Artículo **Revista de Simulación Computacional**
Diciembre, 2018 Vol.2 No.6 9-15

Implementación de un controlador difuso de temperatura desarrollado en un sistema embebido Raspberry Pi

Implementation of a fuzzy temperature controller developed in an embedded Raspberry Pi system

GARCÍA-MARTÍNEZ, José Román†*, RAMÍREZ-GONZÁLEZ, Luis David, EDSON-CRUZ, Miguel y MARTÍNEZ-SÁNCHEZ, Trinidad

Universidad Autonoma de Querétaro, Cerro de Las Campanas, s/n, Col. Las Campanas, 76010 Santiago de Querétaro.

ID 1er Autor: *José Román, García-Martínez* / **ORC ID**: 0000-0001-5773-6068, **CVU CONACYT ID**: 778619

ID 1er Coautor: *Luis David, Ramírez-González* / **ORC ID**: 0000-0003-1886-2330

ID 2do Coautor: *Miguel, Edson-Cruz* / **ORC ID**: 0000-0001-6450-875X, **CVU CONACYT ID**: 632744

ID 3er Coautor: *Trinidad, Martínez-Sánchez* / **ORC ID**: 0000-0002-7493-3079

Recibido: 20 de Septiembre, 2018; Aceptado 29 de Noviembre, 2018

Resumen

En este artículo, se presenta una aplicación práctica de un controlador difuso de temperatura desarrollado en un sistema embebido Raspberry Pi 3, para el control de la temperatura de un horno didáctico basado en la inteligencia artificial, para elaborar el controlador se usó el lenguaje de programación Python 2, así como para el control de los GPIO's (General-Purpose Input/Output) de la Raspberry. El objetivo del controlador es mantener la temperatura del sistema en un intervalo propuesto, lo cual permitió accionar diferentes dispositivos mediante una señal de control. En el tiempo de muestreo de 2 segundos, el sensor captó la temperatura real y pudo pasarla al proceso de fusificación, manipulando la señal en forma de cantidades difusas, llevándose así el proceso de defusificación para obtener la señal de control, que interactuará con los actuadores finales.

Controlador difuso, Sistema embebido, Raspberry Pi

Abstract

In this article, a practical application of a fuzzy temperature controller developed in an embedded Raspberry Pi 3 system is presented, to control the temperature of a didactic oven based on artificial intelligence, to design the controller Python 2 programming language was used, as well as to control the GPIO's (General-Purpose Input / Output) of the Raspberry Pi. The objective of the controller is to maintain the temperature of the system in a proposed range, which allowed to drive different devices using a control signal. In the sampling time of 2 seconds, the sensor captured the real temperature and could pass it to the fuzzification process, manipulating the signal in the form of fuzzy quantities, thus taking the process of defuzzification to obtain the control signal, which will interact with the final actuators.

Fuzzy controller, Embedded system, Raspberry Pi

Citación: GARCÍA-MARTÍNEZ, José Román, RAMÍREZ-GONZÁLEZ, Luis David, EDSON-CRUZ, Miguel y MARTÍNEZ-SÁNCHEZ, Trinidad. Implementación de un controlador difuso de temperatura desarrollado en un sistema embebido Raspberry Pi. Revista de Simulación Computacional. 2018. 2-6: 9-15.

* Correspondencia al autor (correo electrónico: jgarcia198@alumnos.uaq.mx)
† Investigador contribuyendo como primer autor

Introducción

Los avances tecnológicos obligan a los diseñadores a buscar nuevas técnicas para el control de sistemas complejos. Los sistemas de control se pueden dividir en tres categorías: en la primera, los controladores *on/off*, que se comportan como interruptores, apagando y encendiendo la señal que se suministra a la planta. En la segunda, se encuentran los controladores Proporcional-Integral-Derivativo (PID), este tipo de controladores son los más utilizados en la industria por su simplicidad y desempeño, la señal de control es una combinación lineal de la señal de error, de su integral y su derivada (Elnour & Taha, 2013).

Los controladores PID ofrecen un control preciso de la planta debido a que existen diversas técnicas de sintonización y por lo tanto son utilizados en sistemas que presentan perturbaciones frecuentes. Por último, se encuentran los controladores inteligentes, aquellos que ocupan lógica difusa, redes neuronales, algoritmos genéticos, sólo por mencionar algunas herramientas de diseño. Los controladores que utilizan lógica difusa se denominan controladores difusos y se basan en la caracterización del razonamiento humano mediante variables de control lingüísticas, este tipo de variables pueden tener la forma de "más o menos", "casi", "cerca de", "grande", etc. A diferencia de la lógica clásica, la cual trabaja con valores de verdadero o falso (es decir o es 0 o es 1), la lógica difusa trabaja en un intervalo numérico de [0, 1]. Los controladores difusos han demostrado resultados más favorables que los controladores convencionales PID, debido a su fácil representación para la realización de estrategias de control basadas en la experiencia (Isizoh, Okide, Anazia, & Ogu, 2012).

Por otro lado, los sistemas embebidos desempeñan una parte fundamental en el desarrollo de nuevas tecnologías y se encuentran en constante cambio de actualización (Li, Guan, & Yuan, 2008). Con el aumento en el rendimiento de los sistemas tecnológicos y la potencia informática, el sistema operativo integrado desempeña un papel cada vez mas importante en los sistemas embebidos para utilizar de una manera óptima los recursos del sistema brindando un mejor control de recursos, gestión de tareas y memoria, rendimiento en tiempo real e incluso conocimiento del poder de procesamiento (Lin & Hsueh, 2006).

La familia de computadoras individuales Raspberry Pi, ha ganado popularidad en diversas áreas, entre las que se pueden destacar, Programación Orientada a Objetos, Robótica, Telecomunicaciones, entre otras. Este tipo de sistemas embebidos cuentan con un sistema operativo basado en Linux y están diseñados y fabricados con el objetivo de mejorar la educación informática, especialmente en el nivel preuniversitario (Saii Yamanoor & Yamanoor, 2016), sin embargo la potencia que ofrece puede llegar a abarcar temas complejos de investigación y desarrollo dentro de la industria, abriendo paso a la denominada cuarta revolución industrial.

La detección y control de temperatura es una de las tareas más importantes en la industria de manufactura. En los últimos años se han realizado diversos trabajos de investigación aplicando diferentes técnicas de control, por ejemplo, Wei Jiang y Xuchu Jiang (Jiang & Jiang, 2012) diseñó un algoritmo de control de temperatura inteligente basado en un PID de autoajuste difuso. Mohammed Elnour y Waleed Ibrahim (Elnour & Taha, 2013) presentaron un sistema de control de temperatura utilizando PID y lógica difusa, los resultados presentados en ambos artículos muestran resultados satisfactorios por medio de simulaciones, dejando a un lado la aplicación en un entorno real al controlador. El presente artículo expone el diseño y aplicación de un controlador basado en lógica difusa implementado en una Raspberry Pi 3, para esto se desarrolló el algoritmo de control utilizando el lenguaje de programación Python 2.

La organización de este trabajo se presenta de la forma siguiente: en la sección 1, se proporciona una breve introducción al estudio del controlador difuso de temperatura. La sección 2, presenta la metodología utilizada en el desarrollo de un controlador difuso de temperatura, a partir de la decripción de las funciones de membresía, las reglas para el diseño del controlador difuso que son la base para el correcto funcionamiento del mismo, el método de defusificación, el cual transforma los valores difusos de la señal de control, a su equivalente valor nítido de salida. La sección 3, presenta el diseño del controlador, la implementación y los resultados, en donde se utilizó la herramienta computacional Raspberry Pi 3, la librería pyFuzzy y el lenguaje de programación Python 2.7.

ISSN 2523-6865

La sección 4, presenta los agradecimientos.

En la sección 5, se presentan las conclusiones. Finalmente, en la sección 6, se presentan las referencias de este trabajo.

Metodología

El problema a resolver en este artículo es diseñar un sistema de control difuso para el control de la temperatura de un horno didáctico basado en la inteligencia artificial, donde la temperatura del horno a controlar es variada en el rango de 25°C a 50°C. Para esto se utilizó la herramienta computacional Raspberry Pi 3, la librería pyFuzzy y el lenguaje de programación Python 2.7, mostrando además el diseño del controlador y los resultados de temperatura obtenidos.

El control difuso puede ser comprendido como un sistema inteligente, desarrollado a partir de la experiencia del operador para dar solución a algún problema, y es resuelto mediante reglas lingüísticas (Ponce, 2010) de la siguiente forma:

Si (***if***) la temperatura es alta y (***and***) la temperatura deseada es normal, entonces (***then***) la temperatura de salida es baja.

Los valores lingüísticos como "*la temperatura es alta*" y "*la temperatura deseada es normal*" en combinación con los operadores difusos "***and***" son tratados como objetos numéricos y los algoritmos como tablas de funciones, interpolación y comparadores.

El operador difuso "***and***", al igual que en la teoría clásica de conjuntos, representa la intersección o conjunción de dos conjuntos. En términos de lógica difusa, sea $A,\ B\ \in\ F(x)$, la intersección del conjunto A y el conjunto B se presentan en la ecuación (1).

$$F(x) = min\{A(x), B(x)\} = A(x) \wedge B(x), \quad \forall x\ \in X \tag{1}$$

Donde $F(x)$ denota la colección de conjuntos difusos sobre un universo de discurso dado X.

La información compilada será incorporada en implementaciones en tiempo real del controlador difuso.

Desde esta perspectiva, el control difuso se puede catalogar de una manera heurística y modular, para la definición de sistemas no lineales.

La figura 1, muestra la estructura general de un controlador difuso.

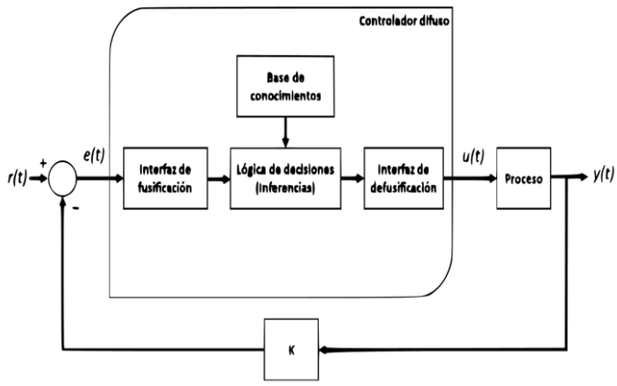

Figura 1 Estructura de un controlador difuso
Fuente: Elaboración propia

Analizando detalladamente la estructura del controlador difuso, se aprecian cuatro bloques que constituyen el funcionamiento correcto del controlador:

– *Interfaz de fusificación*: traduce los valores nítidos de entrada a una representación difusa incorporando la incertidumbre y la impresión del lenguaje natural, para un procesamiento adicional en el controlador.

– *Base de conocimientos*: este bloque es implementado para especificar las reglas de control que comprende el conocimiento del dominio de la aplicación y los objetivos de control del operador, y una vez aquirido el conocimiento sobre el sistema a analizar, este es codificado y decodificado en el bloque de la lógica de desiciones.

– *Lógica de decisiones*: en este bloque se realizan las inferencias sobre un conjunto de reglas aplicadas a la entrada de un conjunto difuso, este bloque puede ser considerado como el núcleo del controlador.

ISSN 2523-6865

– *Interfaz de defusificación*: Los valores nítidos se generan a partir de la salida de un conjunto difuso presente en la interfaz de decisiones.

Funciones de membresía

La función de membresía $\mu(x)$ cuantifica una variable lingüística, que representa el grado de pertenencia de un elemento a un subconjunto definido por una etiqueta.

Y se encuentran en los procesos de fusificación y defusificación, entre las diferentes representaciones elegidas, para el diseño del controlador son: las del tipo triangular, trapezoidal y singleton.

La función de membresía triangular se encuentra definida por el límite inferior α, un límite superior β, y un valor modal m, tal que $\alpha < m < \beta$. Esta función es adecuada para modelar propiedades con un valor de inclusión distinto de cero y un rango de valores estrecho en torno a un punto b (Ibrahim, 2004). La ecuación 2, representa la función de membresía triangular.

$$\mu(x) = \begin{cases} 0 & si\ x \leq \alpha \\ \frac{x-\alpha}{m-\alpha} & si\ x\ \in (\alpha, m] \\ \frac{\beta-x}{\beta-m} & si\ x\ \in (m, \beta) \\ 0 & si\ x\ \geq\ \beta \end{cases} \quad (2)$$

Donde x representa el valor de entrada en su forma nítida. En la figura 2, se muestra la representación gráfica de la de membresía descrita en la ecuación (2).

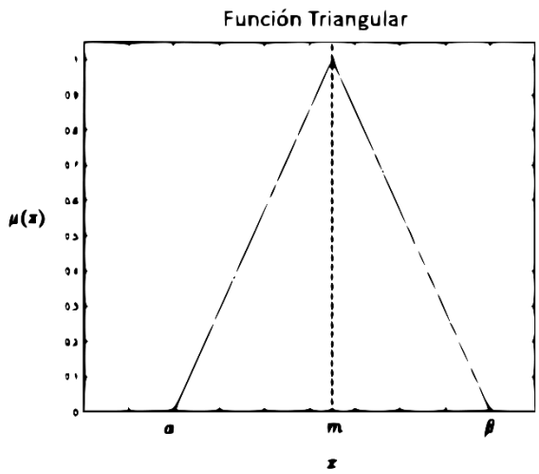

Figura 2 Función de membresía triangular
Fuente: Elaboración propia

Reglas del controlador difuso

La derivación de las reglas para el diseño del controlador difuso de temperatura es la base fundamental para el funcionamiento correcto del mismo.

Contiene la lógica de procesamiento de la información de entrada, para poder arrojar un valor de salida, que en este caso será la señal de control aún en su valor difuso. Para el diseño del controlador, se utilizan dos variables de entrada, *la temperatura de entrada y la temperatura deseada*, el grado de cardinalidad de este sistema difuso es 9 y está construido por las 6 variables lingüísticas establecidas en la tabla 1.

Temperatura de entrada	Temperatura deseada		
	Baja	**Normal**	**Alta**
Baja	S/C	Alta	Alta
Normal	Baja	S/C	Alta
Alta	Baja	Baja	S/C

Tabla 1 Reglas del controlador
Fuente: Elaboración propia

Las reglas del controlador definidas se proponen en términos del operador difuso "***and***" y se obtienen a partir de las reglas propuestas para el controlador de la tabla 1, como sigue:

– Si la temperatura de entrada es *Baja* y (*and*) la temperatura deseada es *Normal*, entonces la temperatura de salida debe ser *Baja*.

– Si la temperatura de entrada es *Baja* y (*and*) la temperatura deseada es *Alta*, entonces la temperatura de salida debe ser *Alta*.

– Si la temperatura de entrada es *Normal* y (*and*) la temperatura deseada es *Baja*, entonces la temperatura de salida debe ser *Baja*.

– Si la temperatura de entrada es *Normal* y (*and*) la temperatura deseada es *Alta*, entonces la temperatura de salida debe ser *Alta*.

– Si la temperatura de entrada es *Alta* y (*and*) la temperatura deseada es *Baja*, entonces la temperatura de salida debe ser *Baja*.

ISSN 2523-6865

– Si la temperatura de entrada es *Alta* y (*and*) la temperatura deseada es *Normal*, entonces la temperatura de salida debe ser *Baja*.

Defusificación

La interfaz de defusificación, es el proceso que se encarga del mapeo a escala que convierte el rango de valores difusos de las variables de salida a sus universos de discurso correspondientes. Es decir, transforma los valores difusos de la señal de control a su equivalente valor nítido de salida (Bai, Zhuang, & Wang, 2006). Existen diferentes métodos para defusificar la señal de control, los más utilizados son: el centro de gravedad, centro máximo y el método de altura.

El método utilizado en este artículo para la etapa de defusificación, es el del centro de gravedad, este método arrojó un valor defusificado de un conjunto difuso como un centroide difuso. En la ecuación (3), se muestra la estructura matemática discreta de este método.

$$z_0 = \frac{\sum_{x=a}^{b} \mu(x)_i \cdot W_i}{\sum_{x=a}^{b} \mu(x)_i} \quad (3)$$

Donde $\mu(x)_i$ es el grado de pertenencia en la salida del singleton i, W_i es el peso de salida del valor difuso de la salida del singleton i. La elección del método de defusificación depende del sistema embebido utilizado, debido a que existen diversos métodos que ocupan un alto nivel de procesamiento.

Implementación y Resultados

Para la implementación del proyecto se utilizó la herramienta computacional Raspberry Pi 3 y el lenguaje de programación Python 2.7. Y en el diseño del controlador se utilizó la librería pyFuzzy que puede ser descargada desde la terminal de la Raspberry. Esta librería permite el diseño de la interfaz de fusificación, la lógica de decisiones y la interfaz de defusificación de una manera fácil e intuitiva.

Para la aplicación del sistema de control se utilizó la metodología de diseño presentada en la figura 5, donde fueron propuestas dos variables de entrada al controlador, la temperatura de entrada y la temperatura deseada.

La temperatura de entrada, fue medida a partir de un sensor de temperatura digital DS18B20 y la temperatura deseada funciona como un *set-point* propuesto por el usuario del sistema.

El algoritmo de control, se diseñó a partir de las reglas presentadas en la sección 2.2 del controlador difuso. La Raspberry Pi 3, es la encargada de adquirir el valor de la temperatura a partir del sensor. Una vez que la temperatura se obtuvo, se procesó para convertirla a su correspondiente valor difuso, esta acción fue desarrollada por la interfaz de fusificación.

Con el valor difuso de la temperatura adquirido, se hizo posible implementar las reglas base del controlador con su lógica correspondiente. Y una vez obtenida la señal de control difusa, el siguiente paso, fue aplicar la defusificación para obtener la señal de control nítida, esta señal ya es comprendida por el sistema embebido.

La señal de control es enviada a los actuadores en forma de señal PWM (*Pulse-width-Modulation*), para regular el voltaje enviado a los actuadores en base a la señal de control aquirida. La figura 3, muestra un diagrama a bloques de la metodología de diseño para la aplicación del sistema.

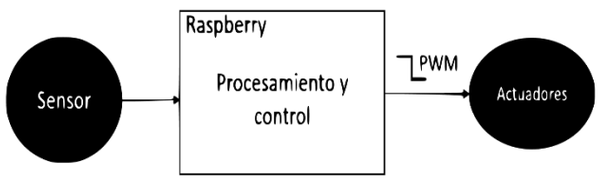

Figura 3 Metodología de diseño
Fuente: Elaboración propia

En la figura 4, se muestran las funciones de membresía basadas a partir de la tabla 2, donde se utilizaron funciones de membresía triangulares, ya que se necesitó un valor óptimo central, que fue definido por la temperatura deseada, además que son ligeras en términos de procesamiento. El rango de operación de la temperatura de entrada y deseada se determinó en el intervalo de 0°-50°C. Para las pruebas del controlador se obtuvo la señal de entrada a partir de la temperatura ambiente, donde el objetivo fue mantener la temperatura del sistema a 30°C.

ISSN 2523-6865

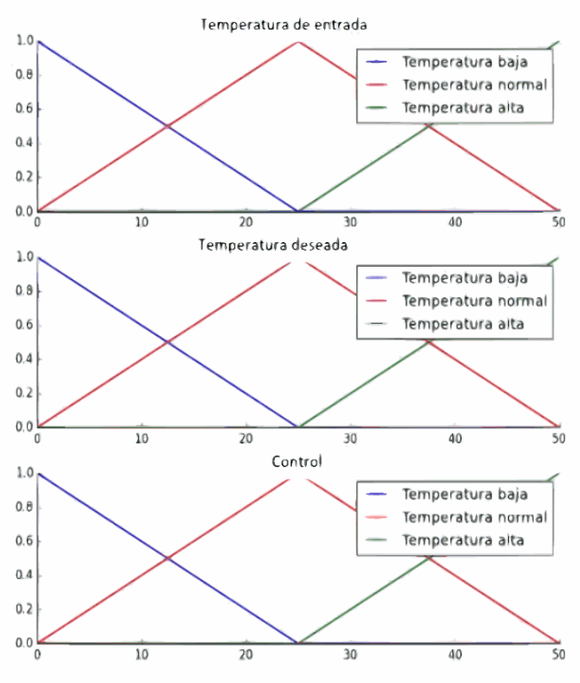

Figura 4 Funciones de membresía
Fuente: Elaboración propia

Se realizó una prueba de la etapa de fusificación, donde se propuso una temperatura de entrada nítida de 25°C con una temperatura deseada de 30°C. El valor difuso obtenido fue de 38.7 que es un valor adimensional. La figura 5, muestra el resultado gráfico obtenido por parte de la interfaz de fusificación.

Por otro lado, en la tabla 2, se muestra la conversión de la temperatura actual medida por el sensor, a su valor difuso. La temperatura deseada se mantuvo en 30°C.

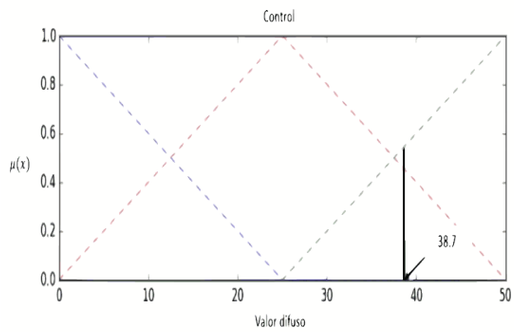

Figura 5 Resultado de la interfaz de fusificación para una temperatura de entrada de 25°C y una temperatura deseada de 30°C
Fuente: Elaboración propia

Analizando la tabla 2, se deduce que a 0°C el valor difuso tiende a ser el valor máximo, en otras palabras, se requiere aumentar la temperatura del sistema. Por el contrario, para una temperatura de entrada de 50°C, se obtiene un valor difuso mínimo. El rango de trabajo difuso oscila entre 8.61– 41.38.

Temperatura °C	Valor difuso
0	41.38
10	40.71
20	38.79
25	38.70
27	29.4
30	25.0
33	21.88
35	20.46
40	18.11
45	16.81
50	8.61

Tabla 2 Conversión de temperatura a su correspondiente difuso
Fuente: Elaboración propia

En la figura 6, se muestra la temperatura controlada, como se puede apreciar es un sistema lento, por lo que se utilizó un tiempo de muestreo de dos segundos.

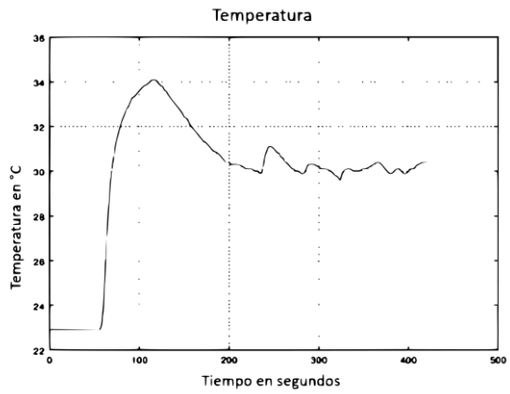

Figura 6 Temperatura controlada.
Fuente: Elaboración propia.

La figura 7, representa la señal de error, es fácil deducir que es una señal inversa a la temperatura controlada y la relación de la señal de error es pequeña con respecto a la temperatura real en el intervalo propuesto.

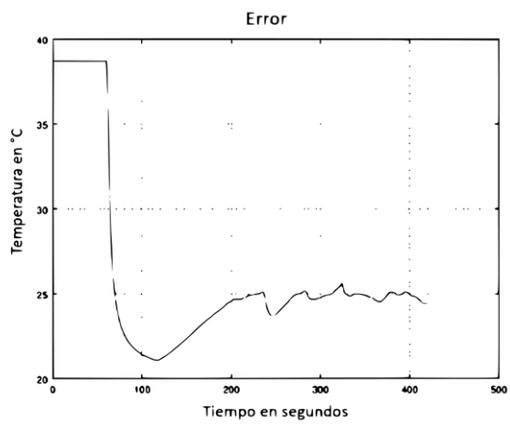

Figura 7 Señal de error
Fuente: Elaboración propia

ISSN 2523-6865

Agradecimiento

El primer autor da las gracias a la Universidad Autonóma de Querétaro y al CONACYT por el apoyo brindado para la realización de este proyecto.

Conclusiones

En este trabajo, se desarrolló un algoritmo de control difuso a partir de las características presentadas en la sección 2.2. El algoritmo de control de temperatura fue implementado en Rasperry Pi 3 con el lenguaje de programación Python 2.7.

El controlador basado en lógica difusa de temperatura presentado en este artículo, cumplió satisfactoriamente las consignas de diseño, medir la temperatura, procesar la información y enviar la señal de control a los actuadores. Los sistemas de control de temperatura son considerados como sistemas lentos, debido a que los cambios en la temperatura en ocasiones tienden a pasar desapercibidos, pero gracias a la complejidad del algoritmo desarrollado, este pudo ser usado para controlar la temperatura de un horno didáctico con cambios de temperatura rápidos, ya que la señal de control fue modulada por una señal PWM. El sistema de control con base en la Raspberry Pi 3, mostró resultados muy positivos, debido a que fue posible hacer la aqdquisicón, el procesamiento y control en tiempo real, ejectutar una interfaz gráfica para visualizar la temperatura actual del sistema. Por estas razones y su bajo costo, este tipo de controladores puede ser aplicado a sistemas no lineales.

Referencias

Bai, Y., Zhuang, H., & Wang, D. (2006). Advanced Fuzzy Logic Technologies in Industrial Applications. Springer.

Elnour, M., & Taha, W. I. M. (2013). PID and fuzzy logic in temperature control system. Proceedings - 2013 International Conference on Computer, Electrical and Electronics Engineering: "Research Makes a Difference", ICCEEE 2013, 172–177. https://doi.org/10.1109/ICCEEE.2013.6633927

Ibrahim, A. m. (2004). Fuzzy logic for embedded systems applications.

Isizoh, an, Okide, S., Anazia, A., & Ogu, C. (2012). Temperature Control System Using Fuzzy Logic Technique. International Journal of Advanced Research in Artificial Intelligence, 1(3), 27–31. Retrieved from http://www.getcited.org/pub/103504268

Jiang, W., & Jiang, X. (2012). Design of an intelligent temperature control system based on the fuzzy self-tuning PID. Procedia Engineering, 43, 307–311. https://doi.org/10.1016/j.proeng.2012.08.053

Li, X., Guan, Y., & Yuan, H. (2008). Curriculum development and progressive engineering practice design in embed system education. 2008 IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications, MESA 2008, 228–232. https://doi.org/10.1109/MESA.2008.4735752

Lin, H. H., & Hsueh, C. W. (2006). COS: A configurable OS for embedded SoC systems. Proceedings - 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA 2006, 242–245. https://doi.org/10.1109/RTCSA.2006.24|

Ponce, P. (2010). Inteligencia artificial con aplicaciones a la ingeniería. Alfaomega, México (Primera ed).

Ross, T. J. (2004). Fuzzy Logic with Engineering Applications. Wiley. https://doi.org/10.1002/9781119994374

Saii Yamanoor, N., & Yamanoor, S. (2016). High Quality, Low Cost Education With the Raspberry Pi, (46).

ISSN 2523-6865