

Aplicación de un controlador PID de orden fraccionario en sistemas físicos: circuito RLC serie y sistema de temperatura

Denilson Alfonso Carrascal Piñeres, Diego Fernando Cárdenas Vargas

Trabajo de Grado para Optar el Título de Ingeniero Electronico

Director

Carlos Rodrigo Correa Cely

Doctor en ciencia e ingeniería de polímeros

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones (E3T)

Bucaramanga

2025

Tabla de Contenido

	Pág.
Introducción	9
1. Objetivos	11
1.1 Objetivo General	11
1.2 Objetivos Específicos.....	11
2. Cuerpo del Trabajo	12
2.1 Marco Referencial.....	12
2.1.1. Marco teórico.....	12
2.1.2 Método.	16
2.1.3 Resultados.....	39
3. Conclusiones	40
4. Recomendaciones	40
Referencias Bibliográficas	43
Apéndices.....	44

Lista de Tablas

	Pág.
Tabla 1. Configuración del sistema de temperatura.....	27
Tabla 2. Comparativa comportamientos control del circuito RLC.....	38
Tabla 3. Comparativa comportamientos control del sistema térmico.....	38

Lista de Figuras

	Pág.
Figura 1. Respuesta del circuito RLC ante un escalón.....	18
Figura 2. Respuesta del circuito RLC con un control de orden entero.....	19
Figura 3. Diagramas de ganancia y fase del circuito RLC controlado con un IOPID.....	20
Figura 4. GUI del optimizador de la toolbox FOMCON.....	22
Figura 5. Comportamiento en el tiempo del circuito con un FOPID.....	23
Figura 6. Diagrama de margen de ganancia y de fase del circuito con un FOPID.....	24
Figura 7. Diagramas de bode de $S(j\omega)$ del circuito con un FOPID.....	25
Figura 8. Diagramas de bode de $T(j\omega)$ del circuito con un FOPID.....	25
Figura 9. Diagrama de bloques para la adquisición de datos del sistema de temperatura.....	28
Figura 10. Programación del Arduino para comunicarse con Simulink.....	29
Figura 11. Gráfica de la caracterización de la planta usando System Identification.....	30
Figura 12. Sistema de temperatura en lazo cerrado con IOPID sintonizado con PID Tuner.....	31
Figura 13. Diagrama de bloques para realizar la implementación de un IOPID.....	32

Figura 14. Gráfica de la temperatura controlada con el IOPID.....33

Figura 15. Configuración de la GIU FOMCON para hallar el FOPID.....35

Figura 16. Temperatura controlada en simulación con FOMCON.....36

Figura 17. Diagrama de bloques para realizar la implementación de un FOPID.....36

Figura 18. Grafica de la temperatura controlada con un FOPID.....37

Lista de Apéndices

	Pág.
Apéndice A. Control IOPID circuito RLC serie.....	19
Apéndice B. Control FOPID circuito RLC serie.....	23
Apéndice C. Diagrama para la adquisición de datos para el sistema de temperatura.....	28
Apéndice D. Programación del Arduino para transmisión y recepción de datos.....	29
Apéndice E. Datos adquiridos del sistema de temperatura.....	30
Apéndice F. Identificación de la planta del sistema de temperatura.....	30
Apéndice G. Diagrama de bloques implementación IOPID.....	32
Apéndice H. Configuración de FOMCON para sintonización del FOPID.....	35
Apéndice I. Diagrama de bloques implementación FOPID.....	36

Resumen

Título: Aplicación de un controlador PID de orden fraccionario en sistemas físicos: circuito RLC serie y sistema de temperatura*

Autor: Denilson Alfonso Carrascal Piñeres, Diego Fernando Cárdenas Vargas, Carlos Rodrigo Correa Cely**

Palabras Clave: calculo fraccionario, control fraccionario, sistemas físicos

Descripción: El presente trabajo de grado aborda la evaluación del desempeño de controladores PID de orden fraccionario (FOPID) frente a controladores tradicionales de orden entero (IOPID), aplicados a dos tipos de sistemas físicos: un circuito RLC simulado y un sistema de control de temperatura implementado físicamente. La creciente automatización de procesos abre una puerta para mejorar la precisión, robustez y eficiencia de los sistemas automatizados motiva la exploración de nuevas estrategias de control como el cálculo fraccionario, el cual permite una mayor flexibilidad mediante la introducción de órdenes no enteros en las acciones derivativas e integrales. Se diseñaron y sintonizaron ambos tipos de controladores utilizando herramientas de optimización en MATLAB, aplicando el toolbox FOMCON para el FOPID. Los controladores fueron evaluados mediante simulación (circuito RLC) e implementación física (sistema de temperatura), comparando métricas como el tiempo de establecimiento, sobreimpulso, error en estado estacionario. Los resultados preliminares evidencian que el FOPID ofrece mejoras significativas en la respuesta dinámica y en la robustez del sistema frente a perturbaciones. Estas conclusiones abren la posibilidad de implementar este tipo de control en aplicaciones reales de mayor complejidad, aportando valor tanto en contextos académicos como industriales.

* Trabajo de Grado

** Facultad de ingenierías fisicomecánicas. Escuela de ingenierías eléctrica, electrónica y telecomunicaciones. Director: Carlos Rodrigo Correa Cely. Doctor en ciencia e ingeniería de polímeros

Abstract

Title: Application of a fractional-order PID controller in physical systems: series RLC circuit and temperature system*

Author: Denilson Alfonso Carrascal Piñeres, Diego Fernando Cárdenas Vargas, Carlos Rodrigo Correa Cely**

Key Words: fractional calculus, fractional control, physical systems

Description: This thesis addresses the evaluation of the performance of fractional-order PID controllers (FOPID) compared to traditional integer-order controllers (IOPID), applied to two types of physical systems: a simulated RLC circuit and a physically implemented temperature control system. The growing automation of processes opens the door to improving the accuracy, robustness, and efficiency of automated systems, motivating the exploration of new control strategies such as fractional calculus, which allows for greater flexibility by introducing non-integer orders in derivative and integral actions. Both types of controllers were designed and tuned using optimization tools in MATLAB, applying the FOMCON toolbox for the FOPID. The controllers were evaluated through simulation (RLC circuit) and physical implementation (temperature system), comparing metrics such as settling time, overshoot, and steady-state error. Preliminary results show that FOPID offers significant improvements in dynamic response and system robustness against disturbances. These conclusions open up the possibility of implementing this type of control in more complex real-world applications, adding value in both academic and industrial contexts.

* Thesis

** Faculty of physical and mechanical engineering, School of Electrical, Electronic and Telecommunications Engineering. Director: Carlos Rodrigo Correa Cely, Doctor of science and polymer engineering

Introducción

En la actualidad, el control automático es un componente esencial en numerosos sistemas físicos que requieren regulación precisa y eficiente, como los circuitos electrónicos y los procesos térmicos. En este contexto, los controladores PID (Proporcional–Integral–Derivativo) de orden entero han sido durante décadas la solución predominante debido a su simplicidad y buen desempeño en una amplia gama de aplicaciones. Sin embargo, estos controladores pueden presentar limitaciones en sistemas con dinámicas complejas o no lineales, lo que ha motivado la búsqueda de alternativas más flexibles.

Una de las alternativas emergentes es el controlador PID de orden fraccionario (FOPID), que extiende el concepto clásico incorporando derivadas e integrales de orden no entero. Esta modificación permite una mayor capacidad de ajuste fino, lo que puede traducirse en mejoras en estabilidad, robustez y tiempo de respuesta. Aunque el control fraccionario ha mostrado buenos resultados en aplicaciones como el péndulo invertido o procesos industriales, su estudio en sistemas eléctricos y térmicos sigue siendo un campo en exploración.

Este trabajo propone evaluar el desempeño del FOPID en dos escenarios complementarios: la simulación de un circuito RLC de orden entero y la implementación física de un sistema de control de temperatura. En ambos casos, se diseñaron y compararon controladores FOPID e IOPID utilizando herramientas de optimización disponibles en MATLAB. A partir de las simulaciones y pruebas experimentales, se analizaron métricas como el sobreimpulso, tiempo de establecimiento, error en estado estacionario y esfuerzo de control.

Lo que distingue a este trabajo es su enfoque dual, que combina simulación y experimentación física para ofrecer una visión más completa del comportamiento del control

fraccionario en distintos tipos de sistemas. Los resultados obtenidos muestran diferencias significativas entre ambos tipos de controladores, lo que refuerza la idea de que el control fraccionario representa un campo con gran potencial para futuras investigaciones.

Este proyecto busca no solo validar técnicamente el FOPID en contextos distintos, sino también aportar al desarrollo de estrategias de control más robustas y adaptables, con posibles aplicaciones en la automatización, electrónica y sistemas embebidos.

1. Objetivos

1.1 Objetivo General

Diseñar un controlador PID de orden fraccionario para analizar su efecto en la dinámica de un circuito RLC y en un sistema de control de temperatura.

1.2 Objetivos Específicos

- Realizar los diseños de controladores PID de orden fraccionario y de orden entero para su aplicación en un circuito RLC y en un sistema de temperatura.
- Implementar en software y/o hardware los controladores PID de orden entero y PID de orden fraccionario.
- Comparar el comportamiento dinámico de los sistemas controlados con PID clásico (IOPID) y PID fraccionario (FOPID) en cada sistema.

2. Cuerpo del Trabajo

2.1 Marco Referencial

2.1.1 Marco teórico

2.1.1.1 Introducción a la teoría de control y controladores PID. El control automático constituye la base para la regulación de variables de procesos industriales y sistemas físicos. El controlador PID (Proporcional, Integral y Derivativo) ha sido el esquema de control más implementado gracias a su simplicidad y robustez. Su versión clásica se expresa como:

$$C(s) = Kp + \frac{Ki}{s} + Kds \quad (1)$$

donde Kp , Ki y Kd son las ganancias proporcional, integral y derivativa respectivamente. (Xue, Chen, & Atherton, 2007)

2.1.1.2 Control Fraccionario: Controlador $PI^\lambda D^\mu$. En los últimos años, se ha demostrado que extender el PID clásico a *órdenes fraccionarios* mejora la flexibilidad de diseño. El *controlador FOPID* o $PI^\lambda D^\mu$ generaliza los términos integral y derivativo con órdenes no enteros:

$$C(s) = Kp + \frac{Ki}{s^\lambda} + Kds^\mu \quad \text{con } 0 < \lambda, \mu < 1 \quad (2)$$

Esta estructura introduce dos grados de libertad adicionales (λ y μ) permitiendo una sintonización más flexible. (Podlubny, 1999)

2.1.1.3 Fundamentos del cálculo fraccionario. El cálculo fraccionario permite extender la definición de derivadas e integrales a órdenes no enteros. Algunas definiciones comunes incluyen:

- **Riemann-Liouville:**

$$D_t^\alpha f(t) = \left(\frac{1}{\Gamma(m-\alpha)} \right) \times \frac{d^m}{dt^m} \times \int_a^t [f(\tau) \times (t-\tau)^{m-\alpha-1}] d\tau \quad (3)$$

- **Grünwald-Letnikov**, útil en discretización numérica:

$$D_t^\alpha f(t) = \lim_{h \rightarrow 0} \left[\frac{1}{h^\alpha} \right] \times \sum_{j=0}^{\lfloor (t-a)/h \rfloor} (-1)^j \binom{\alpha}{j} f(t-jh) \quad (4)$$

Estas definiciones se aproximan en la práctica mediante filtros racionales como el Filtro de Oustaloup, ampliamente implementado en FOMCON. (Tepljakov, Petlenkov, & Belikov, 2011)

- **Filtro Oustaloup:**

$$s^\alpha \approx K \prod_{i=-N}^N \frac{s+w_i}{s+w_i'} \quad (5)$$

$$w_i = w_b \left(\frac{w_h}{w_b} \right)^{\frac{i+N+\frac{1}{2}(1+\alpha)}{2N+1}} \quad (6)$$

$$w_i' = w_b \left(\frac{w_h}{w_b} \right)^{\frac{i+N+\frac{1}{2}(1-\alpha)}{2N+1}} \quad (7)$$

$$K = w_h^\alpha \quad (8)$$

- **Filtro refinado Oustaloup**

$$s^\alpha \approx \left(\frac{9w_h}{10} \right)^\alpha \left(\frac{9s^2+10w_h s}{9(1-\alpha)s^2+10w_h s+9\alpha} \right) G_p \quad (9)$$

$$G_p = \prod_{i=-N}^N \frac{s+w_i}{s+w_i'} \quad (10)$$

$$w_i = \left(\frac{10w_h}{9} \right)^{\frac{\alpha+2i}{2N+1}} \quad (11)$$

$$w_i' = w_b \left(\frac{w_h}{w_b} \right)^{\frac{\alpha-2i}{2N+1}} \quad (12)$$

Estas ecuaciones permiten expresar de forma racional un FOPID. (Tepljakov, Petlenkov, & Belikov, 2011)

2.1.1.4 Ecuaciones clave para el diseño del FOPID. La *sintonización de un FOPID* requiere resolver *cinco ecuaciones* simultáneas para hallar las incógnitas Kp , Ki , Kd , λ , μ . Estas ecuaciones derivan de los requisitos de margen de fase, margen de ganancia, frecuencia de cruce de ganancia y forma de la respuesta deseada.

- **Función de transferencia del lazo abierto:**

$$L(s) = C(s) \times G(s) = \left[Kp + \frac{Ki}{s^\lambda} + Kd \times s^\mu \right] \times G(s), \text{ G(s) denota la planta (13)}$$

- **Condición de cruce de ganancia:**

$$|L(j\omega_c)| = 1 \quad (14)$$

- **Condición de margen de fase:**

$$\angle L(j\omega_c) = -\pi + PM \quad (15)$$

donde ω_c es la frecuencia de cruce y PM es el margen de fase deseado.

- **Derivada de la fase (pendiente de fase):**

$$\frac{d}{d\omega} [\angle L(j\omega)] = 0 \quad (16)$$

Esto controla la pendiente de la fase para optimizar la robustez dinámica.

- **Requisito de sensibilidad complementaria:**

Opcionalmente, se impone una restricción de rechazo de perturbaciones o error de seguimiento:

$$|T(j\omega)| = \frac{|L(j\omega)|}{[1+|L(j\omega)|]} < \text{nivel deseado [dB]} \quad (17)$$

- **Requisito de rechazo de perturbaciones:** Una restricción en la función de la sensibilidad $S(j\omega)$ se puede definir como:

$$S(j\omega) = \left| \frac{1}{1+C(j\omega)G(j\omega)} \right|_{dB} \leq \text{nivel deseado [dB]} \quad (18)$$

Estas ecuaciones permiten la sintonización de controladores fraccionarios. (Monje, Chen, Vinagre, Xue, & Feliu, 2010)

2.1.2 Método.

2.1.2.1 Descripción general. El proceso seguido para el diseño, simulación y validación de controladores PID y FOPID aplicados a dos sistemas físicos: un *circuito RLC serie* y un *sistema de temperatura*. Se utilizó *Simulink*, la librería *FOMCON* y *SISOTOOL* de MATLAB para la etapa de simulación y análisis. Adicionalmente, se realizó una implementación física del

sistema de temperatura usando un *sensor LM35*, *resistencia* y un *Arduino Uno* con el fin de comparar el desempeño real frente a los resultados de simulación.

2.1.2.2 Modelado circuito RLC serie. Primero se modeló la planta RLC serie mediante su función de transferencia, tomando como salida el voltaje en el capacitor, la ecuación resultante es:

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{(LCs^2 + RCs + 1)}$$

Donde:

- **L:** Inductancia
- **C:** Capacitancia
- **R:** Resistencia

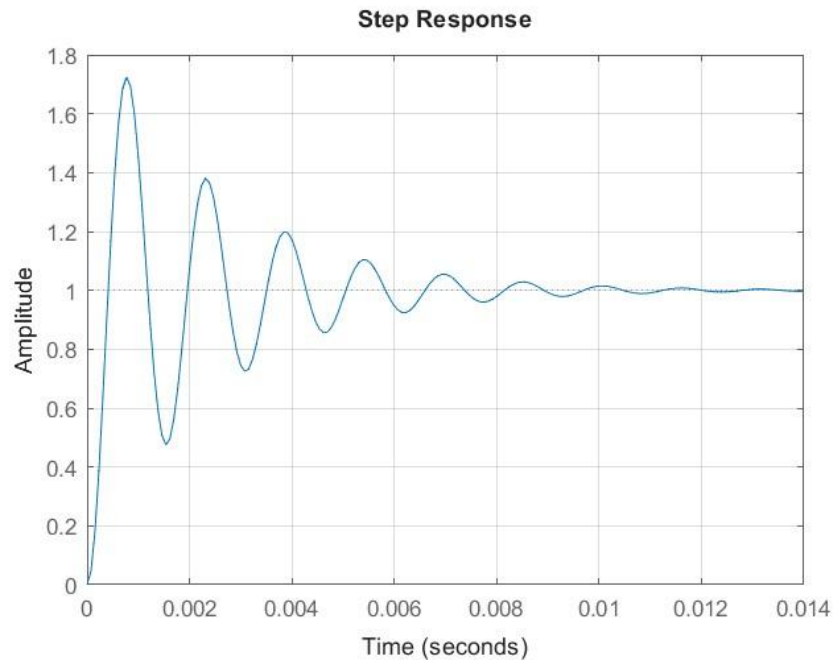
Los valores de R, L y C fueron los siguientes: R = 5 Ω , L = 6mH y C = 10uF. Con esto, la función de transferencia es:

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{(6 * 10^{-8}s^2 + 5 * 10^{-5}s + 1)}$$

La respuesta del sistema en lazo cerrado a un escalón es la siguiente:

Figura 1.

Respuesta del circuito RLC ante un escalón



El sobre impulso (OV) es del 72.4% y el tiempo de establecimiento es de 9.35ms.

2.1.2.3 Controlador IOPID para el circuito RLC serie. Se ajustaron los parámetros del controlador PID clásico (K_p , K_i , K_d) utilizando el optimizador de SISOTOOL fijando como requerimientos el tiempo de establecimiento (T_s) y el sobre impulso (OV). Los requerimientos fueron: $T_s < 7\text{ms}$ y $OV < 40\%$.

La herramienta permite una variedad de algoritmo de optimización como lo son simplex search, gradiente descendiente, pattern search, entre otras. Mediante varias pruebas se obtuvo que el algoritmo que sintonizaba el controlador que cumplía con ambos requerimientos fue Simplex Search, dando el siguiente controlador:

$$C_{IOPID}(s) = \frac{0.001522(s + 1184)(s + 2569)}{s}$$

Con el controlador entero anterior, el sistema en lazo cerrado se comporta de la siguiente forma:

Figura 2.

Respuesta del circuito RLC con un control de orden entero

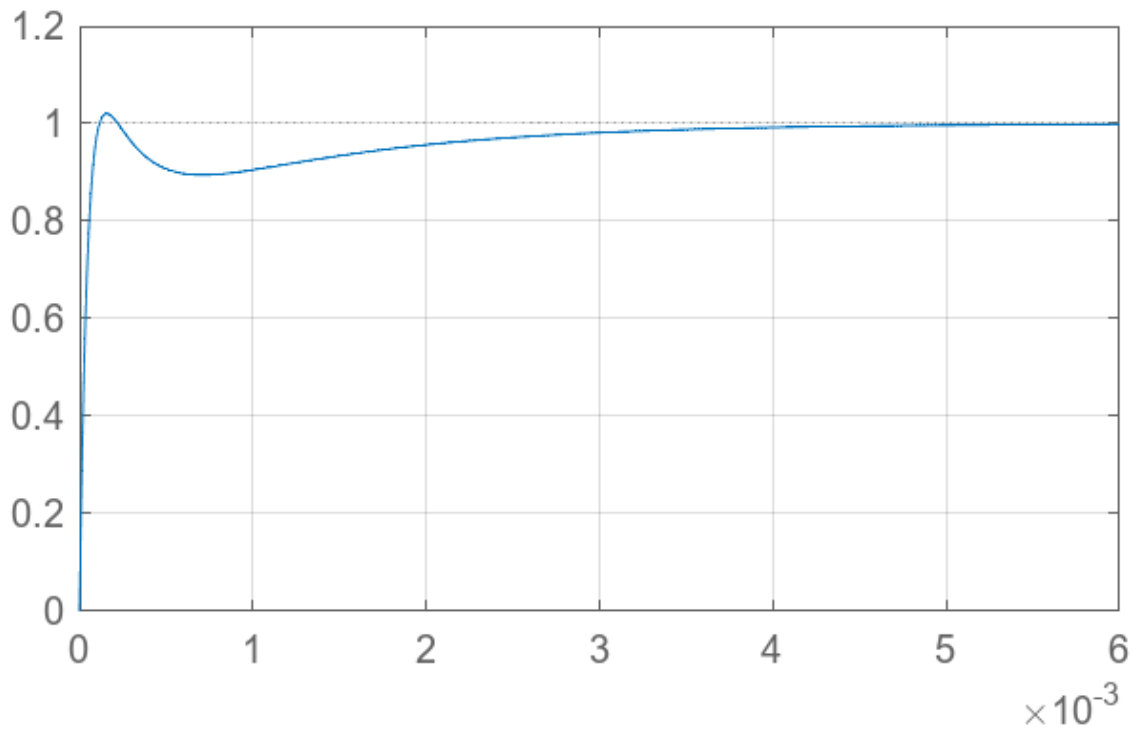
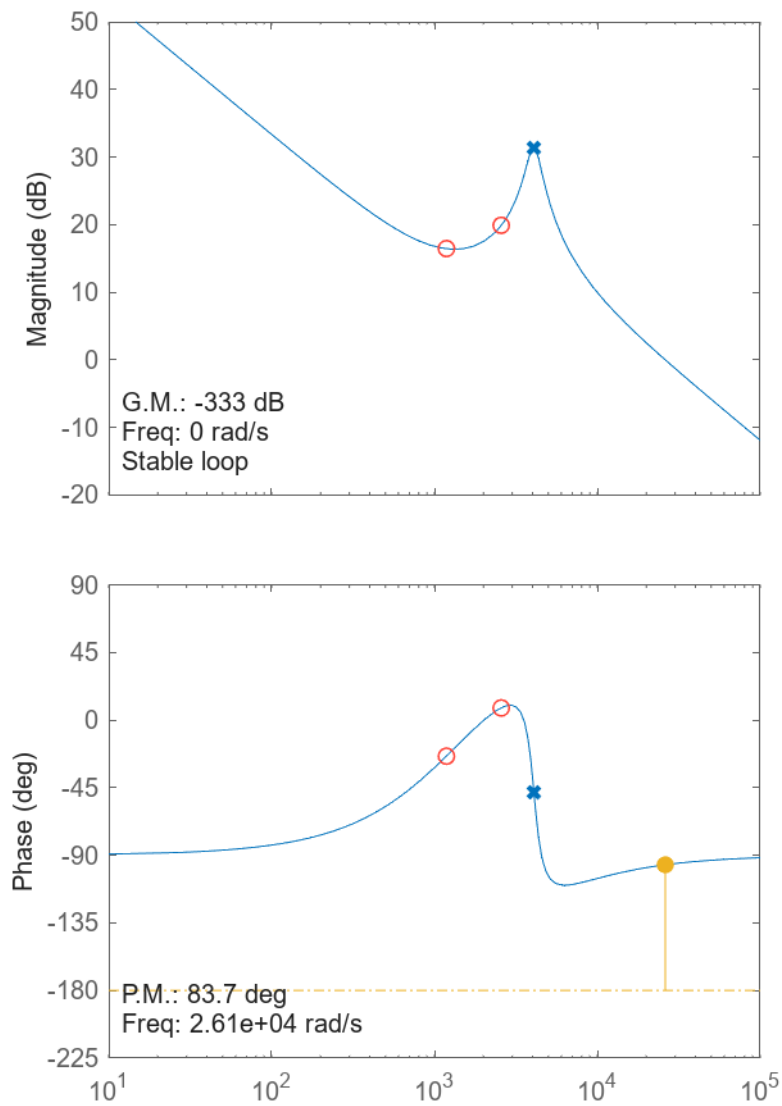


Figura 3.

Diagramas de ganancia y fase del circuito RLC controlado con un IOPID



La figura 2 muestra la respuesta temporal de un circuito RLC serie controlado mediante un PID entero. El sistema presenta un sobre impulso moderado del 2.06%, lo que indica una

respuesta ligeramente subamortiguada, pero bien contenida. El tiempo de establecimiento es de aproximadamente 2.91 ms, lo que demuestra una rápida estabilización alrededor del valor de referencia. Esta respuesta es adecuada para aplicaciones donde se requiere rapidez sin comprometer la estabilidad.

En la figura 3 se observan los diagramas de Bode del sistema en lazo abierto. El margen de fase es de 83.7° a una frecuencia de 2.61×10^4 rad/s, lo cual indica una excelente robustez frente a perturbaciones y variaciones paramétricas. El margen de ganancia aparece como muy negativo (-333 dB), lo cual sugiere que el cruce de ganancia no se alcanza dentro del rango analizado, pero la estabilidad sigue garantizada. En conjunto, el sistema PID entero logra una respuesta rápida, con baja oscilación y buena estabilidad de fase, aunque sin la flexibilidad adicional que ofrecen controladores fraccionarios.

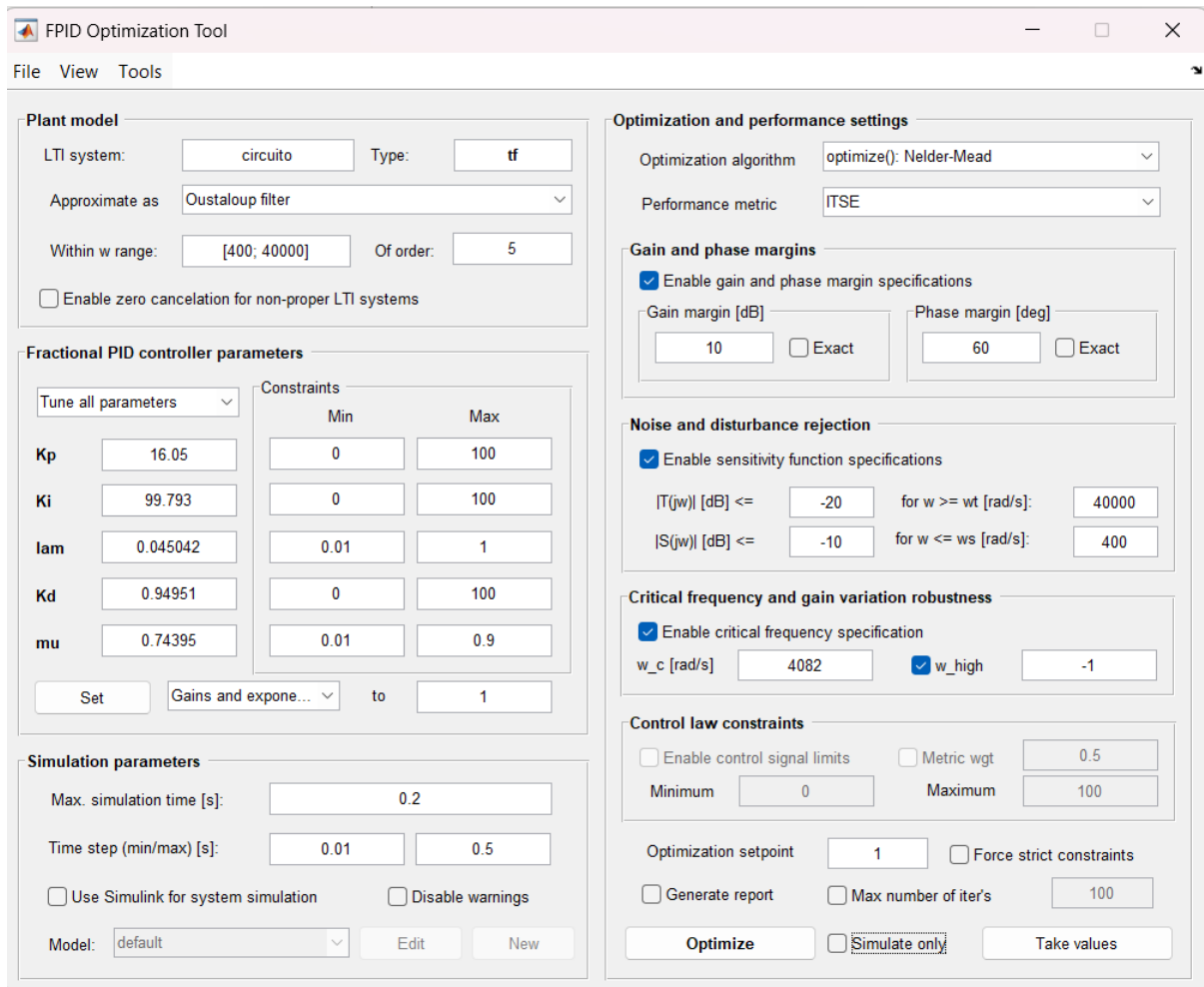
2.1.2.4 Controlador FOPID para el circuito RLC serie. En la configuración mostrada en la interfaz de la herramienta FPID Optimization Tool, se está utilizando un modelo de planta definido como una función de transferencia bajo la representación de filtro de Oustaloup de orden 5, dentro del rango de frecuencias [400,40000] rad/s. Esto permite aproximar los términos fraccionarios de manera precisa en una banda centrada alrededor de la frecuencia de resonancia del sistema RLC (4082 rad/s). El algoritmo de optimización elegido es Nelder-Mead, y se está utilizando el ITSE (Integral of Time-weighted Squared Error) como métrica de rendimiento, que penaliza errores persistentes y favorece una rápida estabilización.

Respecto a las restricciones, se han activado condiciones sobre los márgenes de ganancia y fase (10 dB y 60° respectivamente) para asegurar robustez frente a incertidumbres del modelo. También se limita la función de sensibilidad $|S(j\omega)| \leq -10$ dB para frecuencias bajas

(perturbaciones) hasta 400 rad/s y la función complementaria $|T(j\omega)| \leq -20$ dB para frecuencias altas (ruido) a partir de 40000 rad/s. Además, se habilita una restricción crítica de frecuencia en la resonancia del sistema, $\omega_c=4082$ rad/s para mejorar la respuesta en esa zona. Los parámetros del controlador FOPID (K_p , K_i , K_d , λ , μ) están definidos dentro de límites razonables para permitir una exploración eficaz del espacio de soluciones durante la optimización.

Figura 4.

GUI del optimizador de la toolbox FOMCON



Una vez realizada la optimización el controlador obtenido fue:

$$C_{FOPID}(s) = 16.05 + \frac{99.793}{s^{0.045042}} + 0.94951 * s^{0.74395}$$

La grafica de comportamiento del sistema controlado se observa en la figura 5, mientras que las gráficas de margen de fase y de ganancia se observan en la figura 6. También las gráficas de $T(j\omega)$ y $S(j\omega)$ se pueden ver en las figuras 7 y 8 respectivamente.

Figura 5.

Comportamiento en el tiempo del circuito con un FOPID

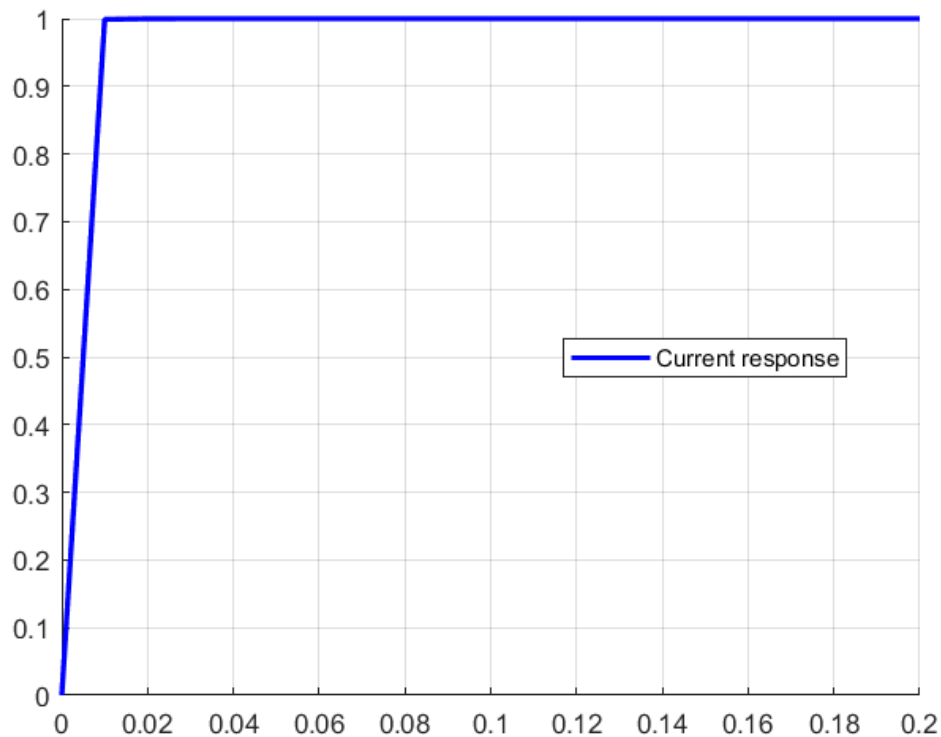
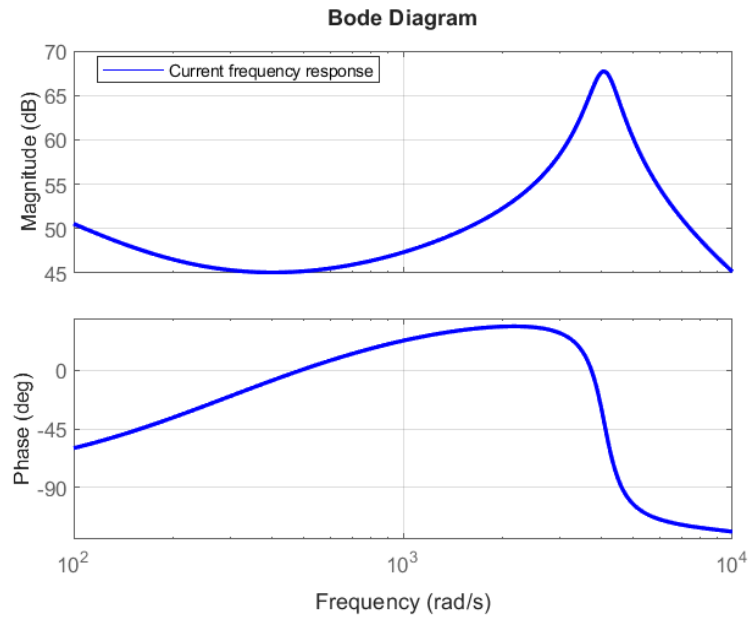


Figura 6.

Diagrama de margen de ganancia y de fase del circuito con un FOPID

**Figura 7.**

Diagramas de bode de $S(j\omega)$ del circuito con un FOPID

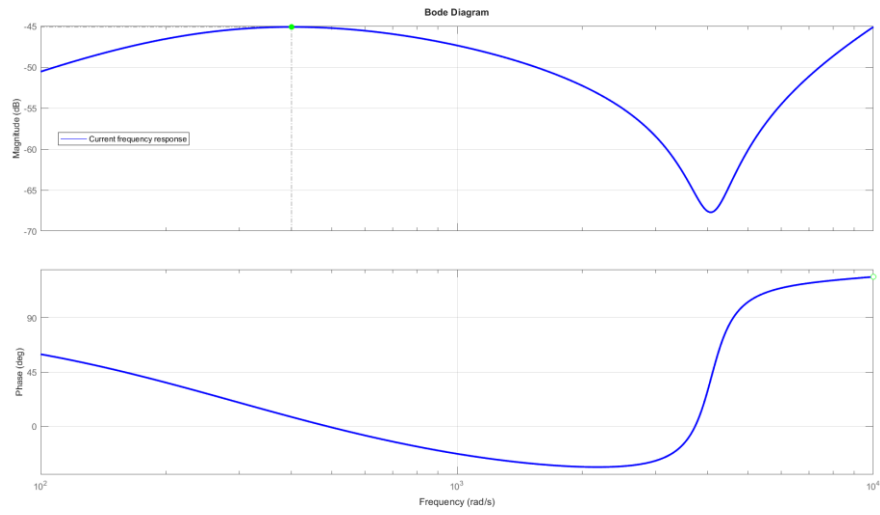
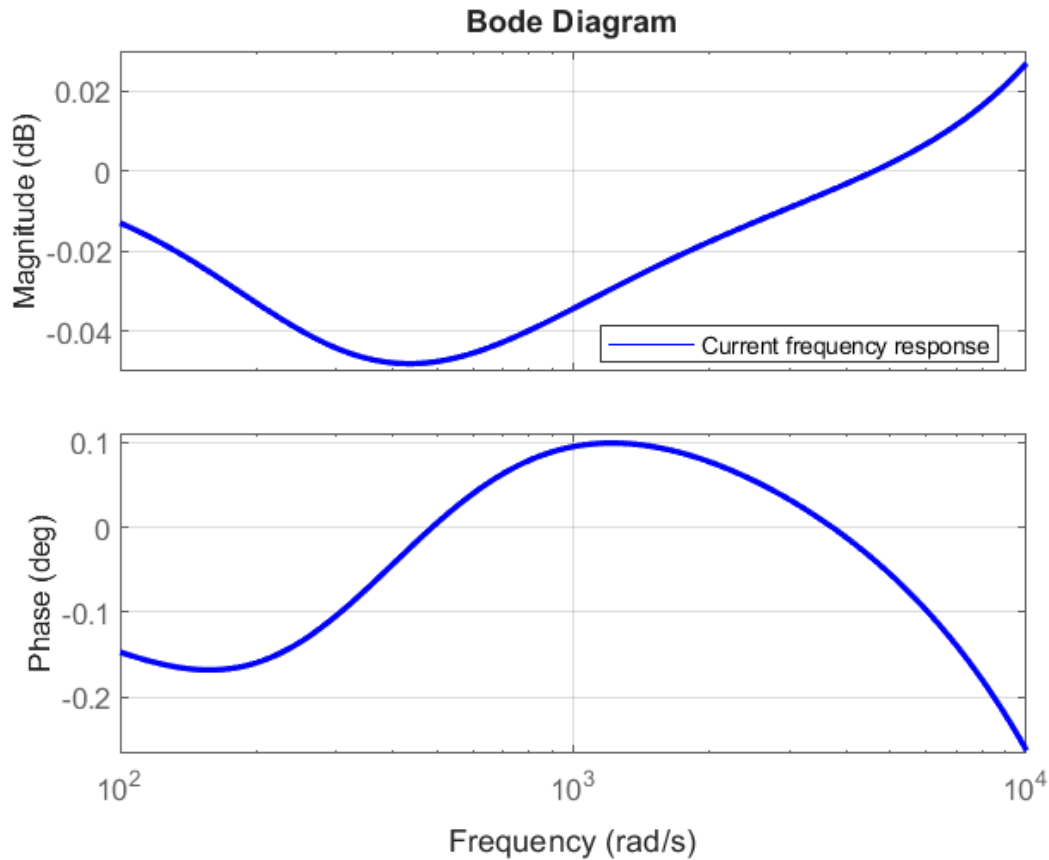


Figura 8.

Diagramas de bode de $T(j\omega)$ del circuito con un FOPID



La figura 5 muestra la respuesta temporal del sistema ante una entrada escalón unitario. La señal alcanza el valor de referencia (amplitud 1) en un tiempo extremadamente corto, alrededor de 20 milisegundos. Este comportamiento sugiere una disminución en la velocidad de respuesta. El sistema no presenta sobre impulso visible ni oscilaciones, lo que refleja una muy buena estabilidad y un amortiguamiento adecuado.

La figura 6 representa los diagramas de Bode de ganancia (magnitud) y fase del sistema controlado. Se puede observar una resonancia clara en la magnitud en torno a los 4000 rad/s, que es coherente con la frecuencia de resonancia del circuito RLC. Este pico se encuentra alrededor de 68 dB, lo cual implica una ganancia significativa a dicha frecuencia. La fase muestra un comportamiento típico de sistemas de segundo orden, con un incremento progresivo que

atraviesa los 0° y se aproxima a -90° para frecuencias altas. Este tipo de respuesta en frecuencia indica que el sistema mantiene una ganancia aceptable y una fase relativamente estable dentro de un rango amplio de frecuencias, lo que es esencial para mantener el margen de fase y evitar inestabilidades. En conjunto, esta respuesta en Bode sugiere un sistema bien sintonizado, con un margen de ganancia infinito (ganancia siempre positiva) y un margen de fase no definido (esto indica un sistema que está sobre amortiguado).

El diagrama de Bode correspondiente a la función de sensibilidad $S(j\omega)$, que representa cómo el sistema responde ante perturbaciones externas y errores, indica un excelente rechazo a perturbaciones de baja frecuencia. Esto se aprecia porque la magnitud de $S(j\omega)$ está por debajo de -10 dB para frecuencias inferiores a aproximadamente 400 rad/s. Esto significa que el sistema es muy eficaz en eliminar errores de seguimiento y rechazar perturbaciones lentas (como variaciones térmicas lentas o interferencias de baja frecuencia). En frecuencias medias y altas, la ganancia se incrementa progresivamente, lo que es esperado en sistemas con buena capacidad de seguimiento, pero limitada capacidad de rechazo en alta frecuencia. Sin embargo, el aumento en la ganancia no llega a comprometer la estabilidad gracias al diseño del controlador FOPID.

Por último, el diagrama de Bode de la función de sensibilidad complementaria $T(j\omega)$, que refleja cómo el sistema transmite la señal de referencia a la salida, muestra una baja magnitud en altas frecuencias (por debajo de 0 dB), lo cual es positivo porque indica que el sistema filtra el ruido de alta frecuencia que pudiera estar presente en la referencia o en la señal de control. En el rango medio de frecuencias, la magnitud de $T(j\omega)$ es cercana a 0 dB, lo que implica que el sistema sigue la referencia con precisión sin amplificarla ni atenuarla excesivamente. Esta

característica es importante en sistemas donde se requiere alta fidelidad en el seguimiento de la señal de referencia sin introducir distorsión ni ruido.

2.1.2.5 Modelado del sistema de temperatura. Para el modelado del sistema de temperatura, que consta de un Arduino Uno(controlador), un MOSFET IRFZ44N y un sensor LM35. La conexión es la siguiente:

Tabla 1.

Configuración del sistema de temperatura

Componente	Descripción
LM35 (sensor temperatura)	Vcc → 5V del Arduino, GND → GND del Arduino, Vout al A0 del Arduino. Sensor de temperatura analógico (10 mV/°C).
MOSFET IRF544N	Gate a la resistencia de 220 Ω desde D3. Drain a la resistencia de carga. Source al GND del Arduino. Transistor de potencia (controlado por PWM).
Resistencia Gate (220 Ω)	Arduino D3 Gate del MOSFET. Protección del pin y control de conmutación.
Resistencia Pull-down (10 kΩ)	Gate del MOSFET al GND. Evita que el Gate quede flotando. Mantiene el MOSFET apagado si PWM está en 0.
Resistencia de carga	Un extremo a +5 V Arduino, Otro extremo al Drain del MOSFET. Debe disipar al menos 5 W. Genera el calor que será controlado.

Se inyectó con el Arduino un PWM con un DC de 98%, se capturaron los datos por medio de Simulink (ver figura 9). también se hizo un promedio exponencial de media móvil con una alfa de 0.05 para suavizar el ruido de la señal y una breve programación para la comunicación serial entre Arduino y Simulink (ver figura 10), todo esto se hizo con un tiempo de muestreo de 125ms y por un tiempo de 300 segundos se tomaron los datos.

Figura 9.

Diagrama de bloques para la adquisición de datos del sistema de temperatura

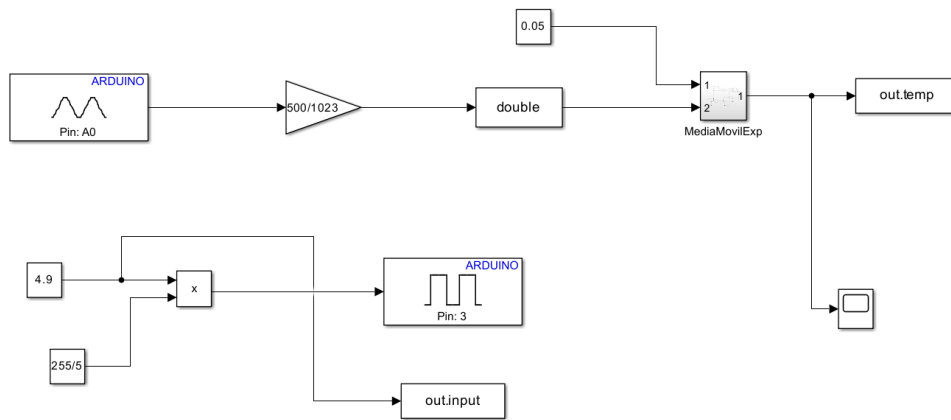


Figura 10.

Programación del Arduino para comunicarse con Simulink

```

1  const int pinSensor = A0; // Entrada analógica (LM35)
2  const int pinOut = 3;    // Salida PWM
3
4  uint8_t out = 0;
5  uint8_t in = 0;
6
7  void setup() {
8    Serial.begin(115200); // Velocidad de comunicación con MATLAB
9    pinMode(pinOut, OUTPUT);
10 }
11
12 void loop() {
13   // Leer sensor analógico
14   out = analogRead(pinSensor); // 0 - 1023
15
16   // Enviar valor por serial (1 byte)
17   Serial.write(out);
18
19   // Leer PWM deseado desde MATLAB (1 byte)
20   if (Serial.available()) {
21     in = Serial.read(); // Valor PWM 0 - 255
22     analogWrite(pinOut, in);
23   } else {
24     analogWrite(pinOut, 0); // Sin dato, apaga salida
25   }
26
27   delay(125); // Tiempo de muestreo = 0.125 s
28 }
29

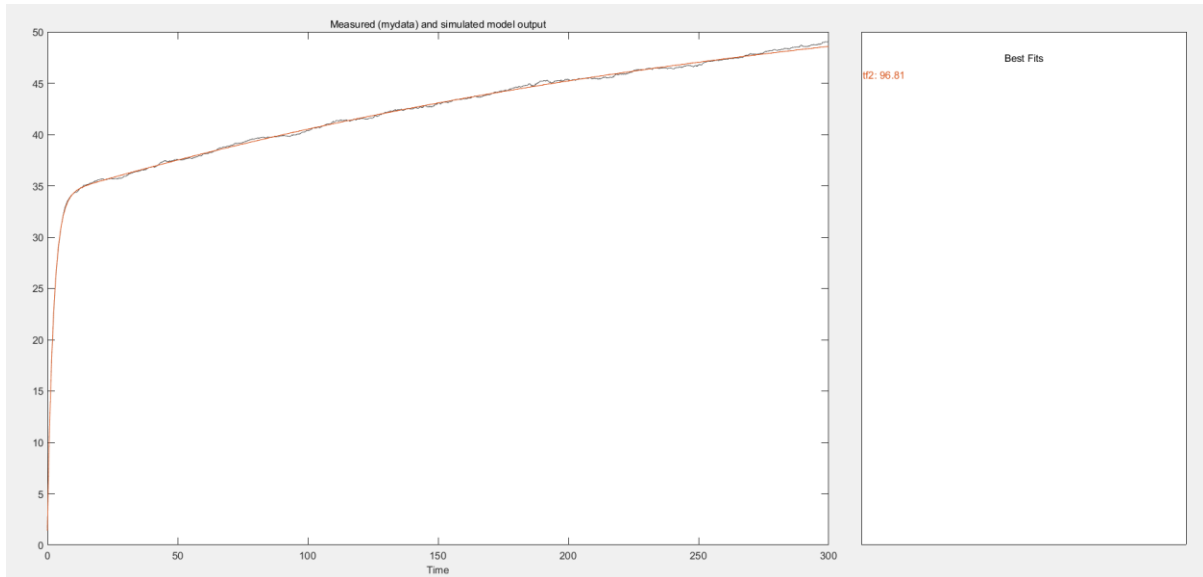
```

Una vez con los datos adquiridos se usó la herramienta “System Identification” de Matlab para hallar la planta del sistema. La mejor caracterización (96.81% de precisión) la dió una planta con un cero y dos polos, todo esto en el dominio de Z ya que se hará un control digital con el Arduino, la ecuación de la planta es:

$$G_{Temp}(z) = \frac{0.000256z^{-1}}{1 - 1.947z^{-1} + 0.9471z^{-2}}$$

Figura 11.

Gráfica de la caracterización de la planta usando System Identification



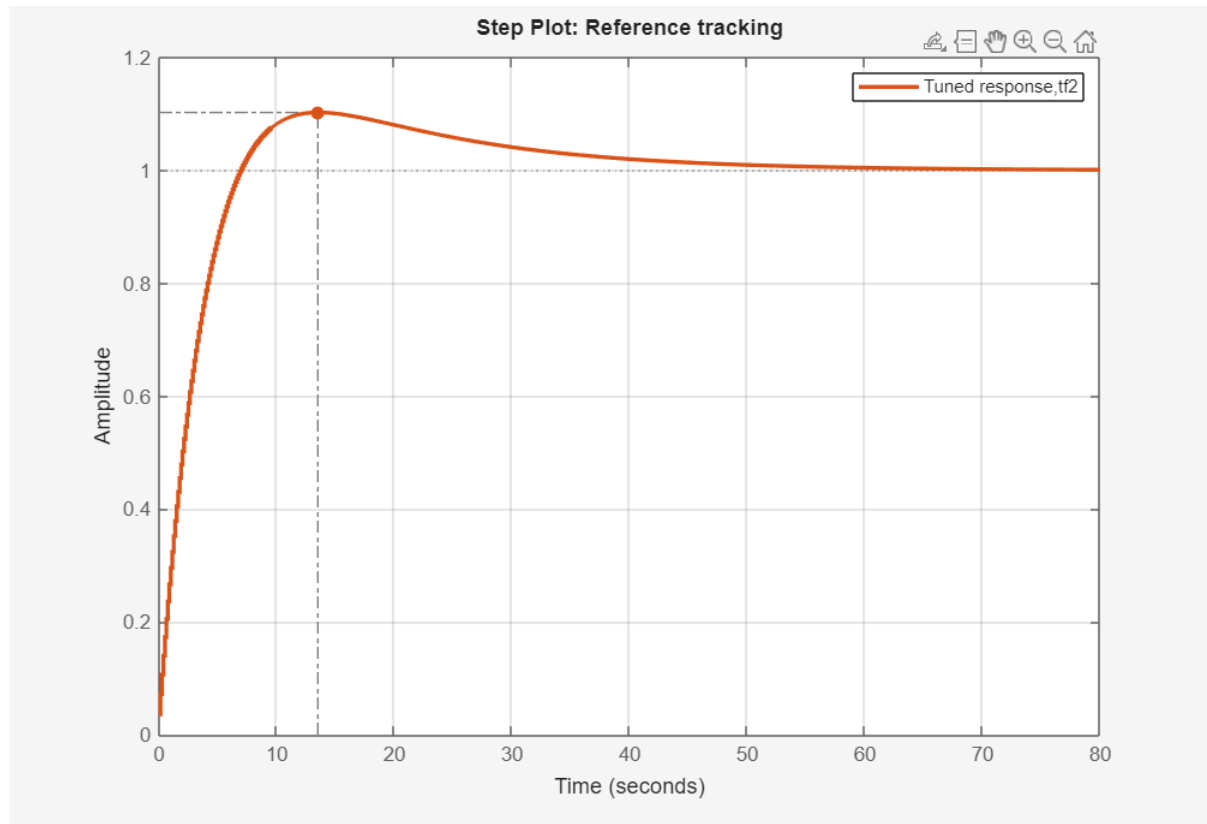
2.1.2.6 Controlador IOPID para el sistema de temperatura.

$$C_{IOPID_{TEMP}}(z) = 9.4601 + \frac{0.47981 * T_s}{z - 1} + 18.506 * \left(\frac{z - 1}{T_s * z}\right)$$

Ts denota el tiempo de muestreo que se tomó como 125ms, en simulación el sistema en lazo cerrado es como lo muestra la figura 12.

Figura 12.

Sistema de temperatura en lazo cerrado con IOPID sintonizado con PID Tuner

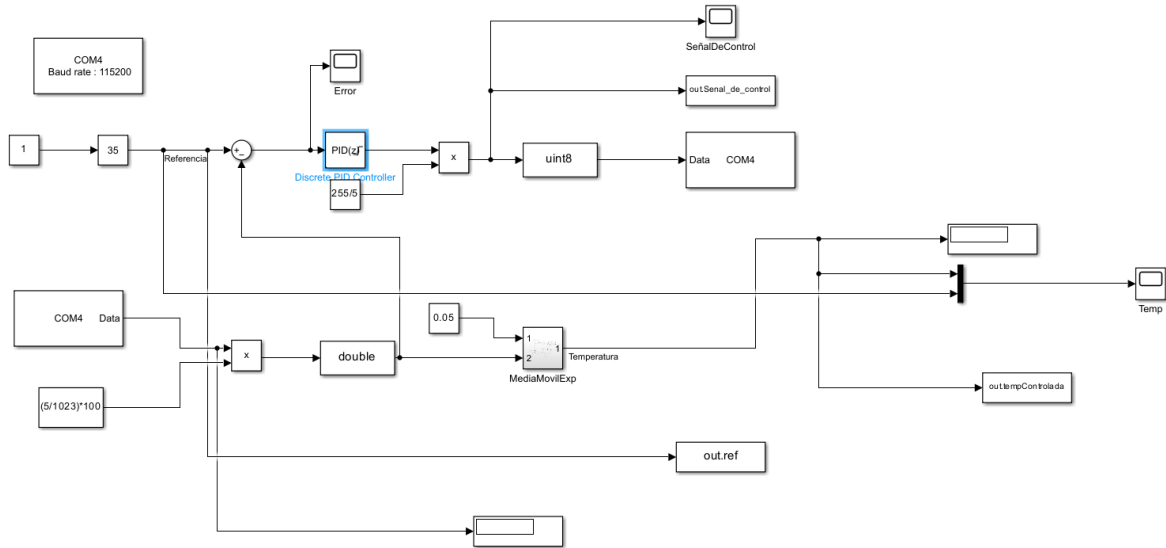


Con un OV de 10.3% y un tiempo de establecimiento de 40s.

Ahora en Simulink se hizo el respectivo diagrama de bloques (ver figura 13) para realizar el control del sistema.

Figura 13.

Diagrama de bloques para realizar la implementación de un IOPID.

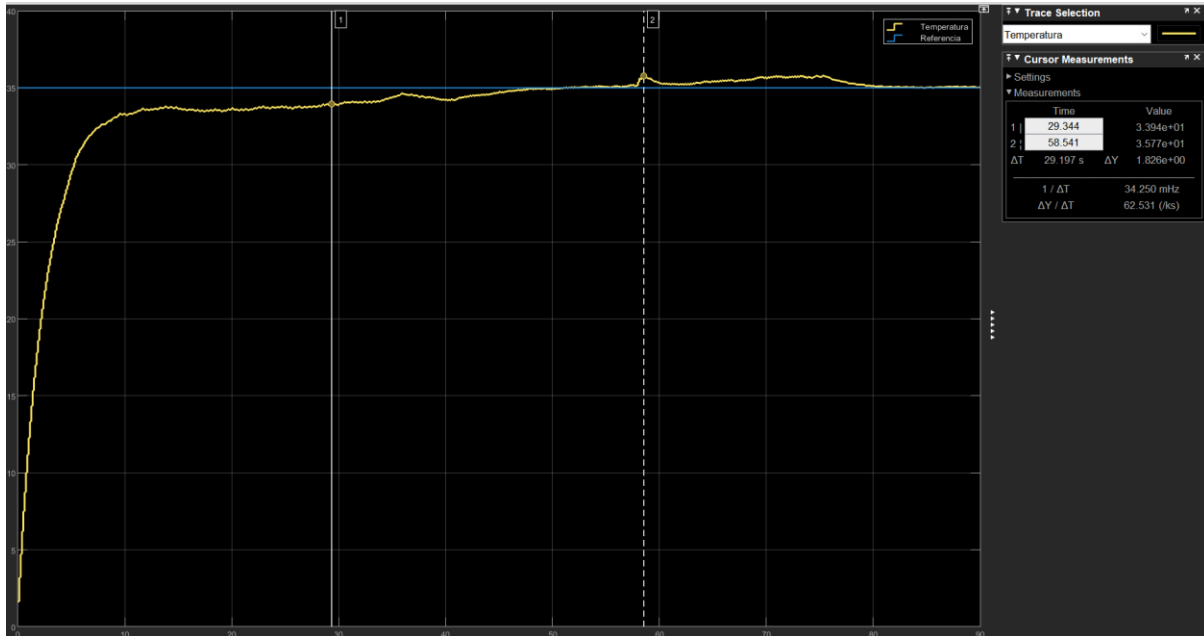


Para el Arduino se mantuvo el código que se implementó para la adquisición de datos.

Con esto el sistema de temperatura controlado es el mostrado en la figura 14. }

Figura 14.

Gráfica de la temperatura controlada con el IOPID



La grafica resultante tiene un OV menor al 1% y un tiempo de establecimiento de 29s, siguiendo el criterio del 3% (cuando sobrepasa el 97% y no sobrepasa el 103% de la temperatura deseada), la temperatura objetivo fue de 35°C.

Las diferencias observadas entre la simulación y la implementación física se deben principalmente a las limitaciones de modelado, efectos no lineales no contemplados, retardo de medición del sensor LM35 y ruidos ambientales. Adicionalmente, existe la posibilidad que la herramienta “PID Tuner” de MATLAB no sea tan rigurosa en algunos aspectos del controlador.

2.1.2.7 Controlador FOPID para el sistema de temperatura. El diseño del controlador FOPID para el sistema térmico siguió el mismo proceso de modelado y sintonización fraccionaria usando FOMCON y Simulink. Se implementó la simulación con la aproximación de

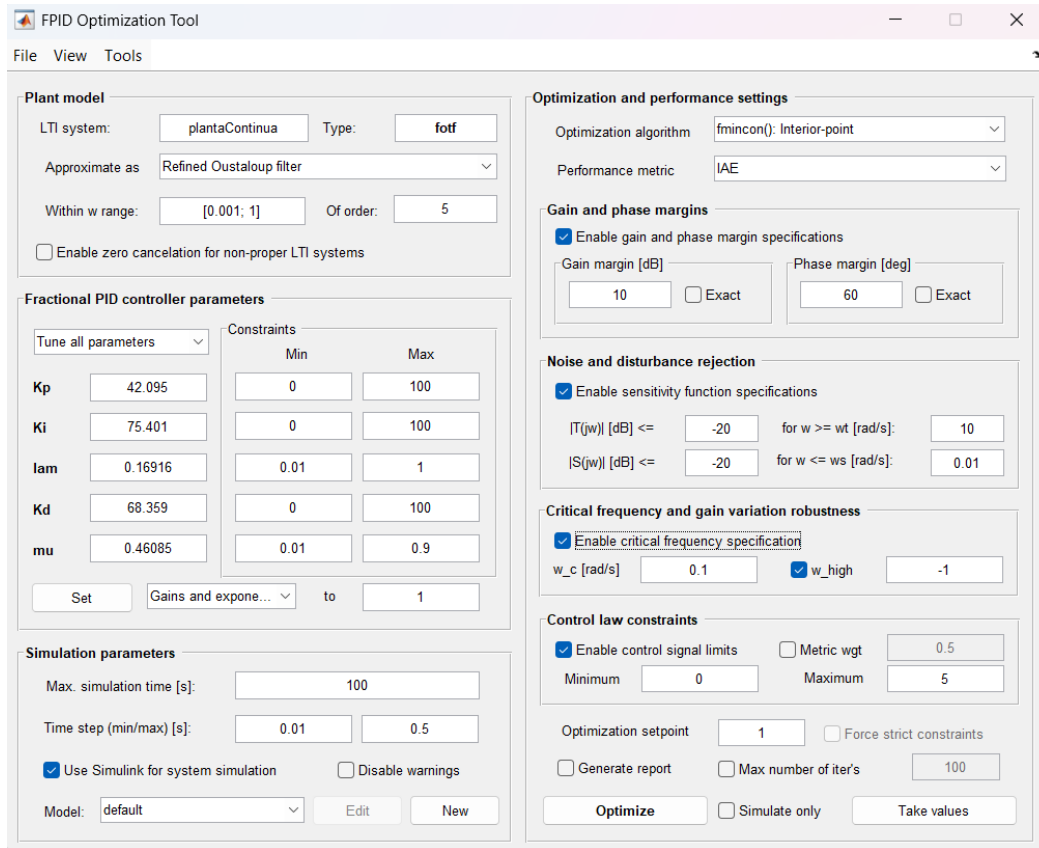
Oustaloup y se compararon los resultados con la planta física para evaluar la fidelidad del modelo y la efectividad del controlador fraccionario.

Para este caso, se usó un rango de frecuencias de $[0.001, 1]$ rad/s en el filtro refinado de Oustaloup, ya que los sistemas térmicos, como el control de temperatura de un MOSFET, son lentos y no necesitan responder a frecuencias altas. El diseño del controlador FOPID se basó en limitar ciertas funciones importantes del sistema: la sensibilidad $S(j\omega)$, que ayuda a rechazar perturbaciones, y la sensibilidad complementaria $T(j\omega)$, que se encarga del seguimiento de la referencia. Se buscó que el sistema rechazara bien perturbaciones lentas (con $S \leq -20$ dB) y que no reaccionara demasiado al ruido (con $T \leq -20$ dB a frecuencias altas). Además, se pidió que el sistema fuera robusto, con márgenes de ganancia y fase amplios (mínimo 10 dB y 60° respectivamente), para que siguiera funcionando bien incluso si el comportamiento real variaba un poco.

También se consideró importante minimizar el IAE, que mide cuánto error se acumula con el tiempo. Esto es clave en sistemas térmicos donde se quiere que la temperatura llegue bien al valor deseado sin muchos vaivenes. Para evitar problemas en el hardware, se limitó la señal de control entre 0 y 5, lo que encaja bien con el PWM del Arduino. Con todo esto, el controlador FOPID final no solo controla bien la temperatura, sino que también es estable, robusto y cuida el sistema físico. Finalmente, el algoritmo se escogió con base en prueba y error (ver figura 15).

Figura 15.

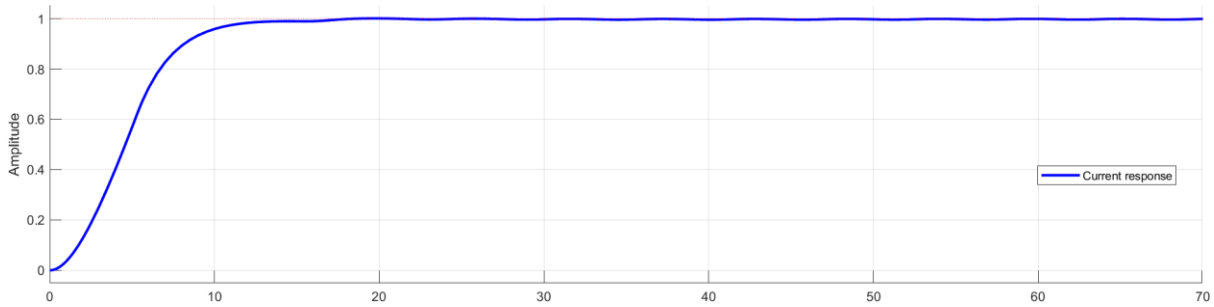
Configuración de la GIU FOMCON para hallar el FOPID.



$$C_{FOPID}(s) = 42.095 + \frac{75.401}{s^{0.16916}} + 68.359 * s^{0.46085}$$

Figura 16.

Temperatura controlada en simulación con FOMCON

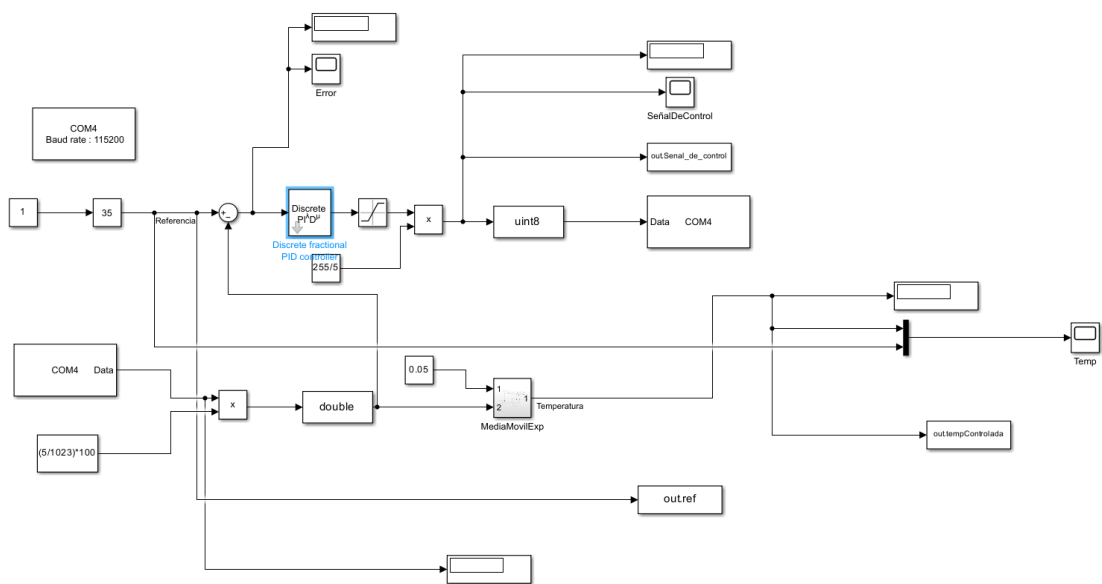


En la figura 11 se observa que el $OV=0\%$ y el tiempo de establecimiento es de 20s aproximadamente.

Para la implementación se hizo uso del diagrama de bloques de la figura 17.

Figura 17.

Diagrama de bloques para realizar la implementación de un FOPID

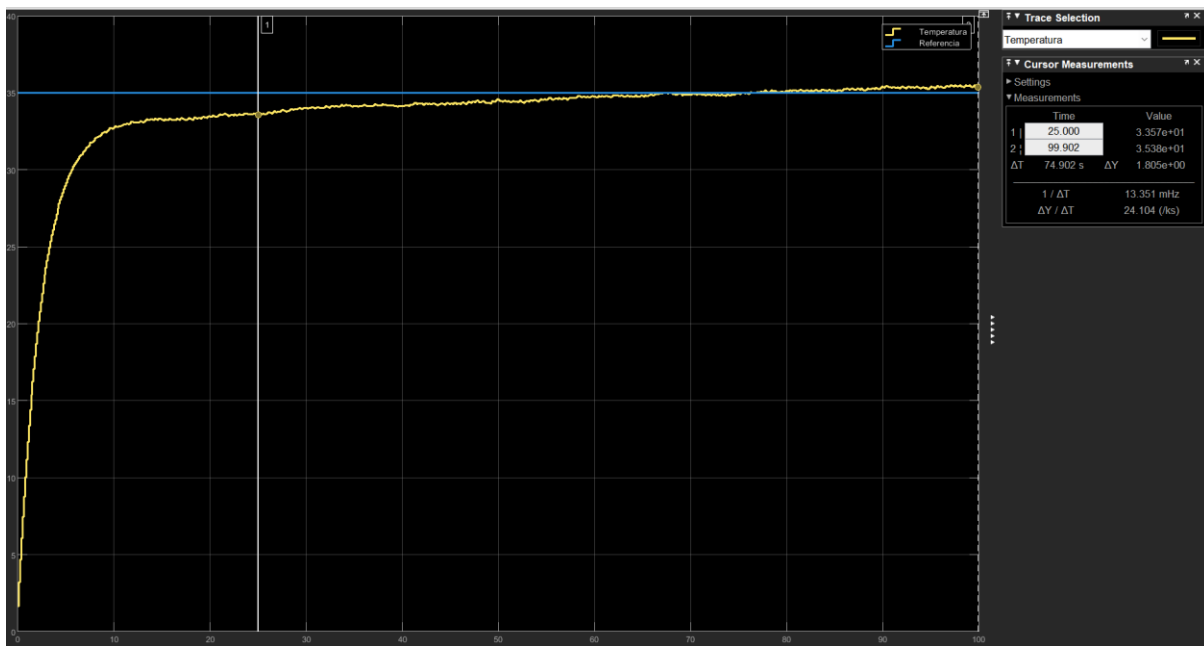


En el método de discretización del bloque “Discrete PID Controller” se usó el método de Tustin, esto debido a su facilidad para saltar del dominio continuo al discreto. Y para la comunicación entre Arduino y Simulink se mantuvo el código de Arduino anteriormente mencionado.

En la figura 18 se observa el experimento y su resultado.

Figura 18.

Gráfica de la temperatura controlada con un FOPID



Para este caso el $OV=0\%$ y el tiempo de establecimiento es de 25s siguiendo el mismo criterio del 3%, la temperatura objetivo también fue de 35°C . En este caso la implementación y la simulación se asemejan bastante, esto muestra un alto grado de confiabilidad en la herramienta FOMCON.

2.1.3 Resultados.

Para analizar de forma clara los resultados, se incluyen tablas comparativas de las principales métricas de interés:

Circuito RLC Serie

Tabla 2.

Comparativa comportamientos control del circuito RLC

Controlador	Tiempo de establecimiento(ms)	Sobre impulso (%)	Error en estado estacionario(%)	Margen de fase (°)	Margen de ganancia(dB)
PID	2.91	2.06	0	83.7	-333
FOPID	20	0	0	No definido	∞

Sistema de temperatura

Tabla 3.

Comparativa comportamientos control del sistema térmico

Controlador	Tiempo de Establecimiento(s)	Sobreimpulso (%)	Error Estacionario (%)
PID (Simulado)	40	10.3	0
PID (Físico)	29	<1%	0
FOPID (Simulado)	20	0	0
FOPID (Físico)	25	0	0

2.1.3.1 Discusión. Los resultados obtenidos permiten analizar de manera comparativa el desempeño de los controladores de orden entero (PID clásico) y de orden fraccionario (FOPID) en los dos sistemas físicos estudiados: el circuito RLC serie y el sistema de temperatura.

En el caso del circuito RLC, el FOPID mostró un comportamiento más robusto en términos de rechazo de perturbaciones y seguimiento de la referencia, tal como lo evidencian los diagramas de Bode. La ausencia de sobreimpulso y la estabilidad alcanzada confirman la capacidad del control fraccionario para adaptarse a sistemas con dinámicas rápidas. Sin embargo, se observó un aumento en el tiempo de establecimiento en comparación con el PID clásico, lo que sugiere que, si bien el FOPID ofrece ventajas en robustez y estabilidad, su desempeño puede estar condicionado por la velocidad de respuesta requerida en ciertas aplicaciones.

En el sistema térmico, el FOPID logró reducir el tiempo de establecimiento y eliminar el sobreimpulso en comparación con el PID entero. La coincidencia entre los resultados de simulación e implementación física refuerza la validez del modelo matemático y de la metodología de sintonización empleada. Estos hallazgos resaltan la aplicabilidad del FOPID en sistemas caracterizados por alta inercia y retardos, donde la precisión en la respuesta y la robustez del control son aspectos críticos.

En conjunto, los resultados confirman que el FOPID constituye una alternativa prometedora frente al PID clásico, al ofrecer mayor flexibilidad en la sintonización y un mejor equilibrio entre estabilidad, precisión y rechazo de perturbaciones. Estas características lo posicionan como una estrategia de control con alto potencial para aplicaciones reales en automatización, electrónica y sistemas embebidos, especialmente en escenarios donde se requiere un compromiso entre velocidad de respuesta y robustez.

3. Conclusiones

- El controlador FOPID superó al PID clásico en el circuito RLC en algunas características, logrando un sobreimpulso de 0% pero un tiempo de establecimiento más grande. Los diagramas de Bode mostraron una buena sintonización en frecuencia, con robustez en la

resonancia (~ 4000 rad/s), excelente rechazo a perturbaciones $S(j\omega)$ y seguimiento preciso de la referencia $T(j\omega)$. Esto confirma su efectividad en sistemas dinámicos rápidos.

- En el sistema térmico, el FOPID redujo el tiempo de establecimiento (25 s frente a 29 s del PID) y eliminó el sobreimpulso. La coincidencia entre simulación e implementación respalda la precisión del modelo y la sintonización lograda con FOMCON y el filtro de Oustaloup, incluso en sistemas con alta inercia y retardos.
- Los resultados muestran que el control FOPID mejora notablemente la respuesta dinámica, la robustez y la precisión frente al PID tradicional. Su mayor flexibilidad lo hace ideal para diversos sistemas físicos, consolidándolo como una alternativa viable para aplicaciones reales en automatización, electrónica y sistemas embebidos.

4. Recomendaciones

El desarrollo de este proyecto permitió evaluar la aplicación de controladores PID de orden entero y fraccionario en dos tipos de sistemas físicos: un circuito RLC serie modelado y simulado, y un sistema de temperatura modelado, simulado e implementado físicamente. Sin embargo, durante el proceso surgieron limitaciones y aprendizajes que abren la puerta a futuras líneas de trabajo.

Una primera recomendación es dar continuidad a la implementación física del circuito RLC serie, que en esta ocasión se abordó solo a nivel de simulación debido a restricciones relacionadas con la disponibilidad de componentes de alta frecuencia y la dificultad de sincronizar la respuesta del hardware con la velocidad de muestreo del microcontrolador

disponible. En futuros trabajos sería pertinente explorar tecnologías o plataformas de adquisición más rápidas para cerrar esta brecha.

Adicionalmente, se recomienda profundizar en el análisis de métodos de aproximación fraccionaria, especialmente para la implementación física de controladores FOPID. El uso del método de Oustaloup resultó eficiente en simulación, pero su traducción a código embebido para microcontroladores de bajo costo presenta desafíos por la complejidad computacional. Explorar aproximaciones simplificadas, filtros digitales optimizados o incluso soluciones basadas en control adaptativo fraccionario podría mejorar la viabilidad práctica.

En cuanto al sistema de temperatura, una línea interesante sería analizar la influencia de condiciones ambientales variables, la calibración de sensores y la optimización de algoritmos de compensación para aumentar la precisión del control. Asimismo, se podría extender la investigación a otros tipos de procesos térmicos industriales que requieran control multivariable o distribuido.

De forma transversal, se sugiere promover la colaboración interdisciplinaria con otros grupos de investigación en control fraccionario, automatización e instrumentación. Establecer vínculos con empresas del sector energético o de climatización permitiría aplicar estos desarrollos en contextos reales, validando su impacto y escalabilidad.

Finalmente, se hace énfasis en la importancia de considerar los aspectos éticos, de seguridad y normativos cuando se implementan sistemas de control con potencial impacto físico. Garantizar que los dispositivos de medición y actuadores cumplan estándares de seguridad y calibración es fundamental para prevenir riesgos y asegurar resultados confiables

Referencias Bibliográficas

- Monje, C.A., Chen, Y.Q., Vinagre, B.M., Xue, D., & Feliu, V. (2010). Fractional-order Systems and Controls: Fundamentals and Applications. Springer-Verlag.
- Podlubny, I. (1999). Fractional-order systems and $PI^\lambda D^\mu$ -controllers. IEEE Trans. on automatic control.
- Tepljakov, A., Petlenkov, E., & Belikov, J. (2011). FOMCON: Fractional-order modeling and control toolbox for MATLAB. Proceedings MIXDES.
- Xue, D., Chen, Y.Q., & Atherton, D.P. (2007). Linear Feedback Control Analysis and Design with MATLAB. SIAM.

Apéndices

Apéndice A

Control IOPID circuito RLC serie.

Los archivos relacionados se encuentran en la ruta:

<https://drive.google.com/drive/folders/181Y7ctuUeNW8N4B4M55ozNo8043wet0P?hl=es>

Apéndice B

Control FOPID circuito RLC serie.

Los archivos relacionados se encuentran en la ruta:

<https://drive.google.com/drive/folders/1FXcJpVKY6pQMBB1PLmKuPI5DHuRrRgeL?hl=es>

Apéndice C

Diagrama para la adquisición de datos para el sistema de temperatura.

Archivo: https://drive.google.com/file/d/1l1ui6d1NB4deDjDq-ufqBWOJ57g6Dy6c0/view?usp=drive_link

Apéndice D

Datos adquiridos del sistema de temperatura.

Archivo:

https://drive.google.com/file/d/15WhzEw5mN9Nb3AaU_Ufydz9OVfCmBYCq/view?usp=drive_link

Apéndice E

Programación del Arduino para transmisión y recepción de datos.

Archivo: https://drive.google.com/file/d/1EiaVC8Zi5WdbhyvTwcs-7WC1mkZt65Bd/view?usp=drive_link

Apéndice F

Identificación de la planta del sistema de temperatura.

Archivo: https://drive.google.com/file/d/1jloOP0ACfZbp04rgqZAx9rYhbJ0-acvE/view?usp=drive_link

Apéndice G

Diagrama de bloques implementación IOPID.

Ruta:

https://drive.google.com/file/d/1Q9Yg9P0RD3BPQKPLRe2kQyhmvoPacEq4/view?usp=drive_link

Apéndice H

Configuración de FOMCON para sintonización del FOPID.

Archivo:

https://drive.google.com/file/d/1NHNMV3aw5911XoZrHkvSlsTBPj5P8ia/view?usp=drive_link

Apéndice I

Diagrama de bloques implementación FOPID.

Archivo:

https://drive.google.com/file/d/1H1T8OnqcyZv6pYjFF2Wbq88Yy70zIZ46/view?usp=drive_link