

CONTROLADORES DIFUSO Y PID EN UN PROCESO TÉRMICO CON SUPERVISIÓN INALÁMBRICA USANDO TECNOLOGÍAS ANDROID Y ARDUINO

Sergio Basilio Sepúlveda Mora¹, María Camila Arenas Ordoñez², Byron Medina Delgado³

¹ Ingeniero Electrónico, Universidad Francisco de Paula Santander; sergio.sepulveda@ufps.edu.co.

² Ingeniero Electrónico, Proyectos y Soluciones en Ingeniería e Integración SAS. proyectos1@psii.co

³ Ingeniero Electrónico, Universidad Francisco de Paula Santander.

RESUMEN

En el proceso de purificación de agua para el consumo humano, controlar la temperatura contribuye con su salubridad e higiene; por lo tanto, el objetivo de este trabajo fue comparar el rendimiento de un controlador tradicional con uno difuso. Las estrategias de control se diseñaron y simularon en la herramienta computacional Matlab®, y se implementaron en el microcontrolador *Atmega* de Arduino; en ambos casos, simulación y experimentación, se consideraron el tiempo de asentamiento, la sobreelongación y el error en estado estable para comparar los controladores. El controlador difuso con respecto al controlador PID, en la simulación eliminó la sobreelongación y redujo el tiempo de asentamiento en un 76 %, y en la experimentación no generó sobreelongación y disminuyó en un 28 % el tiempo de asentamiento. El error en estado estable fue cero para los dos controladores en la simulación y la experimentación. El controlador difuso fue más estable y rápido; concluyendo que esta estrategia es la apropiada para regular la temperatura en este proceso.

Palabras clave: *Arduino, celda Peltier, controlador difuso, controlador PID, PWM.*

Recibido: 01 de noviembre de 2016

Aceptado: 20 de Junio de 2017

Received: November 01th, 2016

Accepted: June 20th, 2017

DIFFUSE AND PID CONTROLLERS FOR A THERMAL PROCESS WITH WIRELESS SUPERVISION USING ANDROID AND ARDUINO TECHNOLOGIES

ABSTRACT

In the process of water purification for human consumption, the control of the temperature contributes to its health and hygiene. The goal of this study was to compare the performance of a traditional controller with a fuzzy controller for this case study. The control strategies were designed and simulated in Matlab® software, and they were implemented in Atmega microcontroller of Arduino; in both cases, simulation and empirical setup, we considered the settling time, overshoot and steady state error to compare the controllers. The fuzzy controller eliminated the overshoot and reduced the settling time by 76 % in simulation, in comparison to the PID controller. In the experimental data, the fuzzy controller did not show overshoot and decreased the settling time by 28 % with respect to the other controller. The steady state error was zero in both controllers for simulation and for the empirical setup. The fuzzy controller was faster and more stable, for this reason, we concluded that this control strategy is appropriate to regulate the temperature in this process.

Keywords: *Arduino, Peltier cell, fuzzy controller, PID controller, PWM.*

Cómo citar este artículo: S. B. Sepúlveda-Mora, M. C. Arenas-Ordoñez, B. Medina-Delgado, "Controladores difuso y PID en un proceso térmico con supervisión inalámbrica usando tecnologías Android y Arduino," *Revista Politécnica*, vol. 13, no. 24, pp. 79-88, 2017.

1. INTRODUCCIÓN

En el proceso de purificación del agua, la etapa más importante es el almacenamiento final, porque en el proceso de purificación se puede afectar el líquido por diferentes agentes, alterando la calidad del agua. El agua de consumo humano debe tener ciertas características de seguridad, salubridad e higiene, ya que su consumo es vital para la salud de los seres humanos. La calidad del agua puede perturbarse por el crecimiento de bacterias estresadas debido a factores del medio ambiente (temperatura, oxígeno y niveles de nutrientes) [1]. La temperatura influye en la proliferación de bacterias perjudiciales para la salud humana, de no ser controlada, la calidad del agua será afectada de forma inadecuada y llegará al estado de no ser apta para su consumo.

Algunos trabajos muestran cómo la lógica difusa presenta gran aplicabilidad y flexibilidad en la implementación de estrategias de control [2]. Por ejemplo en [3], el controlador difuso tiene dos fines: primero, implementar una estrategia de control no convencional, para evaluar mediante comparación, el desempeño del controlador con acciones PID (*Proportional Integral Derivative*); y segundo, explorar el manejo de un controlador inteligente, en el cual, a partir de la experiencia en la dinámica del sistema, se llega a una sintonización adecuada. El controlador difuso rechaza mejor el ruido que el controlador PID.

Por otra parte, en [4] se muestra cómo el rendimiento de un control PID puede ser igualado y superado por un control PID difuso para controlar temperatura en aplicaciones que requieren alta precisión y estabilidad. Lo que concluye también [5], mediante la comparación de la simulación de los dos controles, es que los controladores difusos cubren una amplia gama de condiciones de operación, son fácilmente personalizables en términos de lenguaje natural y, son más rápidos y robustos.

El presente trabajo tiene como finalidad comparar la estrategia de control PID y el control difuso, utilizando tecnología Arduino en su implementación, con el fin de controlar la temperatura a 20 °C en un proceso de purificación de agua, evitando la proliferación de microorganismos perjudiciales. Este sistema se diseñó e implementó acorde con las condiciones

del proceso de purificación de agua en pequeñas industrias locales, que no cuentan con un sistema de monitorización permanente sobre los depósitos de agua.

Para la implementación de las dos estrategias de control, se programó el microcontrolador ATmega328P de la tarjeta Arduino Uno, en su *software* libre. Los diseños y simulaciones de los controles se realizaron con las herramientas *Fuzzy Logic Controller* y *PID Tuner* de Matlab®, usando las caracterizaciones del proceso de purificación de agua y del actuador (celda Peltier TEC1-12706).

En este trabajo se muestra cómo el control difuso supera al control tradicional PID tanto en simulaciones como en su implementación real, obteniendo mejores resultados en las métricas de sobreelongación y tiempo de asentamiento.

El sistema innova al incorporar la tarjeta 1Shield de Integreight Inc, que integra y comunica a Arduino con el *Smartphone*, mediante Bluetooth 2.1 y la utilización de una aplicación móvil. Esta tarjeta tiene ventajas frente a otras tarjetas accesorio de Arduino (*shields*); primero porque se reducen los escudos convencionales como GPS, Wifi, giroscopio y accesorios como LCD, pulsadores y alarmas a solo una tarjeta, dando como resultado una reducción de costos y espacio de aplicación; segundo porque es una herramienta compatible con los *Smartphone* Android; y tercero porque la comunicación por medio de *1Shield* aprovecha los sensores y aplicaciones presentes en los celulares dependiendo de su complejidad, como acelerómetros, sensores internos de temperatura del *Smartphone* y aplicaciones como alarmas de reloj, mensajes de texto y acceso a redes sociales.

2. MATERIALES Y MÉTODOS

2.1 Arquitectura y funcionamiento del sistema

El sistema de control de temperatura implementado en este estudio está compuesto fundamentalmente por los dispositivos que se presentan en la Fig. 1; además de elementos necesarios para el correcto funcionamiento de estos dispositivos. Esta configuración fue utilizada para los controladores PID y difuso.

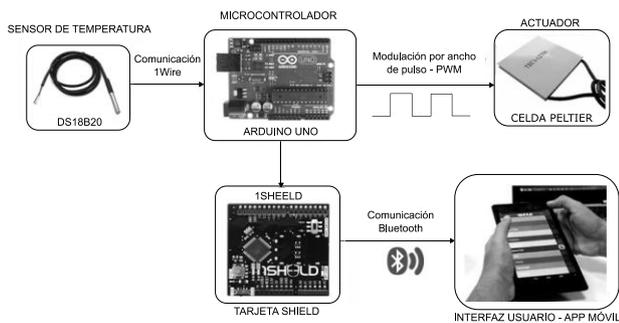


Fig. 1. Arquitectura del sistema de control.

La tarjeta Arduino se comunica con el sensor DS18B20 de temperatura mediante el protocolo de comunicación *1Wire*, que proporciona la temperatura en grados centígrados con una resolución programable de 9 a 12 bits [6-8]. La tarjeta Arduino también establece comunicación inalámbrica vía Bluetooth con el *Smartphone* mediante la tarjeta *1Shield* conectada físicamente a ella. El uso de Bluetooth, en este proyecto, se sustenta por su reconocimiento y respaldo, además por su compatibilidad con las tecnologías Android y Arduino.

Una vez se establece la comunicación, el sensor empieza a registrar el valor de la temperatura en grados centígrados y se visualiza en el *Smartphone* mediante el escudo terminal de la aplicación *1Shield*. Mientras se visualiza el valor de la temperatura, el usuario puede modificarlo enviando el valor que desea ajustar. Al establecerse la temperatura, el microcontrolador procesa la información del sensor, calcula el error de la temperatura y el valor de la salida del controlador, para generar la señal digital que modula el ancho de pulso (*Pulse Width Modulation*, PWM) y controla la celda Peltier [9], [10], a una frecuencia de 31 kHz.

En ambos controles implementados el proceso de generación de la señal PWM es el mismo, solo cambia el ciclo de trabajo de acuerdo con el algoritmo de cada controlador.

En la etapa de potencia de la celda Peltier se utilizó el Mosfet IRFZ44N, ya que conmuta a altas frecuencias y es controlado por tensión entre sus pines de compuerta y fuente, ver Fig. 2. Además, se utilizó un circuito integrado de colector abierto 74LS05 para asegurar la correcta saturación y corte del Mosfet, lo que permite que sobre la celda Peltier

se ejerza control de su voltaje y consecuentemente de su potencia [7].

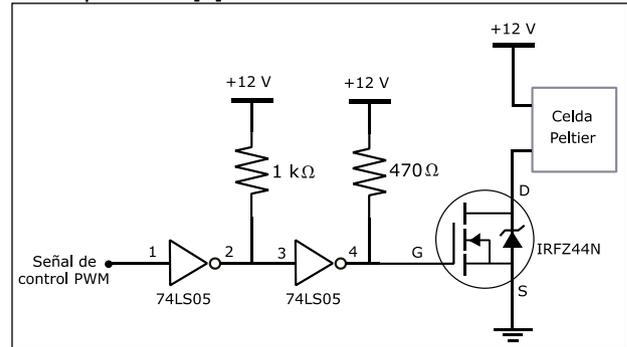


Fig. 2. Circuito de la etapa de potencia.

El diagrama de bloques del sistema de control de temperatura se representa en la Fig. 3, en éste se muestra el control tradicional PID y el control difuso. El *software* utilizado para la programación del controlador es el compilador Arduino v1.6.7 (*Integrated Development Environment*, IDE), un entorno de desarrollo que utiliza lenguaje C.

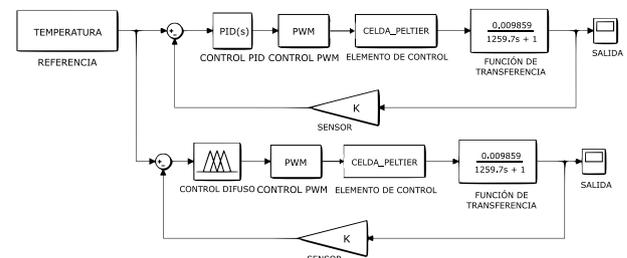


Fig. 3. Diagrama sistema de control PID y difuso.

2.2 Caracterización de la Planta

Experimentalmente se halló la respuesta en lazo abierto del prototipo a escala del cuarto de almacenamiento de agua, implementando una celda Peltier para hacer descender la temperatura en el prototipo hasta 20 °C.

En la Fig. 4 se puede observar el valor máximo de diferencia de temperatura que alcanza la cámara equivalente a los 15 °C (desde 31 °C hasta 16 °C) en un tiempo aproximado de 45 minutos. Sin embargo, debido a la proliferación de algunas bacterias se estableció trabajar a 20 °C.

Para el control de temperatura se identificó la planta obteniendo una función de transferencia mediante la herramienta *Ident*, del *software* Matlab® versión R2015a, donde se eligió, de

manera heurística, el diseño de una planta de primer orden, como se aprecia en la ecuación 1.

$$G(s) = \frac{0,009859}{1259,7 s + 1} \quad (1)$$

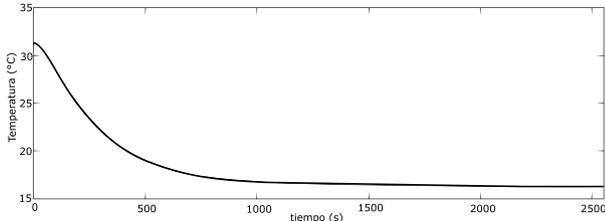


Fig. 4. Respuesta en lazo abierto del prototipo.

La ausencia de retardo en la función de transferencia de la ecuación 1 se puede justificar a partir de las dimensiones del prototipo (un cubo de 30 cm de arista con caras de acero inoxidable). A pesar que la temperatura es una variable que no cambia de forma abrupta, los datos experimentales ajustados con la herramienta *Ident* se adaptaron a la ecuación presentada.

2.3 Sintonización del controlador PID

El cálculo del controlador PID implica tres acciones de control diferentes: la acción proporcional, la integral y la derivativa. El valor proporcional determina la acción al error actual, el integral determina la reacción basada en la suma de errores recientes y el derivativo determina la anticipación a cambios en el error [11], [12]. La suma de estas tres acciones se utilizó para ajustar el proceso de control del sistema. Para este caso, la planta se estableció como un sistema SISO (*Single Input, Single Output*), por lo cual se diseñó un controlador PID para la variable temperatura [13].

Se utilizaron del software Matlab® las herramientas *PID Tuner* para el diseño del control PID y Simulink® para verificar su comportamiento. En la herramienta *PID Tuner* se utilizó el método de lazo cerrado balanceado, en el cual, la sintonización del controlador mantiene un equilibrio entre la búsqueda de referencia y el rechazo a la perturbación. En la Tabla I se presentan las características de la planta sin controlar; estas características definen una planta que presenta valores de parámetros no adecuados, como un tiempo de asentamiento alto y una respuesta lenta. Los datos fueron obtenidos a través de Matlab® y corresponden con la Fig. 4.

Tabla 1. Características de la planta sin controlar.

Característica	Valor
Sobreelongación	0 %
Tiempo de subida	2950 s
Tiempo de ajuste	5250 s
Amplitud pico	0,00976 °C

En la Fig. 5 se representan los valores de las constantes de control Kp (proporcionalidad), Ki (integración) y Kd (derivación), el ángulo de fase y el ancho de banda del controlador.

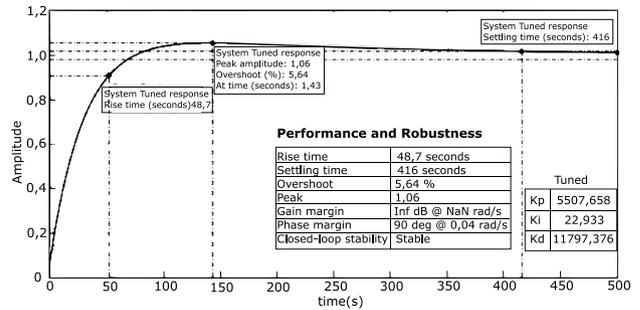


Fig. 5. Respuesta de la planta ante el controlador con PID Tuner.

Por medio del bloque *controller PID* de Simulink® se evalúa el desempeño del controlador PID obtenido con el *PID Tuner*, mediante la estructura en lazo cerrado y se obtiene una respuesta subamortiguada como se representa en la Fig. 6. Las características de la planta con el control aplicado son una sobreelongación de 5,64 %, un tiempo de subida de 48 s, un tiempo de asentamiento de 416 s y una amplitud pico de 1,06. Para el tiempo de asentamiento se usó un criterio del 2 %.

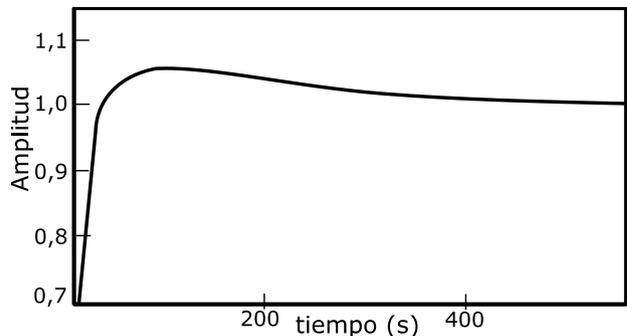


Fig. 6. Respuesta controlador PID con Simulink.

2.4 Diseño control difuso

El controlador lógico difuso se basa en el modelo lingüístico de la estrategia del operador humano. Es

decir, el controlador opera con conceptos del razonamiento cualitativo, tomando decisiones a partir de un conjunto de observaciones y reglas cualitativas. La esencia de tal modelo es un programa basado en reglas, por lo que clasifica entre los llamados sistemas expertos [14].

En el proceso de diseño de un controlador difuso se tienen en cuenta cuatro procesos: fusificación, base de conocimiento, toma de decisiones (mecanismo de inferencia) y defusificación.

El controlador difuso seleccionado es de tipo *Mamdani*, en el cual la entrada y la salida son conjuntos difusos. A diferencia del esquema *Sugeno*, donde la salida es una función relacionada con las variables del proceso como se trabaja en [15]. Se escogió *Mamdani* para este estudio por la facilidad para establecer e interpretar las reglas, y la compatibilidad con Arduino.

Se utiliza una herramienta de Matlab® denominada *Fuzzy Logic Designer*, esta vez para implementar el controlador tipo *Mamdani*. Con la herramienta se definen las entradas y salidas; los métodos de implicación, agregación y defusificación.

En el proceso de fusificación se midieron las variables de entrada al controlador realizando un mapeo a escala que transformó el rango de valores de las variables de entrada en los correspondientes valores lingüísticos (ver Tabla II) en el universo discurso [14].

Definida la cantidad y forma de las funciones de membresía, se experimentó con las reglas que relacionan la entrada Error con la salida PWM. El diseño manejó la variable difusa de entrada para efectuar la señal de control de una manera más rápida, proporcionando en la respuesta del sistema la eliminación de la sobreelongación. Las reglas basadas en el comportamiento deseado se enuncian en la Tabla III mediante la sintaxis si-entonces, característica del controlador tipo *Mamdani*.

El paso seguido dentro de la lógica difusa, para obtener la salida del controlador, es la defusificación. En este paso, se utilizó el método centroide para obtener un valor asociado a los conjuntos difusos como se representa en el simulador de Matlab® (ver Fig. 7).

Tabla 2. Proceso de fusificación.

Características de entrada y salida	Asignación
Entrada	1
Variable lingüística	Error
Universo de discurso	[-10 °C; 10 °C]
Particiones universo discurso	3 conjuntos difusos
Función membresía (Valores lingüísticos entrada)	Negativo, Cero y Positivo
Salida	1
Variable lingüística	PWM
Función membresía (Valores lingüísticos salida)	Setenta, Cincuenta y Noventa

Tabla 3. Reglas difusas de manera lingüística.

Error	Pertenencia	PWM
Cero	1	50 %
Negativo	1	90 %
Positivo	1	70 %
Cero	0,1	70 %
Negativo	0,2	70 %

Las reglas diseñadas en la herramienta *Fuzzy Logic Designer* se evaluaron en Simulink® con el bloque *Fuzzy Logic Controller*. La respuesta al control se evidencia en la Fig. 8. Las características que presenta la planta controlada son: sobreelongación de 0 %, tiempo de subida de 55,1 s, tiempo de ajuste de 98,8 s y amplitud pico de 0,99 °C.

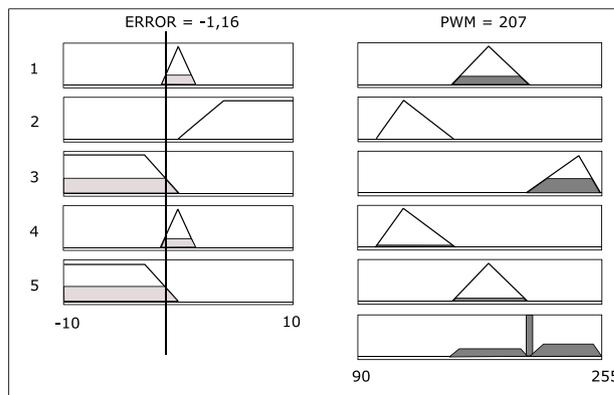


Fig. 7. Activación de reglas con error negativo.

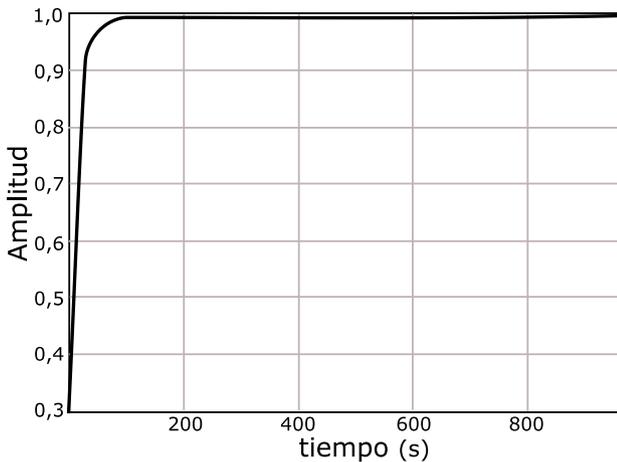


Fig. 8. Respuesta del control difuso con Simulink.

2.5 Implementación en Arduino

Los dos programas desarrollados son diferentes en la arquitectura del procesamiento. Para ejecutar el algoritmo en ambos controladores se instalaron en el IDE de Arduino las bibliotecas: *OneWire* para la comunicación con los sensores digitales en el protocolo *1-Wire*; *DallasTemperature* para la comunicación con el sensor DS18B20; y *OneSheeld* para la comunicación con el *Smartphone*.

En general, para los dos controladores con la implementación de los programas, se lee la variable temperatura del sensor en intervalos de tiempo regulares de 1 s, y se puede monitorizar en el *Smartphone* con la función terminal de la *app* de la *1sheeld*. El tiempo de muestreo, de 1 s, fue suficiente para obtener la información de la dinámica del sistema. Un tiempo de muestreo inferior es innecesario porque la temperatura es una variable que presenta cambios lentos.

Para el control PID se procesó la información del error del sistema y se aplicaron las 3 acciones de control (proporcional, integral y derivativa), basados en las variables encontradas en Matlab® con ayuda del *PID Tuner*. Se cargó el módulo PID y de acuerdo con su salida se establecieron los parámetros del ciclo de trabajo para la salida PWM como se muestra en la Fig. 9.

Para el diseño del controlador difuso se instaló la librería *Fuzzy* de Arduino, conocida como eFLL. Como se presenta en el diagrama de flujo de la Fig. 10, se declararon de manera lingüística las variables de entrada y salida en el proceso de

fusificación, las funciones de membresía de cada variable difusa con su respectivo universo de discurso y niveles de pertenencia; se definieron las reglas basadas en las simulaciones en el *software* Matlab®; y se consolidaron las variables de entrada y salida para realizar el proceso de defusificación cargando el valor de PWM esperado.

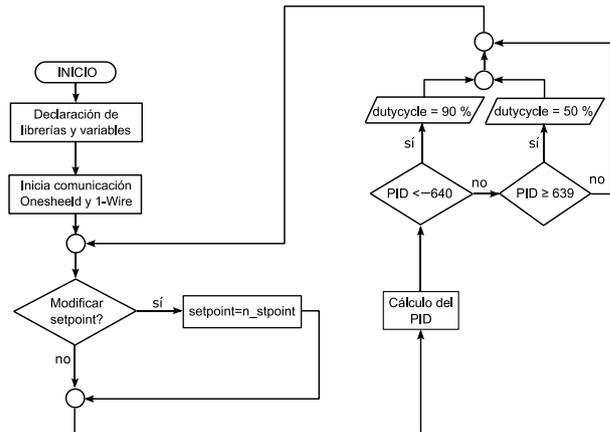


Fig. 9. Diagrama de flujo del controlador PID.

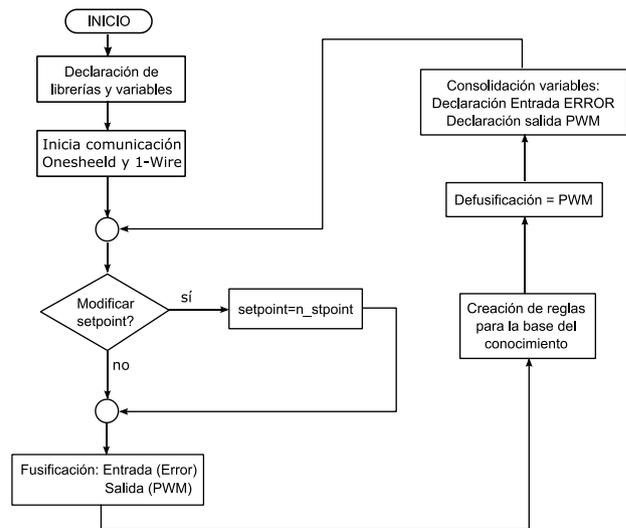


Fig. 10. Diagrama de flujo del controlador Difuso.

3. RESULTADOS Y DISCUSIÓN

3.1 Técnica PWM

El microcontrolador Atmega328 para la placa Arduino Uno maneja en sus salidas 5 V como se aprecia en la Fig. 11, y una corriente de 40 mA, por

lo cual, para manejar cargas de 50 W como las celdas Peltier, se implementó un circuito de potencia controlado por las señales de Arduino, con la capacidad de manejar la celda Peltier.

Para el proceso de enfriamiento debe ocurrir el efecto Peltier, el cual se fundamenta en el movimiento de electrones a través de los TEC (*Thermo Electric Cell*) que conforman la celda Peltier [16]. Estas TEC son dispositivos compuestos por varios termopares, térmicamente conectados en paralelo y eléctricamente en serie [17].

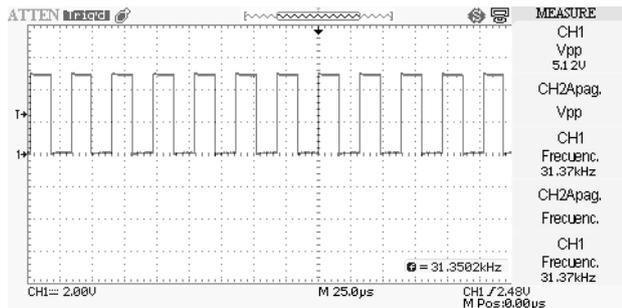


Fig. 11. Salida de señal PWM, tarjeta Arduino.

Basados en [16], se modeló el acondicionamiento de la celda Peltier, para que no se comportara como un generador eléctrico, es decir, que no se generara el efecto Seebeck; evitando que en el cambio al estado bajo de la señal PWM, se presentara un rizado negativo.

Para contrarrestar este efecto se conectó un diodo Schotky. Este dispositivo funcionó como diodo de rueda libre, conectado en antiparalelo a la celda Peltier, y permitió que la corriente generada por el efecto Seebeck mantuviera la dirección establecida cuando la señal PWM estaba en alto; evitando que se generara un movimiento bidireccional de electrones y fonones, al equilibrar la temperatura en la celda [18].

3.2 Respuesta del sistema de control

Dentro de las características eléctricas del prototipo del cuarto de almacenamiento de agua, se encuentran la implementación de dispositivos como una celda Peltier (consumo de 3,4 A a 12 V), un ventilador del lado caliente (consumo de 1,8 A a 12 V), un ventilador del lado frío (consumo de 0,188 A a 12 V), una tarjeta Arduino Uno (consumo de 0,8 A a 9 V), y un sensor DS18B20 (consumo de 0,28 A a 5 V). Por lo tanto, el prototipo de cámara de

almacenamiento de agua consume aproximadamente 6,47 A.

Se realizaron las pruebas de la aplicación de los controles sobre la tarjeta Arduino. Se ingresó a la aplicación *1Shield* para modificar el *setpoint*, se estableció la conexión entre la tarjeta y el móvil vía Bluetooth, se estableció el *Shield* terminal, y se visualizaron los datos en el *Smartphone* (ver Fig. 12).



Fig. 12. Visualización de la aplicación *1Shield*.

En la Fig. 13 se presentan las simulaciones del control PID y del control difuso, bajo las mismas condiciones. El controlador PID presenta una sobreelongación de 5,64 % y un tiempo de asentamiento de 416 s. El controlador difuso no tiene sobreelongación y disminuye el tiempo de asentamiento en un 76 % con respecto al controlador PID. Los datos mencionados evidencian que el controlador difuso tiene un mejor desempeño que el controlador PID.

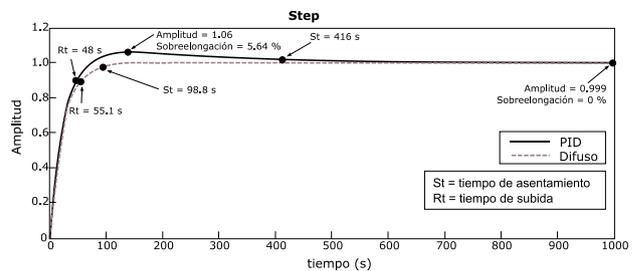


Fig. 13. Simulación de los controles PID y difuso.

Los datos experimentales obtenidos en la implementación de ambos controladores en la tarjeta Arduino, se muestran en la Fig. 14. El tiempo de asentamiento del control PID, está alrededor de los 500 s y la sobreelongación es de 2,5 %; mientras que para el control difuso el tiempo de asentamiento es cercano a los 370 s y la

sobreelongación es de 0 %; es decir, el desempeño del control difuso es superior, porque su tiempo de asentamiento fue menor en un 26 % y no tuvo sobreelongación, con respecto al control PID.

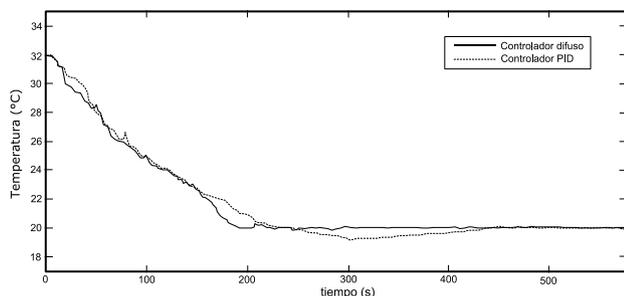


Fig. 14. Controles en Arduino, dato experimental.

En la Tabla IV se resumen los valores de las métricas consideradas para los controladores PID y difuso, obtenidos en la simulación y en la experimentación. Las métricas presentadas son: la sobreelongación (M_p), el tiempo de asentamiento (t_s) y el error en estado estable (e_{ss}).

Tabla 4. Métricas de los controladores

Tipo de controlador	Simulación			Experimentación		
	M_p (%)	t_s (s)	e_{ss}	M_p (%)	t_s (s)	e_{ss}
PID	5,64	416	0	2,5	500	0
Difuso	0	98,8	0	0	370	0

Adicionalmente a las métricas de la respuesta transitoria, se calcularon la Integral del Error Absoluto (*Integral Absolute Error*, IAE) y la Integral del Error Cuadrático (*Integral Squared Error*, ISE) para comparar el desempeño del controlador difuso y el controlador PID. En la Tabla V se presentan los resultados de las integrales del error.

Tabla 5. Integrales del error

Controlador	Integral del error	
	IAE	ISE
PID	1192,17	8346,47
Difuso	1035,85	7714,48

La implementación de los controladores es una ventaja para el cuarto de almacenamiento de agua, porque entre más rápido se establezca la temperatura de referencia mejor es el cuidado del producto, evitando su deterioro y por consiguiente pérdidas para la empresa.

4. CONCLUSIONES

La herramienta *PID Tuner* de Simulink permitió sintonizar y simular el controlador PID, mediante un método heurístico, con un tiempo de asentamiento de 416 s, una sobreelongación de 5,64 % y un error en estado estable igual a cero, para la señal de temperatura, manteniendo estable el proceso de purificación de agua modelado con la herramienta *Ident* de Matlab®.

El controlador difuso se simuló con la herramienta *Fuzzy Logic Designer* de Matlab® y se implementó con la librería *eFLL* de Arduino. Con el uso de estas herramientas se confrontaron los datos experimentales con los simulados. Las métricas de sobreelongación y error en estado estable fueron iguales tanto en la simulación como en la implementación; sin embargo, el tiempo de asentamiento en los datos experimentales fue 3,7 veces mayor en comparación con la simulación. Por lo tanto, se infiere que la diferencia en el tiempo de asentamiento se debe a las capacidades reducidas de procesamiento de la tarjeta Arduino, a las no linealidades presentes en el entorno real y a la propagación de retardos entre componentes hardware.

La ausencia de sobreelongación en el controlador difuso evidencia mayor estabilidad relativa del sistema, con respecto al controlador PID. El menor tiempo de asentamiento del controlador difuso en comparación con el PID, demuestra su mayor rapidez para estabilizar la temperatura. El error en estado estable en ambos controladores fue cero, lo que indica que las dos estrategias de control fueron efectivas para mantener la temperatura en su valor de referencia. Por lo tanto, considerando las métricas de sobreelongación, tiempo de asentamiento y error en estado estable, se concluye que para este estudio, en la simulación y en la experimentación, el controlador difuso fue más estable y rápido.

La comparación realizada a través de la Integral de Error Absoluto, IAE y la Integral del Error Cuadrático, ISE permitió verificar que el controlador difuso tiene un mejor desempeño, considerando el valor del error instantáneo desde la perturbación hasta que el sistema estabiliza la temperatura al valor de referencia. Esto significa que el controlador difuso no solo es más rápido, como lo evidencia el tiempo de asentamiento menor, sino que durante el

tiempo en que actúa la señal de control, la salida es más cercana al valor de referencia en comparación con el controlador PID.

Las respuestas experimentales del sistema para ambos controladores tuvieron un menor desempeño en comparación con las simulaciones, porque en la experimentación se incorporan la incertidumbre en la medición y la no linealidad de algunos componentes, entre otros. Sin embargo, el desempeño del controlador difuso sigue siendo superior que el controlador PID.

La supervisión inalámbrica del proceso de purificación de agua, a través de la tarjeta OneSheeld y el Smartphone, permitió crear una interfaz remota para interactuar con el controlador. La supervisión remota evita posibles perturbaciones en la temperatura por la transferencia de calor entre el cuerpo humano y la planta.

5. AGRADECIMIENTOS

Los autores agradecen al Grupo de Investigación y Desarrollo en Microelectrónica Aplicada, GIDMA y al Grupo de Investigación y Desarrollo en Electrónica y Telecomunicaciones, GIDET de la Universidad Francisco de Paula Santander por suministrar el apoyo logístico y financiero en la ejecución de este proyecto.

6. REFERENCIAS BIBLIOGRÁFICAS

- [1] Wang, H., Zhu, Y. y Hu, C. Impacts of bacteria and corrosion on removal of natural organic matter and disinfection byproducts in different drinking water distribution systems. *International Biodeterioration & Biodegradation*, 117, 52-59, 2017.
- [2] Espitia H. E. y Soriano-Méndez, J. J. Diseño y simulación de un controlador difuso de temperatura empleando el congresor basado en relaciones booleanas, *Tecnura*, 16(32), 29–40, 2012.
- [3] Rairan-Antolines, J. D., Guerrero-Cifuentes, C. E. y Mateus-Pineda, J. A., Diseño de controladores de tipo proporcional integral derivativo (PID) y difuso para la posición de un motor de corriente continua (DC), *Ingeniería y Universidad*, 14(1), 137–160, 2010.
- [4] Ferreyra, A. y Fuentes, R. Estudio comparativo entre control PID y difuso. *Memorias, XV Congreso Nacional de Instrumentación*. Ciudad de México, México, 1-6, 2012.
- [5] Abdullah, A. y Ayman, A. The advantages of PID fuzzy controllers over the conventional types. *American Journal of Applied Sciences*, 5(6), 653–658, 2008.
- [6] Amarin, H. S. y Dias, M. A. Sensores digitais de temperatura com tecnologia one-wire: Um exemplo de aplicação didática na área de condução térmica. *Revista Brasileira de Ensino de Física*, 37(4), 4310–1–4310–9, 2015.
- [7] Scharfglass, K. y Lehmer, A. *Wireless sensor network for wine fermentation*, California Polytechnic State University, Tech. Rep., 2012.
- [8] Vera-Romero, C. A., Barbosa-Jaimes, J. E. y Pabón-González, D. C. Acople de sensores en la medición de variables ambientales usando tecnología Zigbee. *Scientia Et Technica*, 19(4), 419–424, 2015.
- [9] Galván-Cruz, G. S. Control retroalimentado de un módulo termoelectrico. *Universidad Tecnológica de la Mixteca*, Tech. Rep., 2012.
- [10] De-Moura, P. R. y Almeida, D. Refrigerador termoelectrico de Peltier usado para estabilisar um feixe laser em experimentos didáticos. *Revista Brasileira de Ensino de Física*, 36(1), 1–5, 2014.
- [11] Golnaraghi, F. y Kuo, B. C. *Design of control systems*. En: *Automatic control systems*, 9th Ed., Wiley, reading 664-835, 2010.
- [12] Ogata, K. *PID Controllers and Modified PID Controllers*. En: *Sistemas de control automático*, 5th Ed., Prentice Hall, Inc., reading 567-647, 2010.
- [13] Solaque-Guzmán, L., Cristancho-Cardozo, C. A. y Gil-Cárdenas, C. A. Diseño e implementación de una plataforma experimental de dos grados de libertad controlada por dos técnicas: PID y lógica difusa. *Ciencia e Ingeniería Neogranadina*, 24(1), 99–115, 2014.
- [14] Castillo, O. y Melin, P. A review on the design and optimization of interval type-2 fuzzy

controllers. *Applied Soft Computing*, 12, 1267-1278, 2012.

- [15] Sa-ngawong, N. y Ngamroo, I. Intelligent photovoltaic farms for robust frequency stabilization in multi-area interconnected power system based on PSO-based optimal Sugeno fuzzy logic control. *Renewable Energy*, 74, 555-567, 2015.
- [16] Hermes, C.J. y Barbosa Jr, J. R. Thermodynamic comparison of Peltier, stirling, and vapor compression portable coolers. *Applied Energy*, 91(1), 51 – 58, 2012.
- [17] Paez-Moreno, J. J. y Rodríguez-López, J. E. Propiedades eléctricas y estructurales de compuestos cerámicos del tipo camno-y. *Momento*, 44, 22–34, 2012.
- [18] Leal, J. F., León, M. M. y Sepúlveda, S. B. El diodo Schottky como atenuador del efecto Seebeck en una celda Peltier para un control PID de temperatura. *Entre Ciencia e Ingeniería*, 9(18), 75–83, 2015.