad (8)$$

Los valores de T_i y T_d enviados desde Python se convierten mediante las siguientes ecuaciones:

$$K_i = \frac{K_p}{T_i} \quad (9)$$

$$K_d = K_p \cdot T_d \quad (10)$$

En la plataforma de desarrollo, este cálculo se realiza de la siguiente manera [8]:

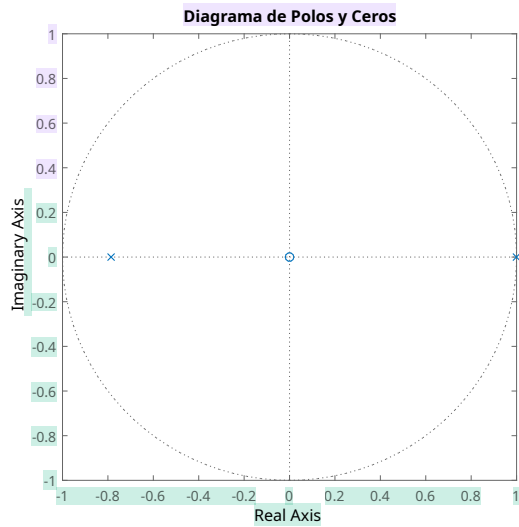


Fig. 8. Polos y ceros

Controller Parameters	
	Tuned
Kp	17.9236
Ti	333.3827
Td	0.0073384
N	n/a
Performance and Robustness	
	Tuned
Rise time	116 seconds
Settling time	813 seconds
Overshoot	8.85 %
Peak	1.09
Gain margin	32.3 dB @ 0.547 rad/s
Phase margin	78.6 deg @ 0.0136 rad/s
Closed-loop stability	Stable

Fig. 9. Sintonización.

```
double error = setpoint - filteredInput;
integral += error * T;
double derivative = (error - prevError) / T;
output = Kp * error + Ki * integral + Kd * derivative;
prevError = error;
```

Donde output es el valor que después se escala en un pulso PWM.

En el bloque de cálculo del PID se emplea una variable denominada *filteredInput*, que corresponde a la señal de entrada filtrada. Dado que el sensor de temperatura utilizado (LM35) presenta ruido de alta frecuencia superpuesto a la señal útil, se implementó un filtro de tipo promediado móvil. Este filtro puede representarse mediante la siguiente ecuación en diferencias [9]:

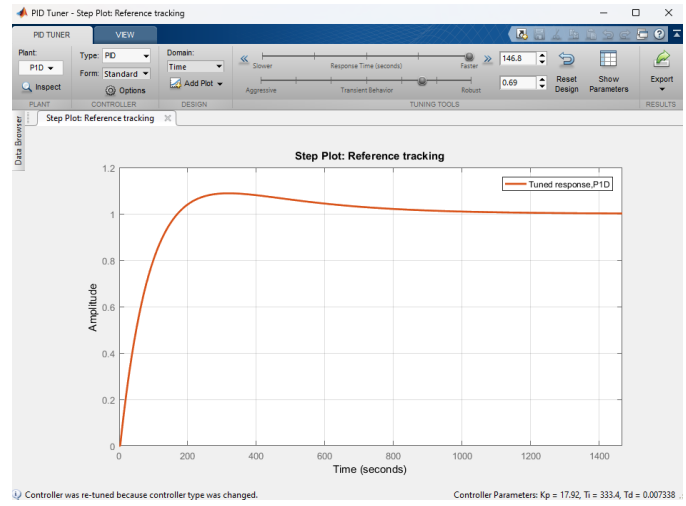


Fig. 10. Simulación de la planta

$$\text{filteredInput} = \frac{1}{N} \sum_{i=0}^{N-1} \text{input}[k-i] \quad (11)$$

Donde $\text{input}[k]$ es la medición del sensor en el instante de muestreo k , y N es el número de muestras consideradas en la ventana. Se tomó un período de muestreo de $T_s = 100$ ms dado que un sistema térmico no produce cambios de temperatura de forma brusca, además de facilitar el análisis posterior en MATLAB, donde las funciones de transferencia discretas se analizan por defecto con $T_s = 0.1$ s. En este caso, se consideró $N = 25$, y el período de muestreo del sistema es $T_s = 100$ ms. Por lo tanto, el filtro realiza el promedio de los datos capturados durante:

$$T = N \cdot T_s = 25 \cdot 100 \text{ ms} = 2.5 \text{ s} \quad (12)$$

En sistemas térmicos (de respuesta lenta) es posible utilizar ventanas de filtrado relativamente grandes, como $N = 25$, sin comprometer la dinámica del sistema, especialmente cuando el tiempo de muestreo es suficientemente pequeño. Este intervalo de filtrado de 2.5 segundos proporciona un funcionamiento adecuado limitando el ruido aleatorio. Dado que las variaciones de temperatura en la planta son lentas, aplicar un filtro de ventana de 25 muestras no degrada significativamente la respuesta del sistema, pero mejora notablemente la estabilidad y la calidad de la señal medida [10].

PRUEBAS REALIZADAS

Se realizaron diversas pruebas para comprobar el correcto funcionamiento del PID Discreto.

En la Fig. 11 y en la Fig. 12 se observa que al cambiar los valores de K_p , T_i y T_d , la respuesta de la planta varía, el objetivo es que los estudiantes vean los efectos de cambiar los valores. Con los datos del documento se calculó:

Error de estado estable:

$$e_{ss} = 70.85^\circ\text{C} - 70^\circ\text{C} = 0.85^\circ\text{C} \quad (13)$$

Sobreimpulso:

$$M_p = \left(\frac{73.76^\circ\text{C} - 70^\circ\text{C}}{70^\circ\text{C}} \right) \times 100\% = 5.37\% \quad (14)$$

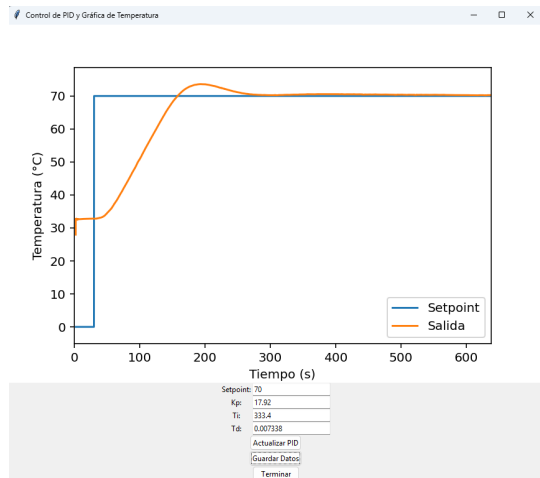


Fig. 11. Respuesta con valores de MATLAB.

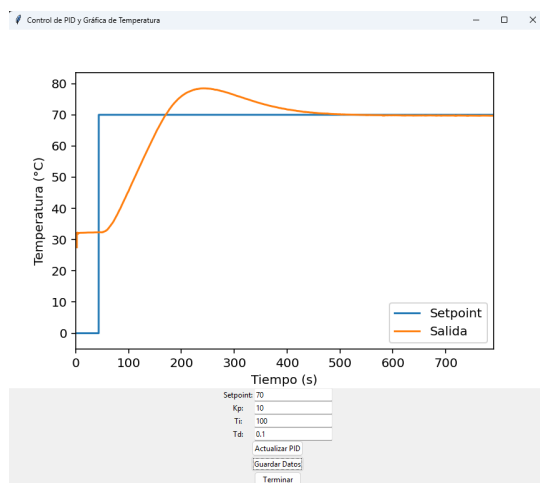


Fig. 12. Respuesta con valores arbitrarios.

El tiempo de estado estable (t_{ss}) fue de 443.612 segundos, con un error de ± 0.1 segundos, correspondiente al período de muestreo utilizado.

$$t_{ss} = 443.612 \pm 0.1 \text{ s} \quad (15)$$

ANÁLISIS, COMPARATIVA Y VALIDACIÓN DIDÁCTICA DE SOLUCIONES EXISTENTES

Con el objetivo de exponer y validar la propuesta desarrollada, se realizó una comparación con un proyecto externo titulado *Diseño e implementación de módulos para prácticas de controladores PID convencionales*, desarrollado en la Universidad Politécnica Salesiana [11]. En dicho proyecto se desarrollaron diversos módulos didácticos utilizando PID. Para ser objetivos, se comparará con el PID discreto desarrollado en este trabajo, ya que en la documentación existen módulos análogos.

Como se muestra en la Tabla I, ambos sistemas son bastante similares; sin embargo, falta comparar los costos aproximados de ambas soluciones, dado que se estableció que la solución aquí planteada debía ser asequible para cualquier institución que no tuviera la capacidad de adquirir material didáctico

Tabla I
COMPARATIVA DE MÓDULOS PID

Parámetro	Solución propia	Solución previa
Tipo de implementación	Digital	Digital
Método de ajuste PID	Digital des-de interfaz	Digital des-de interfaz
Visualización de resultados	En tiempo real	En tiempo real
Software de visualización	Python	LabVIEW
Capacidad de registrar datos	Sí	Sí
Capacidad de guardar datos	Sí	No
Capacidad de cambiar período de muestreo	No	Sí
Tipo de planta	Calefactora	Calefactora
Facilidad de montaje y conexión	Alta	Alta
Seguridad eléctrica para uso didáctico	Sí	Sí
Flexibilidad para modificar parámetros	Alta	Media
Capacidad de simular perturbaciones	No	No
Interfaz de usuario	Software + LCD	Software
Disponibilidad del código fuente	Sí	No

Tabla II
COSTOS ESTIMADOS SOLUCIÓN PROPIA. FUENTE: [12].

Componente	Precio
Arduino	\$11.49
Transistor de Potencia	\$0.50
Sensor LM35	\$1.75
Resistencia calefactora	\$0.60
Pantalla LCD	\$5.25
Fuente de alimentación	\$13.95
Resistencia 1/2 W (1)	\$0.06
Estructura	\$5.75
Total	\$39.35

especializado. El cálculo para la solución externa se realizó basándose en los costos actuales de los materiales descritos en el documento [11].

El análisis comparativo de costos (Tablas II y III) revela que la solución propuesta no solo es más económica (\$39.35 vs. \$44.05), sino técnicamente superior. Nuestro diseño reemplaza los frágiles focos de la solución externa, susceptibles a daños por golpes o sobrecalentamiento, con una resistencia calefactora robusta de mayor vida útil. Esta elección de componentes, junto con un esquema de potencia simplificado, reduce puntos críticos de falla mientras mantiene un 12 % menor costo total.

Se realizaron sesiones de prueba con dos estudiantes que están cursando la materia correspondiente al análisis y entendimiento del PID, a ellos se les brindó una laptop con la IGU en Python y el módulo desarrollado. Ambos estudiantes hicieron 3 pruebas cada uno y comentaron que:

- Aumentar K_p reduce el tiempo de respuesta pero puede generar sobreimpulso.
- Al reducir demasiado el T_i la variable se vuelve inestable

Tabla III
COSTOS ESTIMADOS SOLUCIÓN EXTERNA. FUENTE: [12], [13]

Componente	Precio
Arduino	\$11.49
Transistor de Potencia	\$0.50
Sensor LM35	\$1.75
Focos 21 W (2)	\$0.70
Ventilador 12 V (2)	\$14.98
Potenciómetro	\$0.52
Diodo 1N4007	\$0.20
Transistor 2N2222	\$0.23
Resistencias (2)	\$0.18
Regulador LM7805	\$1.00
Estructura	\$12.50
Total	\$44.05

y comienza a oscilar.

- El término T_i sin ayuda de T_d tiende a ser inestable si no se sintoniza bien.
- Un aumento moderado en K_p puede reducir el tiempo de estado estable (t_{ss}), pero si se excede puede provocar oscilaciones que lo alargan en lugar de acortarlo

Además obtuvimos las siguientes retroalimentaciones:

- Se sugirió dar la opción de escoger el nombre del documento generado desde una ventana (actualmente el código genera un documento bajo el nombre data.xlsx).
- Cambiar la planta calefactora por una de respuesta más rápida.

La visualización gráfica en Python fue especialmente útil para reforzar los conceptos teóricos mediante la observación directa de la dinámica del sistema.

Nuestro código fuente está disponible en GitHub.¹

CONCLUSIONES

La solución planteada resulta ideal para comprender los conceptos fundamentales del control PID a nivel básico, permitiendo modificar los parámetros K_p , T_i y T_d para observar directamente su impacto en el comportamiento de la planta mediante representación gráfica. Este sistema destaca por su accesibilidad económica, facilidad de implementación y adecuadas protecciones eléctricas para su categoría.

El análisis de estabilidad del sistema se realiza mediante la ubicación de polos en el diagrama de polos y ceros en tiempo discreto, donde herramientas como *PID Tuner* y *System Identification* de MATLAB resultan fundamentales para obtener la representación matemática precisa de las plantas físicas a partir de datos experimentales.

El control PID digital en tiempo discreto demuestra una eficiencia comparable a su contraparte analógica en tiempo continuo cuando está correctamente implementado, ofreciendo además las ventajas inherentes a los sistemas digitales: versatilidad, compactibilidad y facilidad de ajuste.

La comunicación serial entre la computadora y el Arduino permite no solo la visualización en tiempo real de la respuesta térmica frente al *setpoint* establecido, sino también la transferencia eficiente de datos para su almacenamiento y posterior

análisis. Python emerge como plataforma ideal para esta interfaz, gracias a su flexibilidad para manejar comunicación serial, procesamiento de datos en tiempo real y generación de registros completos para análisis matemáticos avanzados.

Los estudiantes fueron capaces de comprender las bases de un sistema PID y los efectos de las distintas constantes a través de nuestro módulo.

REFERENCIAS

- [1] K. Ogata, *Ingeniería de control moderna*, 5.a ed., PEARSON, 2010.
- [2] B. C. Kuo, *Sistemas de control automático*, Pearson Educación, 1996.
- [3] C. A. Núñez Molina, *Diseño e implementación de un módulo para apoyo a la docencia: Controlador PID discreto*, Universidad del Bío-Bío, 2015. [En línea]. Disponible en: <http://repobib.ubiobio.cl/jspui/bitstream/123456789/3271/1/Nuez%20Molina%2C%20Carlos%20Andrs.pdf>
- [4] S. Dormido y F. Morilla, *Controladores PID: Fundamentos, sintonía y autosintonía*, UNED. [En línea]. Disponible en: <http://www0.unsl.edu.ar/~control1/apuntes/C6.2.pdf>
- [5] C. K. Alexander y M. N. O. Sadiku, *Fundamentos de circuitos eléctricos*, 5.a ed., McGraw-Hill, 2013. [En línea]. Disponible en: <https://www.cartagena99.com/recursos/alumnos/apuntes/1.1/%20Fundamentos%20de%20circuitos%20electricos%205ta.%20Edicion%20sadiku.pdf>
- [6] J. L. Valls, J. M. López, y J. M. Marí, *Fundamentos de electrónica analógica*, Universitat Politècnica de València, 2017. [En línea]. Disponible en: https://api.pageplace.de/preview/DT0400.9788437086545_A42772655/preview-9788437086545_A42772655.pdf
- [7] Centro de ayuda de MATLAB, "PID Tuner - Tune PID controllers - MATLAB." [En línea]. Disponible en: <https://www.mathworks.com/help/control/ref/pidtuner-app.html>
- [8] S. A. C. Giraldo, "Lazo abierto y lazo cerrado, Control Automático Educación." 14 de agosto de 2024. [En línea]. Disponible en: <https://controlautomaticoeducacion.com/control-realimentado/lazo-abierto-y-lazo-cerrado/>
- [9] W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, 1997.
- [10] K. J. Åström y B. Wittenmark, *Computer-Controlled Systems: Theory and Design*, 3.a ed., Dover Publications, 2011.
- [11] E. G. Paredes, *Diseño e implementación de módulos para prácticas de controladores PID convencionales*, Universidad Politécnica Salesiana, 2019. [En línea]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/17259/1/UPS-GT002570.pdf>
- [12] Steren, *Tienda Electrónica*, [En línea]. Disponible en: <https://www.steren.com.sv>. [Accedido: Mayo 2025].
- [13] Freund, *La ferretería de los salvadoreños*, [En línea]. Disponible en: <https://www.freundferreteria.com>. [Accedido: Mayo 2025].

¹<https://github.com/Alb3rt0467/PID-DISCRETO>