

Simulación Hardware-in-the-Loop y Control en Tiempo Real de un Reactor de Tanque Continuamente Agitado basado en Arduino

An Arduino-based platform for Hardware-in-the-Loop simulation and control of Continuous Stirred-Tank Reactor

Héctor Daniel Hidalgo-Leal

Instituto tecnológico Superior de Villa La Venta, México

E-mail: hector.hl@laventa.tecnm.mx ORCID: <https://orcid.org/0000-0001-9028-8711>

Fabiola Cruz-Gutiérrez

Instituto tecnológico Superior de Villa La Venta, México

E-mail: faby.cg@laventa.tecnm.mx ORCID: <https://orcid.org/0000-0003-2577-7265>

Luis David May-Burgos

Instituto tecnológico Superior de Villa La Venta, México

E-mail: luis.mb@laventa.tecnm.mx ORCID: <https://orcid.org/0000-0001-9588-685X>

Guadalupe Hernández-Hernández

Instituto tecnológico Superior de Villa La Venta, México

E-mail: guadalupehh.1@gmail.com ORCID: <https://orcid.org/0009-0005-5192-6422>

Resumen

Los métodos tradicionales de validación para sistemas de control de reactores de tanque continuamente agitado (CSTR, por sus siglas en inglés) presentan un compromiso crítico. Las plataformas comerciales de simulación en tiempo real (HIL, por sus siglas en inglés) son costosas, mientras que las simulaciones de software puro carecen de realismo de hardware. Este trabajo presenta una alternativa de HIL de bajo costo basada en Arduino que utiliza dos microcontroladores interconectados: uno que simula la dinámica del CSTR en tiempo real y el otro que implementa un algoritmo de control de retroalimentación. Para validar la fidelidad, el sistema HIL se compara con un modelo de referencia de MATLAB bajo parámetros de control y escenarios de perturbación idénticos. Los resultados demuestran una desviación despreciable en el seguimiento de la temperatura entre las simulaciones de Arduino y MATLAB. La arquitectura propuesta mantiene la operación en tiempo real a bajas tasas de muestreo con una baja carga computacional, lo que la hace viable tanto para laboratorios educativos como para el prototipado industrial.

Palabras Clave

CSTR, Arduino, simulación en tiempo real, HIL, controlador digital.



Abstract

Traditional validation methods for Continuous Stirred Tank Reactor (CSTR) control systems face a critical trade-off. Commercial Hardware-in-the-Loop (HIL) platforms are prohibitively expensive, while pure software simulations lack hardware realism. This work presents a low-cost Arduino-based HIL alternative using two interconnected microcontrollers, one simulating the CSTR dynamics in real-time and the other implementing a feedforward-feedback control algorithm. To validate fidelity, HIL system is compared against a MATLAB reference model under identical control parameters and disturbance scenarios. Results demonstrate negligible deviation in temperature tracking between the Arduino HIL and MATLAB simulations. The proposed architecture maintains real-time operation at low sampling rates with negligible computational burden, making it viable for both educational laboratories and industrial prototyping.

Keywords

CSTR, Arduino, real-time simulation, HIL testing, digital controller.

Zusammenfassung

Herkömmliche Validierungsmethoden für Steuerungssysteme von kontinuierlich gerührten Kesselreaktoren (CSTR) stehen vor einem kritischen Kompromiss. Kommerzielle Hardware-in-the-Loop (HIL)-Plattformen sind unerschwinglich teuer, während reine Softwaresimulationen nicht die nötige Hardware-Realität aufweisen. Diese Arbeit stellt eine kostengünstige Arduino-basierte HIL-Alternative mit zwei miteinander verbundenen Mikrocontrollern vor, von denen einer die CSTR-Dynamik in Echtzeit simuliert und der andere einen Feedforward-Feedback-Steuerungsalgorithmus implementiert. Zur Validierung der Genauigkeit wird das HIL-System unter identischen Steuerungsparametern und Störszenarien mit einem MATLAB-Referenzmodell verglichen. Die Ergebnisse zeigen eine vernachlässigbare Abweichung in der Temperaturverfolgung zwischen den Arduino-HIL- und MATLAB-Simulationen. Die vorgeschlagene Architektur ermöglicht Echtzeitbetrieb bei niedrigen Abtastraten mit vernachlässigbarem Rechenaufwand und ist daher sowohl für Ausbildungslabore als auch für die industrielle Prototypenentwicklung geeignet.

Schlüsselwörter

CSTR, Arduino, Echtzeitsimulation, HIL-Test, digitaler Controller.



1. Introducción

El CSTR es uno de los procesos más representativos y estudiados en el ámbito de la ingeniería química y de control (Anbu & Senthilkumar, 2018; Tantawy et al., 2019). Su comportamiento dinámico no lineal, junto con su capacidad para operar en estado estacionario o transitorio bajo distintas condiciones de entrada y perturbaciones, lo convierten en un modelo de referencia para el análisis, diseño e implementación de estrategias de control. Además, su estructura relativamente simple, pero abundante en fenómenos dinámicos como la retroalimentación cruzada entre variables de estado, lo hacen especialmente útil en contextos académicos y de investigación como banco de pruebas para metodologías de identificación de sistemas, sintonización de controladores y validación de algoritmos robustos o adaptativos.

En paralelo, el avance en plataformas de bajo costo como Arduino ha abierto nuevas oportunidades para el desarrollo de prototipos en tiempo real mediante técnicas de HIL (Biswas et al., 2017). Esta estrategia permite evaluar el comportamiento de algoritmos de control implementados en hardware físico sin necesidad de contar con la planta real, reduciendo significativamente los riesgos, costos y tiempos asociados al desarrollo de sistemas de control. La simulación HIL se ha consolidado como una herramienta clave tanto en entornos industriales como educativos, al permitir la experimentación con sistemas complejos de forma segura, repetible y económica (Gaur, 2020; Zhang et al., 2024).

Si bien se han reportado diversas aplicaciones de HIL utilizando microcontroladores para sistemas eléctricos, térmicos y mecánicos, existe una notable ausencia de trabajos enfocados en procesos químicos con comportamiento no lineal, como el CSTR. Esta carencia representa una oportunidad para extender el uso de plataformas embebidas hacia procesos tradicionalmente simulados en software especializado, promoviendo la comprensión y validación de controladores en tiempo real con recursos limitados.

En este contexto, el presente trabajo propone la implementación de un modelo dinámico de un CSTR simulado en tiempo real sobre una plataforma Arduino Due, con el objetivo de servir como herramienta de experimentación para el diseño, análisis y



validación de estrategias de control mediante la técnica HIL. Esta contribución no solo busca mostrar la viabilidad técnica del enfoque, sino también su aplicabilidad en la formación académica y la experimentación didáctica en control de procesos no lineales.

2. Estado del arte o Marco Teórico

2.1. HIL en el control de sistemas dinámicos

La implementación segura y eficiente de controladores en procesos industriales que utilizan CSTR requiere pruebas exhaustivas antes de ser aplicados en la planta real. En este contexto, la simulación HIL se posiciona como una herramienta clave que permite evaluar controladores reales en combinación con modelos virtuales de planta, en un entorno de tiempo real.

Uno de los principales beneficios del enfoque HIL es que permite realizar pruebas controladas y repetibles sin poner en riesgo los equipos ni al personal. Según Hidalgo et al. (2024), la validación anticipada de estrategias de control mediante HIL reduce los tiempos de implementación y los costos asociados, al detectar errores en etapas tempranas del desarrollo. Este enfoque ha sido particularmente útil en sistemas de potencia y movilidad eléctrica, pero su aplicabilidad se extiende a procesos químicos con dinámicas complejas como el CSTR.

Por su parte, Mihalič et al. (2022) destacan que el uso de plataformas HIL no solo mejora la robustez del diseño de controladores, sino que también ofrece un entorno ideal para evaluar el comportamiento ante fallas y condiciones no contempladas, contribuyendo así a la robustez del sistema cerrado. Esta capacidad de simular escenarios extremos resulta especialmente útil en sistemas no lineales y altamente sensibles como los CSTR.

Además, (Lozada & Panchi, 2023) enfatizan el valor pedagógico y experimental del uso de HIL en el desarrollo de sistemas de control. La posibilidad de integrar el hardware del controlador real con modelos computacionales precisos permite una validación más cercana al comportamiento operativo del sistema físico, haciendo de HIL una herramienta formativa y profesional a la vez.



En el caso de un CSTR, donde la dinámica térmica y de concentración presenta características no lineales y acopladas, el uso de HIL facilita la experimentación en tiempo real con algoritmos de control avanzados (como PID, MPC o μ -síntesis), sin comprometer la seguridad ni requerir el acceso constante a la planta. Por tanto, la simulación HIL no solo garantiza una transición más segura y efectiva hacia la implementación física, sino que también permite iterar rápidamente sobre diferentes configuraciones de control.

2.2. Arduino DUE

Los microcontroladores se caracterizan por incorporar arquitecturas basadas predominantemente en conjuntos de instrucciones reducidos (RISC, por sus siglas en inglés), con anchos de palabra que varían entre 8 y 32 bits, lo que permite alcanzar un equilibrio entre eficiencia computacional y simplicidad en el diseño del hardware (Sozański, 2023). Estos dispositivos operan comúnmente a frecuencias de reloj que oscilan entre los 16 MHz, como los microcontroladores AVR utilizados en plataformas como Arduino, hasta los 240 MHz en arquitecturas más avanzadas como el ESP32. Esta variabilidad en capacidad de procesamiento permite seleccionar microcontroladores adecuados según la complejidad del algoritmo de control, la frecuencia de muestreo requerida por el sistema, y las necesidades específicas de comunicación o procesamiento en tiempo real. Su bajo costo, facilidad de integración y disponibilidad de entornos de desarrollo accesibles los hacen ideales para validar estrategias de control en plataformas HIL educativas y experimentales, especialmente en sistemas dinámicos como el CSTR.

Las tarjetas Arduino se caracterizan por su flexibilidad y estandarización, empleando microcontroladores ATmega328P o SAMD21 con frecuencias de 8 MHz hasta 84 MHz. Los Arduino más utilizados operan a 5 V, con salidas digitales para modulación por ancho de pulso (PWM, por sus siglas en inglés), entradas analógicas 10 bits, y buses UART, I²C y SPI (Guerreiro et al., 2014; Izotov et al., 2021). Su conectividad incluye módulos compatibles y USB para programación/comunicación. La alimentación admite USB, voltajes de alimentación externa de hasta 12 volts. La memoria varía desde 16 KB de Flash hasta 2 MB, con SRAM limitada de 2 KB hasta 256 KB, restringiendo aplicaciones intensivas en datos. El



ecosistema de programación del IDE Arduino esta basado en lenguaje C++ con el apoyo de librerías de terceros.

El Arduino DUE por su parte utiliza un microcontrolador SAM3X8E que incorpora un procesador ARM®Cortex®-M3 con arquitectura Harvard. Además, incluye una Unidad Aritmético Lógica (ALU) de 32 bits (Melin et al., 2022). Por otro lado, es importante mencionar que el DUE cuenta con 54 pines Entrada/Salida, con las siguientes características:

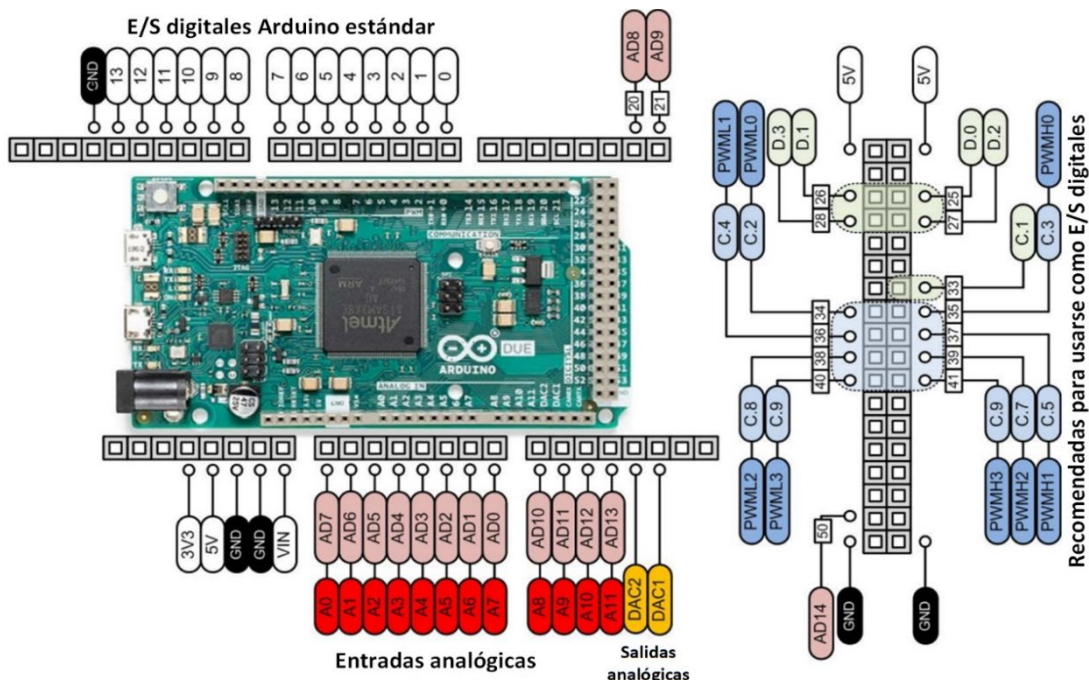
- Contador temporizador (TC) de 9 canales y 32 bits para captura, comparación y modo PWM.
- Convertidor analógico-digital (ADC) de 16 canales y 12 bits de aproximación sucesiva (SAR) a 1 MSPs (multiplexado), uno de ellos reservado para el sensor de temperatura interno.
- Seis controladores de entrada/salida en paralelo de 32 bits.
- Convertidor PWM con salida complementaria (PWMC) de 8 canales y 16 bits y generador de tiempo muerto de 12 bits.
- Convertidor digital-analógico (DAC, por sus siglas en inglés) de dos canales y 12 bits a 1 MBPs (multiplexado).

DUE presenta compatibilidad con todos los módulos diseñados para Arduino que operen a un voltaje de 3.3 V. Este modelo incorpora un microcontrolador basado en un resonador de cuarzo de 84 MHz, lo que garantiza una alta precisión en el control de temporización. Además, cuenta con dos interfaces USB: una dedicada exclusivamente a la programación del dispositivo y otra configurable como USB Host, permitiendo la conexión con periféricos externos. En la Figura 1 se detalla la disposición de los puertos de E/S, así como la distribución de los pines de alimentación.

Figura 1

Distribución de los puertos de Arduino DUE.





Nota. Adaptada de (Melin et al., 2022).

2.3. Reactor de Tanque Continamente Agitado (CSTR)

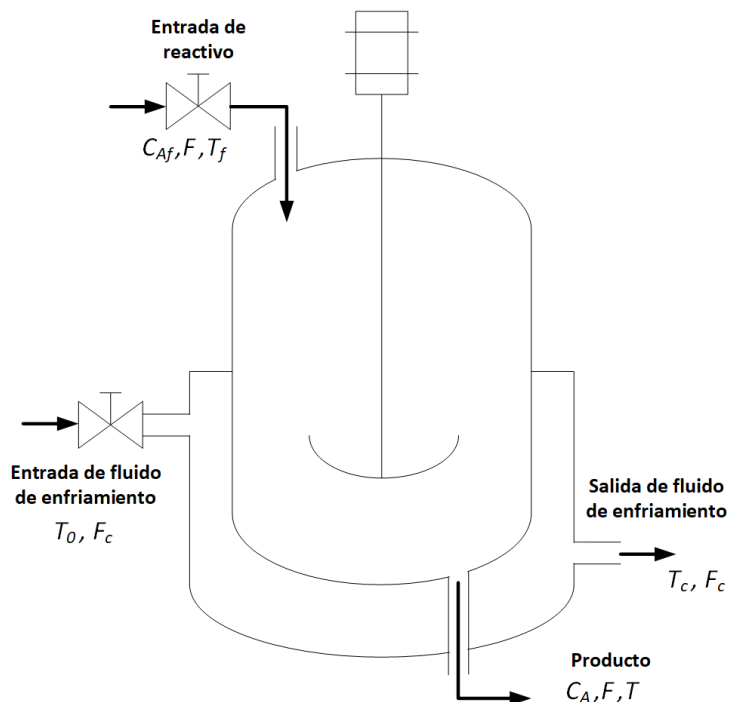
El CSTR es ampliamente utilizado en la industria química, farmacéutica y biotecnológica debido a su capacidad para mantener condiciones de operación constantes mediante agitación continua y flujo estable. Entre sus aplicaciones más comunes se encuentra la fermentación, donde es empleado en la producción de etanol, antibióticos y enzimas, al proporcionar un entorno uniforme para el crecimiento microbiano (Kumar et al., 2019). También se utiliza en reacciones de síntesis química, como la fabricación de polímeros, detergentes o fertilizantes, aprovechando su capacidad de control cinético (Loo Chavez, 2023). En procesos de descomposición o degradación, como el tratamiento biológico de aguas residuales, el CSTR permite una mezcla eficaz de contaminantes y agentes reactivos (Hurtado et al., 2015). Asimismo, es útil en reacciones de neutralización ácido-base, permitiendo ajustes precisos de pH en flujos continuos. Aunque no es el ideal para procesos de separación, también puede participar en precipitación y cristalización controlada bajo condiciones específicas. Su versatilidad, la facilidad con la que se puede implementar la medición y compatibilidad con técnicas de monitoreo lo convierten en un



modelo ideal para el desarrollo, simulación y validación de sistemas de control avanzados, incluidos esquemas basados en HIL. El CSTR que se estudia en este trabajo tiene la estructura que se muestra en la figura 2.

Figura 2

Representación de un Reactor de Tanque continuamente agitado CSTR.



Nota. Fuente: elaboración propia.

El CSTR es un reactor químico que tiene el objetivo de controlar la temperatura o concentración de los reactivos. La ecuación general de balance de materia del reactor es

$$\rho \frac{dV}{dt} = F_{in}\rho_{in} - F_{out}\rho, \quad (1)$$

donde V es el volumen del reactor, ρ es la densidad del fluido dentro del reactor y ρ_{in} es la densidad del fluido en la entrada del reactor. F_{in} y F_{out} son los flujos de entrada y salida en el reactor, respectivamente. Por simplicidad se considera una reacción simple $A \rightarrow B$ de primer orden y demás $F_{in} = F_{out} = F$. De acuerdo al trabajo de Shi et al. (2023) el balance del reactivo A es



$$V \frac{dC_A}{dt} = F(C_{Af} - C_A) - Vr_A, \quad (2)$$

donde C_{Af} es la concentración en la entrada del reactor, C_A es la concentración del componente A en el reactor y r_A es la tasa de reacción por unidad de volumen. La ley de Arrhenius que describe la tasa de reacción se representa como

$$r_A = k_0 \exp\left(\frac{-E_a}{RT}\right) C_A, \quad (3)$$

donde k_0 es la constante de la tasa de reacción, E_a es la energía de activación, R es la constante universal de los gases ideales y T es la temperatura en el interior del reactor.

El balance de energía del reactor, asumiendo volumen, calor específico C_p , densidad constante ρ y despreciando los cambios en la energía potencial y cinética es

$$V\rho C_p \frac{dT}{dt} = F\rho C_p(T_f - T) + (-\Delta H)Vr_A - UA(T - T_c), \quad (4)$$

donde $-\Delta H$ es la entalpia de reacción, U es el coeficiente de transferencia de calor, A es el área de transferencia de calor, T_f es la temperatura del reactivo en la entrada del reactor y T_c es la temperatura dentro de la chaqueta de enfriamiento.

3. Materiales y métodos

La presente investigación se enmarca dentro de un enfoque cuantitativo, dado que se fundamenta en la formulación matemática del modelo dinámico no lineal de un CSTR, así como en el diseño, implementación y validación de estrategias de control a través de una arquitectura HIL basada en una plataforma Arduino. La obtención de datos se realiza mediante la ejecución de pruebas controladas en tiempo real, en las cuales se registra el comportamiento de variables clave del proceso, tales como la temperatura del reactor y la señal de control aplicada, bajo condiciones de operación definidas y una frecuencia de muestreo determinada. El análisis de dichos datos se lleva a cabo empleando herramientas computacionales, con el objetivo de evaluar indicadores cuantitativos de desempeño como el error en régimen permanente, el tiempo de establecimiento y la respuesta ante perturbaciones. La elección del enfoque cuantitativo obedece a la necesidad de contar con una metodología objetiva, reproducible y sistemática que permita validar la efectividad del

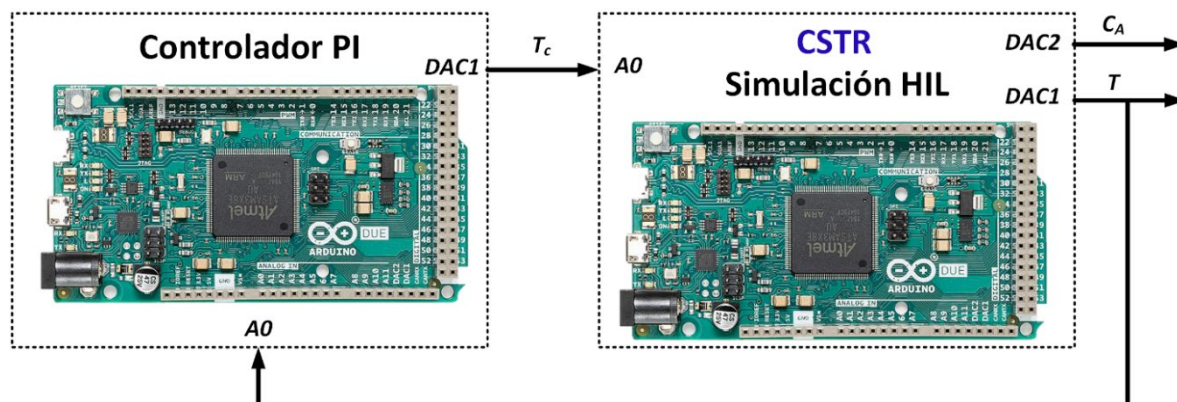


controlador propuesto antes de su aplicación en una planta física, garantizando así la estabilidad y la seguridad del proceso bajo estudio

3.1. Estructura HIL basada en Arduino

La implementación de la simulación en tiempo real se llevó a cabo utilizando dos placas Arduino Due, como se muestra en la figura 3. En esta arquitectura HIL, una de las placas desempeña la función de controlador, mientras que la otra simula el comportamiento dinámico del reactor CSTR. La señal de control T_c , generada por el controlador, se emite a través del pin DAC1 en un rango de voltaje comprendido entre 0.55 V y 2.75 V, lo cual corresponde a una temperatura de chaqueta entre 298 K y 303 K. Por su parte, la temperatura del reactor T , simulada por la planta virtual, también se transmite mediante una salida DAC, manteniendo el mismo rango de voltaje. La realimentación del sistema se logra mediante el pin analógico A0, tanto en el controlador como en la planta virtual, respetando un voltaje máximo de entrada de 3.3 V. Esta configuración permite cerrar el lazo de control en tiempo real utilizando señales analógicas, facilitando la validación del controlador bajo condiciones físicas representativas.

Figura 3
 Simulación en tiempo real y control.



Nota. Fuente: elaboración propia.

3.2. Solución numérica del CSTR

La simulación en tiempo real consiste en encontrar la solución numérica de las ecuaciones diferenciales (2) y (4). Para ello se usa un método numérico que el microcontrolador del DUE sea capaz de procesar. En este trabajo se considera el método de



Euler, que resulta apropiado para ecuaciones de primer orden. Dada la ecuación diferencial en espacio de estados:

$$\frac{dx}{dt} = f(x, u, t) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \quad (5)$$

donde t representa el tiempo, $x = [C_A, T]^T$ y $u = T_c$. Para entender la idea del método, se considera aproximar la deriva como

$$\frac{dx}{dt} \approx \frac{x(k+1) - x(k)}{T_s}, \quad (6)$$

donde k es el número de iteración y T_s es el tiempo entre cada iteración. Para cada iteración se pueden obtener valores aproximados uniformemente espaciados $k = 0, 1, 2, \dots, N$. Es importante mencionar que se requiere conocer el valor inicial $x(0)$.

3.3. Linealización y discretización del modelo CSTR

Para realizar el diseño del controlador se requiere del modelo lineal en el punto de operación del sistema. Esto debido a que la mayoría de los controladores industriales son lineales basados en métodos clásicos. Para obtener el modelo lineal se requiere del uso de series de Taylor y matrices Jacobianas. Utilizando la ecuación (5) de la forma

$$f(x, u, t) = \begin{bmatrix} \frac{F}{V}(C_{Af} - C_A) - r_A \\ \frac{F}{V}(T_f - T) + \frac{r_A}{\rho C_p}(-\Delta H) - \frac{UA}{\rho C_p V}(T - T_c) \end{bmatrix}$$

La forma linealizada alrededor del punto de operación (\bar{C}_A, \bar{T}) , se expresa en espacio de estados como

$$\begin{bmatrix} \delta \dot{C}_A \\ \delta \dot{T} \end{bmatrix} = \mathbf{A}_l \begin{bmatrix} \delta C_A \\ \delta T \end{bmatrix} + \mathbf{B} T_c, \quad (7)$$

$$\delta T = \mathbf{C} \begin{bmatrix} \delta C_A \\ \delta T \end{bmatrix},$$

donde $\delta C_A = C_A - \bar{C}_A$, $\delta T = T - \bar{T}$, la matriz de estado es

$$\mathbf{A}_l = \begin{bmatrix} \frac{\partial f_1}{\partial C_A} & \frac{\partial f_1}{\partial T} \\ \frac{\partial f_2}{\partial C_A} & \frac{\partial f_2}{\partial T} \end{bmatrix} = \begin{bmatrix} -\frac{F}{V} - \bar{r}_A & -\bar{C}_A k_T \\ \frac{(-\Delta H)}{\rho C_p} \bar{r}_A & -\frac{F}{V} - \frac{UA}{\rho C_p V} + \frac{(-\Delta H)}{\rho C_p} \bar{C}_A k_T \end{bmatrix},$$

considerando que $\bar{r}_A = k_0 \exp\left(\frac{-E_a}{RT}\right)$ y $k_T = \bar{r}_A \left(\frac{E_a}{RT^2}\right)$, la matriz de entrada es



$$B = \begin{bmatrix} \frac{\partial f_1}{\partial T_c} \\ \frac{\partial f_2}{\partial T_c} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{UA}{\rho C_p V} \end{bmatrix}, \text{ la matriz de salida es } C = [0 \ 1].$$

Una vez definida la ecuación lineal en espacio de estados, la función de transferencia que relaciona a la entrada de control con la temperatura de salida, se obtiene mediante

$$G(s) = C(sI - A)^{-1}B. \quad (8)$$

Tomando los valores de la tabla 1, se puede obtener la función de transferencia anterior como

$$G(s) = \frac{2.092(s + 1.1337)}{s^2 + 6.2607s + 5.816}.$$

Dado que los microcontroladores operan en un entorno digital y ejecutan instrucciones de manera secuencial y en tiempo discreto, no es posible implementar directamente la ecuación (8). Por esta razón, se requiere transformar la representación en tiempo continuo del sistema $G(s)$ a su equivalente discreto $G(z)$. Aplicando la transformada bilineal (Tustin) con $T_s = 0.1$ segundos, se obtiene

$$G(z) = \frac{0.09975 + 0.01076z^{-1} - 0.089z^{-2}}{1 - 1.798z^{-1} + 0.8107z^{-2}} \quad (9)$$

3.4. Controlador para CSTR

Para garantizar la estabilidad del sistema y asegurar el seguimiento adecuado de la variable de salida a su valor de referencia, es necesario incorporar un esquema de control retroalimentado. En este contexto, se emplea un controlador proporcional-integral (PI), debido a su capacidad para eliminar el error en estado estacionario y mejorar la respuesta dinámica del sistema frente a perturbaciones o cambios de referencia. La ecuación que representa el controlador es expresa

$$T_c = K_p e + K_i \int e dt, \quad (10)$$

donde el error se define como $e = T_{ref} - T$, K_p es la ganancia proporcional y K_i es la ganancia integral. Sin embargo, al igual que la ecuación (8), no es posible implementarla de



forma directa en el DUE. Para implementar la ecuación (10), se emplea un integrador numérico que permite calcular el término integral del error, el cual se expresa como

$$e_i(k) = e_i(k - 1) + T_s e(k). \quad (11)$$

Por lo tanto, la ecuación del controlado PI adaptada para el DUE se expresa a continuación

$$T_c(k) = K_p e(k) + K_i e_i(k). \quad (12)$$

Es importante mencionar, que, aunque el controlador esta discretizado para hacer cálculos en cada muestra k , la frecuencia de muestreo del controlador puede ser diferente a la del sistema a regular.

Tabla 1
 Parámetros del CSTR.

Parámetro	Valor
F	$100.0 \text{ m}^3/\text{s}$
V	100.0 m^3
ρ	$1000.0 \text{ kg}/\text{m}^3$
C_p	$0.239 \text{ J}/\text{kg} \cdot \text{K}$
$-\Delta H$	$5.0 \times 10^4 \text{ J}/\text{mol}$
E_a/R	8750.0 K
k_0	$7.2 \times 10^{10} \text{ s}^{-1}$
UA	$5.0 \times 10^4 \text{ W}/\text{K}$
\bar{T}	324.475 K
\bar{C}_a	$0.87725 \text{ mol}/\text{m}^3$

Nota. Fuente: AP Monitor (2025).

4. Resultados

La simulación en tiempo real se presenta mediante tres escenarios: operación en lazo abierto, validación del modelo linealizado y evaluación en lazo cerrado utilizando el controlador propuesto. En cada escenario se comparan los resultados obtenidos con la

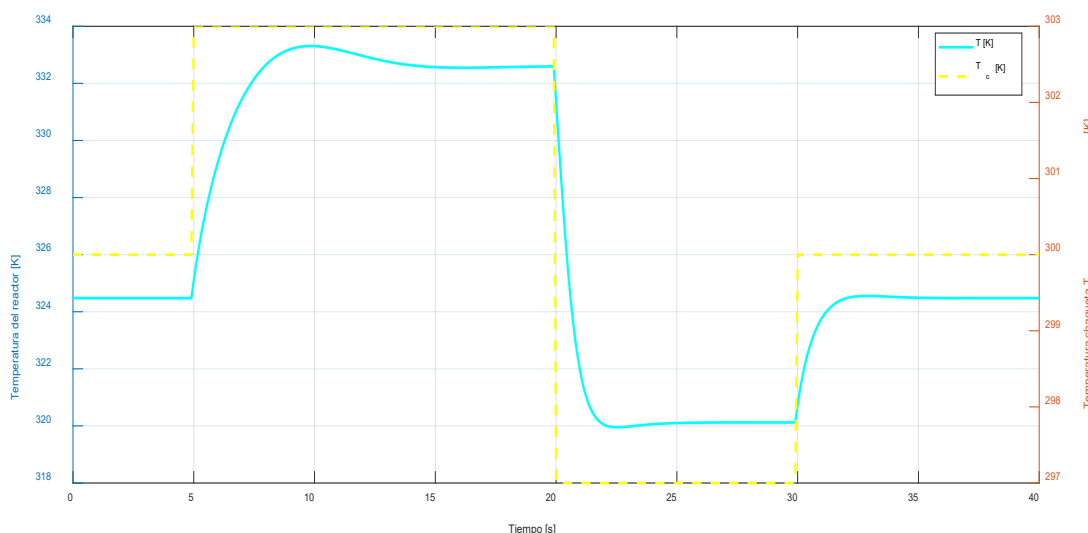


simulación en un software especializado. En el caso del presente trabajo se realiza la validación utilizando MATLAB.

Para analizar el comportamiento dinámico del CSTR en lazo abierto, se introdujo una perturbación en la entrada de control correspondiente a la temperatura de refrigeración, aplicando un doble escalón. En el tiempo $t = 5.0 s$, se realiza un cambio de 300.0 K a 303.0 K, posteriormente a los $t = 20.0 s$, la temperatura pasa de 303.0 K a 297.0 K. Por último, a los $t = 30.0 s$, la temperatura vuelve a los 300.0 K. La simulación en MATLAB del comportamiento del CSTR se aprecia en la figura 4.

Figura 4

Simulación en MATLAB de la respuesta en lazo abierto del CSTR.



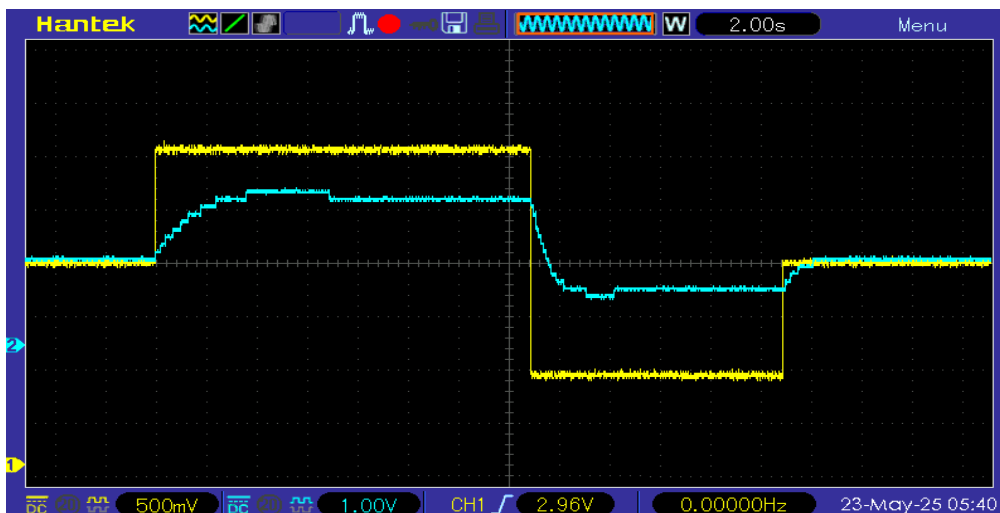
Nota. Fuente: elaboración propia.

Para llevar a cabo la simulación en tiempo real, se implementaron las ecuaciones (2) y (4) en el microcontrolador Arduino DUE, utilizando los parámetros físicos presentados en la Tabla 1. El sistema fue discretizado empleando el método de Euler con un intervalo entre cada iteración de $T_s = 0.1 s$, según lo indicado en la ecuación (6). Bajo estas condiciones, el comportamiento térmico del CSTR observado en el entorno HIL se muestra en la figura 5.

Figura 5

Simulación HIL de la respuesta en lazo abierto del CSTR.





Nota. La línea azul corresponde a la temperatura del CSTR y la línea amarilla corresponde a la temperatura de control. Fuente: Osciloscopio Hantek DSO5102P.

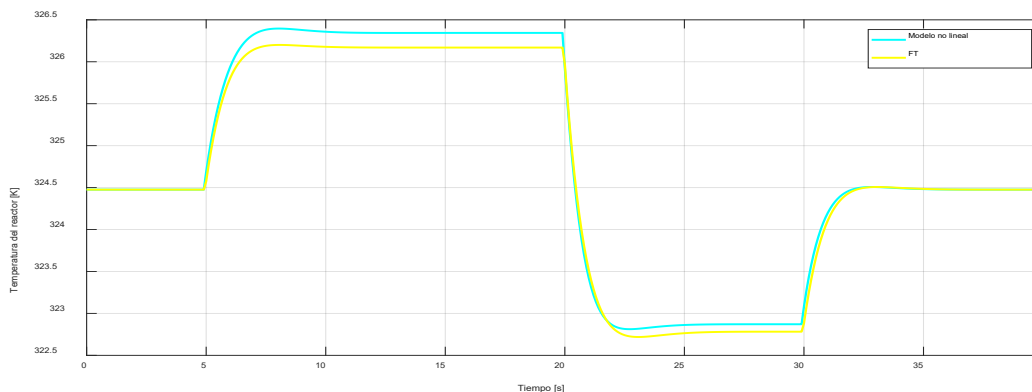
Cabe señalar que, debido a la magnitud de la variación térmica del reactor respecto a la temperatura de la chaqueta, se aplicaron escalas verticales diferenciadas para cada variable. La línea correspondiente a la temperatura del reactor se representa con una escala ampliada para permitir una visualización más precisa del fenómeno.

La simulación en tiempo real se realizó replicando las mismas condiciones que en el entorno de MATLAB, incluyendo las perturbaciones escalonadas aplicadas a la entrada de control T_c . La comparación entre ambas implementaciones permite validar la fidelidad del modelo embebido respecto al comportamiento dinámico simulado numéricamente.

El diseño del controlador parte de la necesidad de linealizar el modelo dinámico del proceso, con el fin de obtener una función de transferencia que represente adecuadamente el comportamiento del sistema en torno a su punto de operación. En la Figura 6 se muestra una comparación entre la respuesta del sistema no lineal, descrito por las ecuaciones (2) y (4), y el modelo lineal obtenido mediante la ecuación (8), ante una perturbación moderada en la entrada de control. Todas las ecuaciones fueron implementadas en MATLAB para evaluar la concordancia entre ambos enfoques y validar la fidelidad del modelo linealizado.

Figura 6
Simulación en MATLAB para comparar la respuesta del modelo linealizado y no lineal.



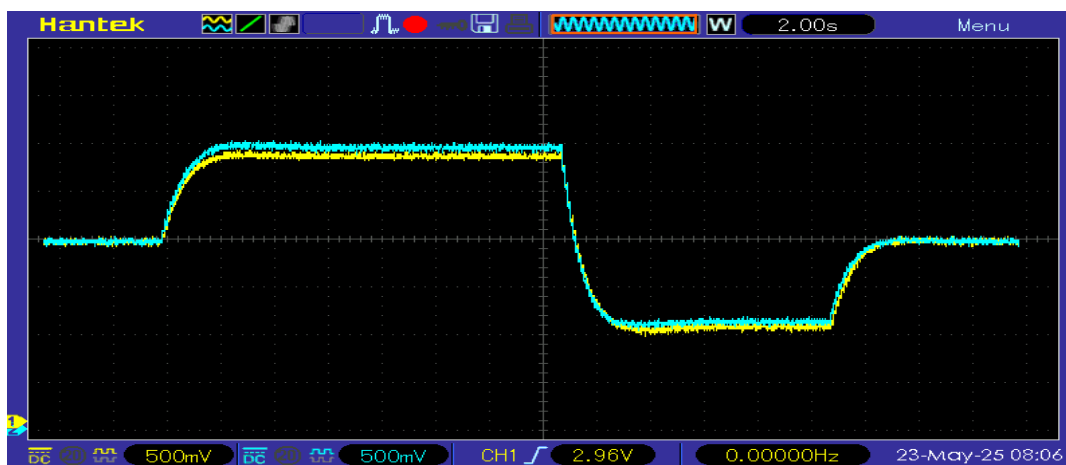


Nota. Fuente: elaboración propia.

Para llevar a cabo la comparación entre el modelo lineal y el modelo no lineal del sistema, se implementó una simulación HIL, cuyos resultados se presentan en la Figura 7. En dicha figura se observa la respuesta obtenida a partir de la ecuación (9). Las similitudes entre ambas curvas sugieren que el modelo lineal implementado en el entorno HIL es capaz de reproducir con alta fidelidad la dinámica del sistema original no lineal, validando así su aplicabilidad en el diseño y prueba de controladores en tiempo real.

Figura 7

Simulación HIL para comparar la respuesta del modelo linealizado y no lineal.



Nota. La línea azul corresponde al modelo no lineal y la línea amarilla corresponde al modelo linealizado. Fuente: Osciloscopio Hantek DSO5102P.

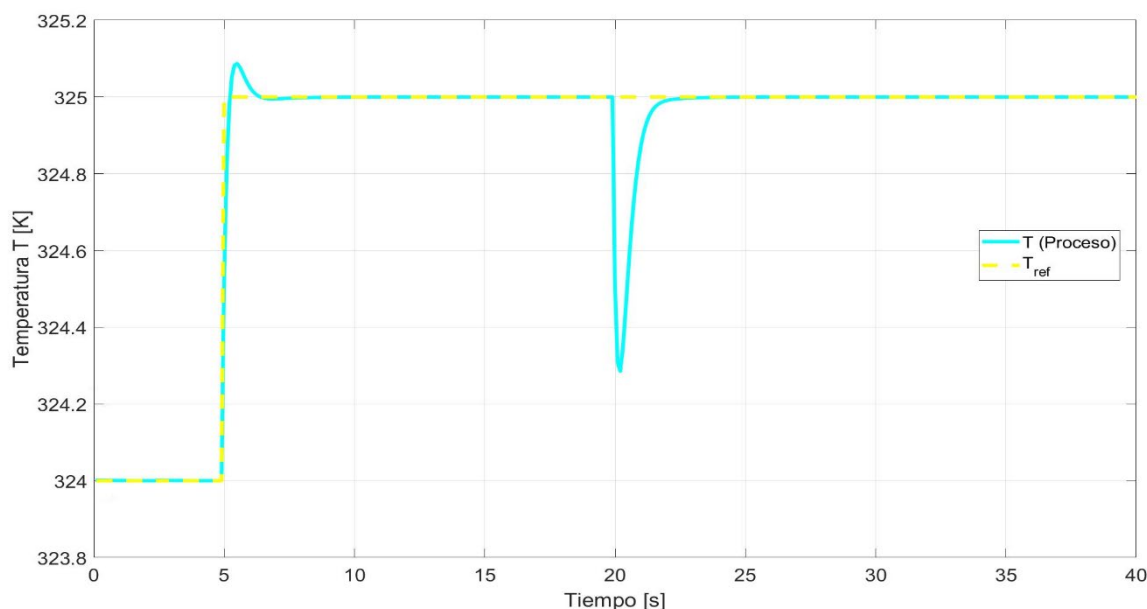
Por último, se llevó a cabo la simulación completa en lazo cerrado del sistema conformado por el controlador PI y el modelo del CSTR, utilizando el entorno de MATLAB, como se muestra en la figura 8. En el controlador se definieron las ganancias $K_p = 2.039$,



$K_p = 4.639$ y un tiempo de muestreo de $T_s = 0.1$ s. La temperatura de referencia inicial se fijó en $T_{ref} = 324$ K, posteriormente en $t = 5.0$ s, se aplicó un cambio escalonado en la referencia, aumentado de 324 K a 325 K. Para evaluar la robustez del controlador ante perturbaciones externas, se simuló una variación en la temperatura del reactivo en la entrada del reactor. En $t = 20.0$ s, la temperatura T_f se redujo de 350 K a 345 K, lo que provocó un descenso significativo de la temperatura interna del reactor. A pesar de esta perturbación, el controlador propuesto fue capaz de compensar eficazmente el efecto y restablecer la salida hacia el valor deseado.

Figura 8

Simulación en MATLAB de la respuesta en lazo cerrado del controlador PI y CSTR.



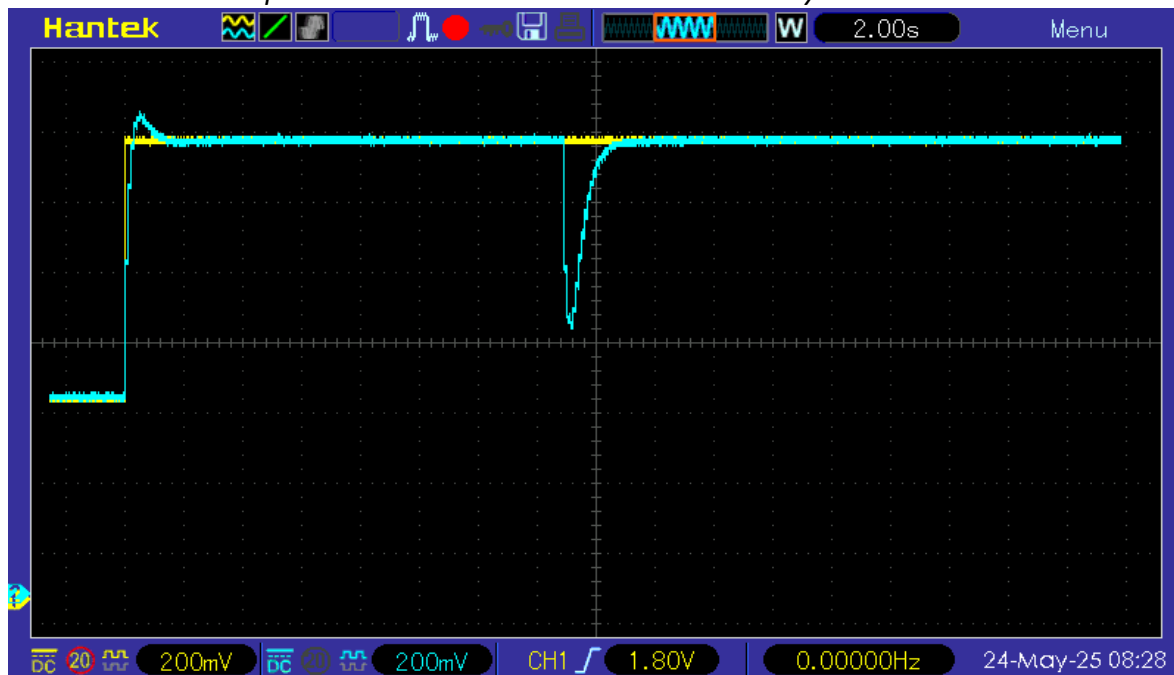
Nota. Fuente: elaboración propia.

La regulación de la temperatura mediante simulación en tiempo real con HIL se muestra en la Figura 9. Para validar el desempeño del sistema en lazo cerrado, se utilizaron las mismas ganancias del controlador PI empleadas en la simulación en MATLAB, así como las mismas condiciones de perturbación. Como se aprecia en la figura, el controlador mantiene la estabilidad del sistema y es capaz de compensar eficazmente los efectos de las perturbaciones aplicadas, asegurando un adecuado seguimiento de la referencia. También, se puede apreciar una alta congruencia con la simulación en el software.



Figura 9

Simulación de la respuesta en lazo cerrado del controlador PI y HIL CSTR.



Nota. La línea azul corresponde a la temperatura del reactor y la línea amarilla corresponde a la temperatura de referencia. Fuente: Osciloscopio Hantek DSO5102P.

5. Discusión

Los resultados obtenidos confirman la validez del enfoque HIL basado en Arduino para simular y controlar en tiempo real un CSTR. Tal como se reporta en estudios previos (Mihalič et al., 2022; Lozada & Panchi, 2023), las simulaciones HIL ofrecen una plataforma robusta y flexible para evaluar el desempeño de controladores antes de su implementación en plantas reales. En este trabajo, las simulaciones realizadas en MATLAB sirvieron como referencia para evaluar la precisión de los resultados obtenidos en el entorno embebido, mostrando una alta concordancia en la evolución temporal de las variables controladas. En particular, la comparación entre el modelo no lineal y el modelo linealizado dentro del entorno HIL demostró que la simplificación lineal conserva la dinámica dominante del sistema, lo que valida su uso para el diseño del controlador PI. Además, los resultados en lazo cerrado revelan la efectividad del controlador propuesto para compensar perturbaciones y garantizar el seguimiento de referencias, incluso ante cambios abruptos



en las condiciones del proceso. Este comportamiento es coherente con la teoría del control clásico y con las ventajas del control PI en sistemas térmicos, como se establece en Shi et al. (2023).

6. Conclusiones

Este trabajo ha demostrado que es posible implementar una plataforma de simulación HIL de bajo costo y alto valor didáctico para el estudio y validación de estrategias de control aplicadas a sistemas no lineales como el CSTR. Utilizando dos microcontroladores Arduino Due, se logró emular tanto la dinámica del proceso como la acción del controlador PI en tiempo real, con resultados comparables a los obtenidos mediante simulaciones en MATLAB. La estrategia implementada permitió evaluar tanto el comportamiento en lazo abierto como en lazo cerrado del sistema, validando la fidelidad del modelo discreto respecto al modelo continuo y no lineal. Los resultados experimentales muestran que el controlador diseñado es capaz de mantener la estabilidad del sistema y compensar eficazmente perturbaciones significativas. La arquitectura presentada ofrece una solución accesible para laboratorios educativos y escenarios de prototipado, facilitando la comprensión práctica de conceptos avanzados en control de procesos y modelado dinámico.

7. Declaración de contribución de autoría: CREdiT

Hector Daniel Hidalgo-Leal: Conceptualización, Metodología. Fabiola Cruz-Gutiérrez: Redacción, borrador original. Hector Daniel Hidalgo-Leal y Guadalupe Hernández-Hernández: Resultados, Discusión. Luis May-Burgos: Conclusiones y Revisión de estilo

8. Agradecimientos

Los autores agradecen a la jefa de división de estudios de ingeniería mecánica Ing. Gloria Susana Villalobos Rodríguez por las facilidades administrativas otorgadas y el apoyo financiero otorgado por la academia de ingeniería Mecatrónica del Instituto Tecnológico Superior de Villa La Venta.



9. Declaración de conflicto de intereses

Los autores declaran que no tienen intereses financieros en competencia conocidos ni relaciones personales que pudieran haber parecido influir en el trabajo reportado en este artículo

10. Disponibilidad de los datos

Todos los datos están disponibles en el contenido del artículo.



Referencias

1. Anbu, S., & Senthilkumar, M. (2018). Modelling and Analysis of Continuous Stirred Tank Reactor through Simulation. *Asian Journal of Engineering and Applied Technology*, 7(1), 78–83. <https://doi.org/10.51983/ajeat-2018.7.1.970>
2. AP Monitor. (2025, May 15). *CSTR Modeling and Control Case Study*. <https://apmonitor.com/che436/index.php/Main/CaseStudyCSTR>
3. Biswas, S. P., Hosain, Md. K., & Rahman, Md. W. (2017). Real-time arduino based simulator enabled hardware-in-the-loop electric DC machine drive system. *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 823–826. <https://doi.org/10.1109/R10-HTC.2017.8289082>
4. Gaur, R. (2020). Hardware in Loop Simulation to Control the Speed of DC Motor by using PID Controller. *International Journal of Computer Applications*, 175(13), 1–4. <https://doi.org/10.5120/ijca2020920400>
5. Guerreiro, J., Lourenço, A., Silva, H., & Fred, A. (2014). Performance Comparison of Low-cost Hardware Platforms Targeting Physiological Computing Applications. *Procedia Technology*, 17, 399–406. <https://doi.org/10.1016/j.protcy.2014.10.204>
6. Hidalgo, H., Orosco, R., Huerta, H., Vazquez, N., Estrada, L., Pinto, S., & de Castro, A. (2024). A Finite-Set Integral Sliding Modes Predictive Control for a Permanent Magnet Synchronous Motor Drive System. *World Electric Vehicle Journal*, 15(7), 277. <https://doi.org/10.3390/wevj15070277>
7. Hurtado, F. J., Kaiser, A. S., & Zamora, B. (2015). Fluid dynamic analysis of a continuous stirred tank reactor for technical optimization of wastewater digestion. *Water Research*, 71, 282–293. <https://doi.org/10.1016/j.watres.2014.11.053>
8. Izotov, Y. A., Velichko, A. A., Ivshin, A. A., & Novitskiy, R. E. (2021). Recognition of handwritten MNIST digits on low-memory 2 Kb RAM Arduino board using LogNNet reservoir neural network. *IOP Conference Series: Materials Science and Engineering*, 1155(1), 012056. <https://doi.org/10.1088/1757-899X/1155/1/012056>
9. Kumar, M., Prasad, D., Giri, B. S., & Singh, R. S. (2019). Temperature control of fermentation bioreactor for ethanol production using IMC-PID controller. *Biotechnology Reports*, 22, e00319. <https://doi.org/10.1016/j.btre.2019.e00319>
10. Loor Chavez, A. A. (2023). Diseño de un reactor químico de polimerización para la producción de ácido poliláctico. *Revista de Iniciación Científica*, 9(1). <https://doi.org/10.33412/rev-ric.v9.1.3560>
11. Lozada, C., & Panchi, D. (2023). Implementación de Hardware In The Loop para el Análisis de Escenarios de Control de Frecuencia en una Microrred Utilizando WAMS. *Revista Técnica "Energía,"* 19(2), 69–80. <https://doi.org/10.37116/revistaenergia.v19.n2.2023.558>
12. Melin, P., Baier, C., Espinosa, E., Riedemann, J., Espinoza, J., & Pena, R. (2022). Study of the Open-Source Arduino DUE Board as Digital Control Platform for Three-Phase



- Power Converters. *IEEE Access*, 10, 7574–7587.
<https://doi.org/10.1109/ACCESS.2021.3138705>
13. Mihalič, F., Truntič, M., & Hren, A. (2022). Hardware-in-the-Loop Simulations: A Historical Overview of Engineering Challenges. *Electronics*, 11(15), 2462.
<https://doi.org/10.3390/electronics11152462>
14. Shi, X., Zhao, H., & Fan, Z. (2023). Parameter optimization of nonlinear PID controller using RBF neural network for continuous stirred tank reactor. *Measurement and Control*, 56(9–10), 1835–1843.
<https://doi.org/10.1177/00202940231189307>
15. Sozański, K. (2023). Low Cost PID Controller for Student Digital Control Laboratory Based on Arduino or STM32 Modules. *Electronics*, 12(15), 3235.
<https://doi.org/10.3390/electronics12153235>
16. Tantawy, A., Abdelwahed, S., & Chen, Q. (2019). Continuous Stirred Tank Reactors: Modeling and Simulation for CPS Security Assessment. *2019 11th International Conference on Computational Intelligence and Communication Networks (CICN)*, 117–123. <https://doi.org/10.1109/CICN.2019.8902372>
17. Zhang, A., He, X., Cammi, A., & Wang, X. (2024). Modelica and Arduino-based hardware-in-the-loop simulation of a nuclear-powered engineering ship. *Nuclear Engineering and Design*, 429, 113650.
<https://doi.org/10.1016/j.nucengdes.2024.113650>

