

Diseño de biorreactor difuso

Diego Antonio Flores Solorzano¹, Gilberto Silos Chincoya¹,
Gonzalo Guillermo Martínez Oliva¹, Francisco Javier García Camacho¹,
Jesús Alberto Vázquez Santacruz¹, María Guadalupe Ramírez Sotelo²,
Agustín Ignacio Cabrera Llanos¹

¹ Instituto Politécnico Nacional,
Unidad Profesional Interdisciplinaria de Biotecnología,
Departamento de Bioprocesos,
México

² Instituto Politécnico Nacional,
Unidad Profesional Interdisciplinaria de Biotecnología,
Departamento de Bioingeniería,
México

chinco.sg12@gmail.com

Resumen. Este trabajo presenta el diseño y desarrollo del control de un biorreactor de flujo ascendente mediante algoritmos de lógica difusa aplicados mediante la programación e implementación de código en Python. El trabajo se dividió en cuatro etapas: Diseño de los controles difusos para cada variable de control en un biorreactor e implementación del programa, desde la creación de funciones para facilitar el análisis de la lógica difusa y el uso de funciones predefinidas de librerías de Python. El procedimiento realizado se describe como: primeramente, se diseñaron los controles difusos para las variables temperatura del medio, velocidad de agitación y pH, el algoritmo difuso utilizado fue el método de Mamdani, proceso que se realiza en cuatro pasos: La fuzzificación de las variables de entrada, definiendo primeramente las funciones de membresía para después ser agregadas las variables lingüísticas; la evaluación de las reglas de inferencia mediante la creación de la función “cut” para evaluación; agregación de las salidas de las reglas mediante la creación de la función “unión”, y la defuzzificación para poder obtener una respuesta de salida al sistema.

Palabras clave: Biorreactor, temperatura, ph, velocidad de agitación, lógica difusa, Python.

Fuzzy Bioreactor Design

Abstract. This work presents the design and development of the control of an upflow bioreactor through fuzzy logic algorithms applied through the programming and implementation of code in Python. The work was divided into four stages: Design of fuzzy controls for each control variable in a bioreactor and implementation of the program, from the creation of functions to facilitate the analysis of fuzzy logic and the use of predefined functions from Python libraries. The procedure performed is described as: first, the fuzzy controls were designed

for the variables temperature of the medium, stirring speed and pH, the fuzzy algorithm used was the Mamdani method, a process that is carried out in four steps: The fuzzification of the variables input, first defining the membership functions and then adding the linguistic variables; the evaluation of the rules of inference by creating the "cut" function for evaluation; aggregation of the outputs of the rules by creating the "union" function, and defuzzification in order to obtain an output response to the system.

Keywords: Bioreactor, temperature, ph, agitation speed, fuzzy logic, Python.

1. Introducción

Un biorreactor es un equipo capaz de simular medios de cultivo en estado sólido o líquido. En este dispositivo se debe garantizar la máxima conversión de materia prima en el producto de conversión, por lo que su participación es de vital importancia para este tipo de bioprocesos.

Debido a la incertidumbre que puede existir en un biorreactor por las variables que deben de controlar como el pH, temperatura, saturación de oxígeno, velocidad de agitación, la naturaleza compleja del crecimiento de microorganismos y la formación de productos en cultivo. El uso de modelos matemáticos o sistemas de control por retroalimentación suelen ser imprecisos, siendo una opción viable la lógica difusa teniendo como principal ventaja la facilidad de implementación en un sistema, ya que no es necesario conocer el modelo matemático o emplear matemática avanzada para describir el comportamiento que rige a dicho sistema para poder controlarlo, añadiendo el hecho de que en trabajos pasados donde se implementa este tipo de control se ha encontrado que han sido más eficientes en cuestión de la metodología para poder desarrollarlo de forma experimental; se puede dar la manipulación de varias variables del controlador como conjuntos difusos, factores de escalamiento, etc., junto con las simulaciones y el método de prueba y error.

La lógica difusa es utilizada en sistemas de control modernos siendo capaz de reaccionar a cambios continuos en el sistema y otorgar valores diferentes a la lógica booleana que es la lógica que utilizan los sistemas de control clásicos. El diseño de la lógica difusa permite interpretar y expresar el lenguaje ambiguo humano mediante funciones de pertenencia; en las cuales - como su nombre indica -, cada elemento de un universo dado pertenece a un conjunto específico.

Las funciones de pertenencia (o membresías), son asignadas con base en el rango en el cual se ejecutará dicho valor correspondiente a los valores de entrada de un control difuso, el cual puede ser de tipo SISO, MIMO (Single Input – Single Output, Multiple Input- Multiple Output), entre otros.

A partir de los valores de entrada, se obtienen diferentes valores de pertenencia para cada uno, a esto se le llama fuzzificación. De acuerdo con estos valores, se declaran las funciones de inferencia que, siguiendo condiciones si-entonces, dan como resultado otro valor difuso a la salida. Esta, por último, se defuzzifica mediante el método del centroide, el cual se basa en determinar el centro de gravedad del conjunto de salida.

Una función de membresía puede definirse como todos los conjuntos difusos que permiten medir el grado en que los objetos pertenecen a dichos conjuntos y que

satisfacen las propiedades definidas imprecisamente. Una función de membresía se define por tres propiedades:

- Núcleo: Es la región del universo caracterizado por un valor de membresía igual a 1 (membresía completa), en el conjunto.
- Soporte: Es la región del universo en el que la membresía tiene un valor distinto de 0 en el conjunto.
- Límites: Es la región del universo en donde una membresía contiene valores mayores que 0 pero menores de 1.

Python es el lenguaje de programación utilizado para programar la lógica difusa de los controles del biorreactor debido a varias razones. Una de ellas es porque es un lenguaje Open Source, lo cual significa que posee código abierto, librerías de funciones y brinda la facilidad para ejecutar los programas realizados en cualquier software reconozca este lenguaje, el cual es libre y gratuito para su descarga en múltiples plataformas tecnológicas; es útil para proyectos de programación orientada a objetos y de tarjetas de adquisición, lo cual, es bastante útil para implementar sistemas de embebido en objetos de interés como en el presente trabajo.

Python permite el desarrollo de páginas web, programación orientada a objetos y programación de tarjetas de adquisición. Es bastante útil para implementar sistemas de embebido en objetos de interés como un biorreactor.

2. Metodología

2.1. Conjuntos difusos

Los conjuntos difusos surgieron como una nueva forma de representar la imprecisión y la incertidumbre, permiten formalizar expresiones lingüísticas que típicamente contiene algún grado de ambigüedad. La teoría clásica de conjuntos se define un conjunto crisp A sobre X mediante la función característica de A como:

$$\mu_A(x) = \begin{cases} 1, & \text{si } x \in A, \\ 0, & \text{si } x \notin A. \end{cases} \quad (1)$$

Si la función de pertenencia para un valor dado de $\mu_A(x)$ toma el valor de 1, ese valor es un elemento conjunto A ; por el contrario, si $\mu_A(x)$ toma el valor de cero, no pertenece al conjunto A (González-Morcillo, C,2011).

Para un conjunto difuso, sin embargo, la cuestión de pertenencia de un elemento al conjunto no es cuestión de todo o nada, existen diferentes grados de pertenencia, La función de pertenencia puede tomar cualquier valor dentro del intervalo de $[0,1]$.

La función de pertenencia $\mu_A(x)$ de un conjunto difuso A es una función:

$$\mu_A: X \rightarrow [0,1]. \quad (2)$$

El conjunto difuso A se define como:

$$A = \{(x, \mu_A(x)) : x \in U, \mu_A(x) \in [0,1]\}. \quad (3)$$

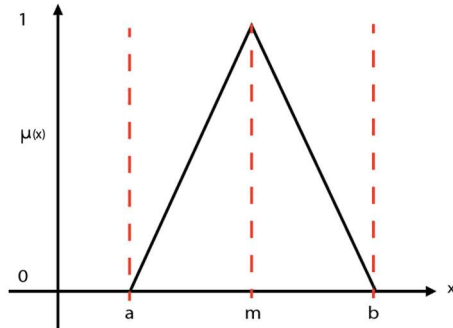


Fig. 1. Función Triangular.

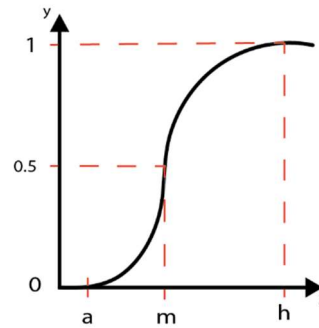


Fig. 2. Función Sigmoidal.

En cualquier elemento x en tiene un grado de pertenencia $\mu_A(x) \in [0,1]$.

2.2. Variables lingüísticas

Mientras que una variable algebraica toma números como valores, una variable lingüística toma palabras u oraciones como valores (Zadeh en Zimmermann 1993). El bloque fundamental de los sistemas basados en lógica difusa es la variable lingüística. La variable lingüística describe el razonamiento subjetivo que describen el contexto, es un medio de trasladar conceptos descripciones lingüísticas a descripciones numéricas. La forma de representar una variable lingüística esta descrita por cinco elementos:

$$(X, U, T(X), G, M),$$

donde X es el nombre de la variable, U el dominio subyacente, $T(X)$ es el conjunto de términos o etiquetas que puede tomar X , G es una gramática para generar las etiquetas $T(X)$: “Muy”, “No muy”, “Extremadamente”, “Bajo”, “Normal” y los conjuntos conectivos lógicos: operadores lógicos NOT, AND y OR y M es un regla semántica que asocia cada elemento de $T(X)$ con un conjunto difuso en U de entre todos los posibles.

2.3. Funciones de membresía

Las funciones de pertenencia que se utilizaron en este trabajo son las siguientes:

A. Función triangular

La función triangular (4) se describe de la siguiente manera:

$$\mu_{\text{Triangular}}(x) = \begin{cases} 0, & x < a, \\ \frac{x-a}{b-a}, & a \leq x \leq b, \\ \frac{c-x}{c-b}, & b \leq x \leq c. \\ 0, & x > c. \end{cases} \quad (4)$$

La representación gráfica se muestra en la Fig. 1.

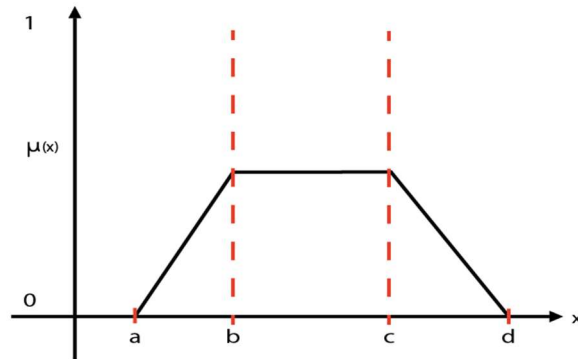


Fig. 3. Función Trapezoidal.

B. Función sigmoideal

La función sigmoideal (5) es descrita por:

$$\mu_{\text{Sigmoideal}}(x) = \begin{cases} 0, & x < a, \\ 2 \left(\frac{x-a}{m-a} \right)^2, & a \leq x \leq m, \\ 1 - 2 \left(\frac{x-a}{m-a} \right)^2, & m \leq x \leq h, \\ 1, & x > h. \end{cases} \quad (5)$$

La representación gráfica se muestra en la Fig. 2.

C. Función Zeta

La función zeta está definida (6) opuesta de la función sigmoideal:

$$\mu_{\text{Zeta}}(x) = 1 - \mu_{\text{Sigmoideal}}(x). \quad (6)$$

D. Función Trapezoidal

La función triangular (7) se describe de la siguiente manera:

$$\mu_{\text{Trapezoidal}}(x) = \begin{cases} 0, & x < a, \\ \frac{x-a}{b-a}, & a \leq x \leq b, \\ 1, & b \leq x \leq c, \\ \frac{d-x}{d-c}, & c \leq x \leq d, \\ 0, & x > d. \end{cases} \quad (7)$$

La representación gráfica se muestra en la Fig. 3.

E. Función Singleton

La función singleton **¡Error! No se encuentra el origen de la referencia.** tiene un valor único cuando $x=a$, se describe de la siguiente manera:

$$\mu_{\text{Singleton}}(x) = \begin{cases} 0, & x = a, \\ 1, & x \neq a. \end{cases} \quad (8)$$

2.4. Diseño de control difuso

Utilizando los algoritmos de lógica difusa se puede diseñar un sistema de control inteligente que permita decidir entre las diferentes variables lingüísticas de la señal de entrada y proporcione una señal de salida que permita el control de esta variable. El desarrollo de la estructura de un sistema de control difuso se realiza mediante tres etapas.

2.5. Etapa de fuzzificación

Esta etapa calcula el grado de pertenencia de la entrada del sistema dentro de cada una de las funciones de membresía definidas para las variables de entrada.

Una función de membresía indica el grado de pertenencia de cada elemento dentro de un conjunto delimitado de valores llamado universo.

2.6. Etapa de reglas de inferencia

El control difuso usa reglas lingüísticas sobre los resultados que fueron generado en la etapa de fuzzificación, las reglas difusas son sentencia SI-ENTONCES (IF-THEN) que describen la acción a ser tomada en respuesta a varias entradas difusas. Las reglas de inferencia especifican conclusiones extraídas de afirmaciones conocidas o asumidas como verdaderas, para hacer las reglas de inferencia se parte de lo siguiente: A y B son conjuntos difusos definidos en el universo X y Y respectivamente. Esto es una implicación donde el antecedente es “x es A” y el consecuente es “y es B”, la premisa A’ es ligeramente diferente de A y por lo tanto la conclusión B’ es ligeramente diferente de B. Se representa de la siguiente forma:

$$\frac{A' \quad A \rightarrow B}{B'}$$

El modus ponens generalizado. Sean A y A’ conjuntos difusos definidos en X, y sean B sea un conjunto difuso definido en Y. Entonces el conjunto difuso B’, inducido por “x es A” de la regla difusa:

$$\frac{x \text{ es } A' \quad \text{Si } x \text{ es } A \text{ entonces } y \text{ es } B}{y \text{ es } B'} \quad (9)$$

2.7. Etapa de defuzzificación

Esta etapa consiste en encontrar la salida final, las salidas que fueron encontradas en la etapa de reglas de evaluación modificaran a su respectiva función de pertenencia de salida. El método de defuzzificación utilizado es el Método de centro de Masa

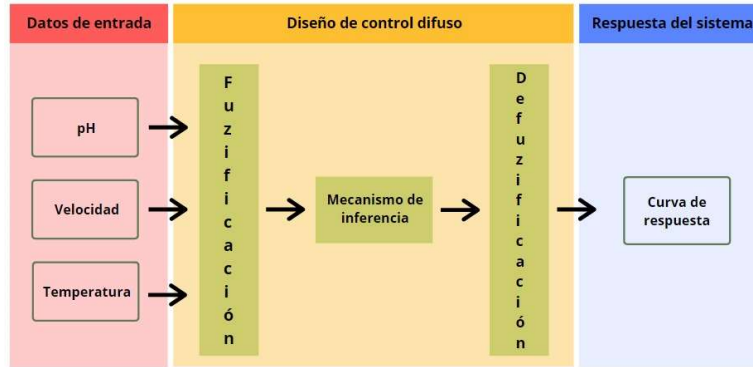


Fig. 4. Diagrama a bloques del algoritmo de programación.

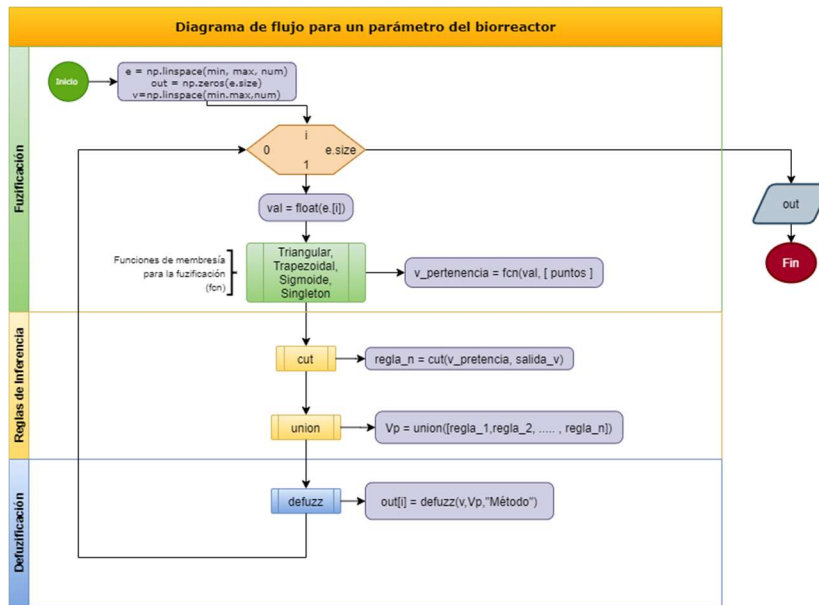


Fig. 5. Diagrama de flujo del control difuso.

(Centroide) (Silos et al., 2020), en el cual es determinado el centro de gravedad del conjunto de salida – se opta por este método debido a los resultados obtenidos en trabajos previos. Este método expresa el resultado final mediante un valor crítico o un promedio:

$$y_0 = \frac{\sum y\mu(y)}{\sum \mu(y)}. \quad (10)$$

Al valor obtenido con la expresión anterior permite el control de las variables temperatura, pH y velocidad de agitación, cada variable está determinado con su respectivo rango de valores.

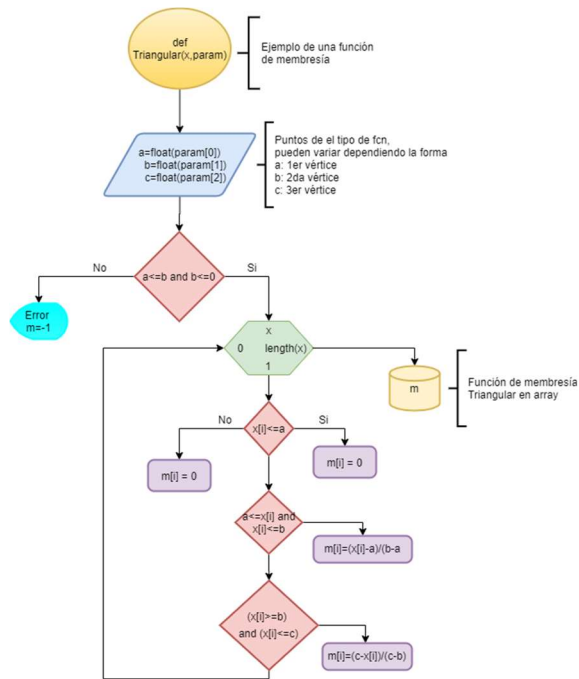


Fig. 6. Diagrama de flujo de la programación de una función de membresía.

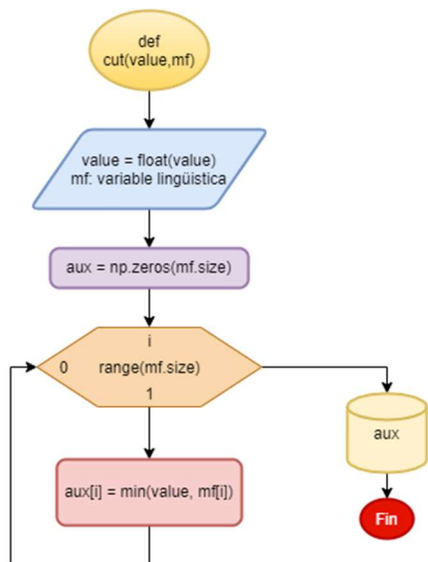


Fig. 7. Diagrama de flujo de la función “cut”.

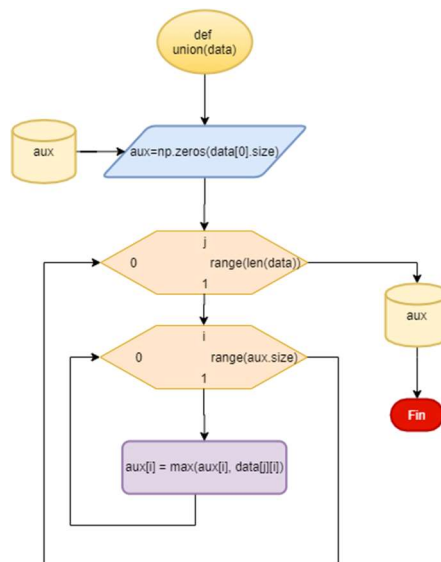


Fig. 8. Diagrama de flujo de la función “unión”.

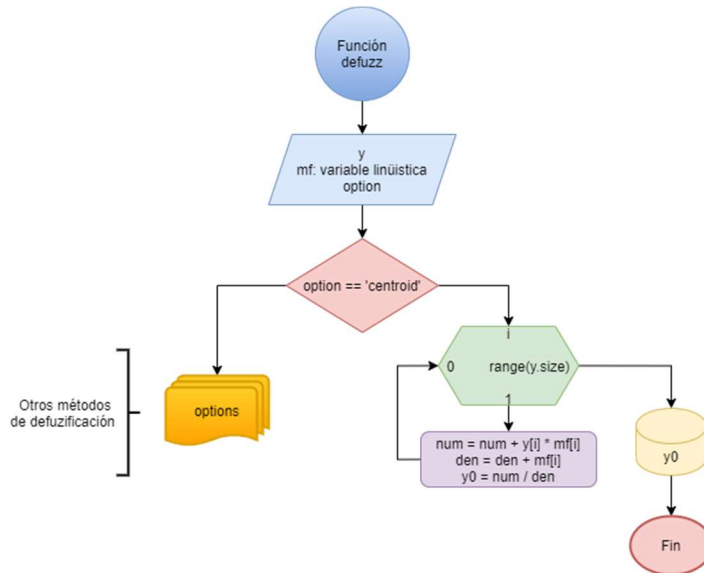


Fig. 9. Diagrama de flujo de la función “defuzz”.

2.8. Algoritmo de programación

El algoritmo que se presenta a continuación se desarrolló mediante el lenguaje de programación Python, se estructuró en cuatro etapas: Diseño de funciones de membresía, fuzzificación, reglas de inferencia y defuzzificación.

El código para el control difuso se representa en el diagrama de flujo de la Fig. 5. donde se crea un ciclo de lectura de las variables de entrada y se inicia el proceso de control difuso mediante las funciones de fuzzificación con las funciones de membresía, las funciones cut y unión para las reglas de inferencia y finalmente la función defuzz que tiene la labor de defuzzificar los valores de pertenencia con base en las reglas de inferencia y mediante los métodos de centroide, bisección, MOM, SOMO o LOM. Finalmente, el programa entrega un vector con los valores de salida del control difuso.

A partir de las ecuaciones mencionadas anteriormente, se describen las funciones de membresía para el proceso de fuzzificación. Para la implementación de la función triangular se programó mediante el siguiente diagrama de flujo en la Fig. 6., así mismo, se programaron las funciones de membresía restantes vistas en el punto 2.3.

Las reglas de inferencia se definieron mediante las funciones “cut”, que relaciona los valores de pertenencia con su valor fuzzificado, y “union” que se describe posteriormente. La función “cut” recibe los valores de las funciones de membresía de salida y un dato que recibe el nombre de valor de fuzzificación, este último delimita la función de membresía de salida, obteniendo como salida una función de membresía de salida cortada.

Se definió la función auxiliar “union”, los datos de entrada de la función fueron: las funciones de membresía de salida cortadas que se generaron en la etapa anterior.

A partir de la unión de todas las funciones de membresías de salidas cortadas, se obtiene un vector con los datos fuzzificados que son contrapuestos contra los valores del espacio de las funciones de membresía de salida. El método de defuzzificación es seleccionado a manera de string y la función fue programada con los procesos mencionados al principio de esta sección.

La salida de esta función son los valores para la curva de respuesta que representa el conjunto difuso de salida. Por último, para obtener un único dato de salida, se utiliza la fórmula del centroide y debido a que es un control de un biorreactor difuso y se han obtenido resultados óptimos bajo este método. (Silos et al., 2020).

A continuación, se muestra el programa realizado en Python para el diseño del control de velocidad de agitación, siguiendo este procedimiento manera se realiza el diseño de las variables temperatura y pH.

El pseudocódigo del programa de funciones de membresía velocidad de agitación:

Velocidad de agitación

```
#ENTRADA
x_rpm = np.linspace(-100,100,1001)
rpm_cero = Sigmoide(x_rpm,[-0.4,-50])
rpm_alto_n = Triangular(x_rpm,[-60,-40,-10])
rpm_medio_n = Triangular(x_rpm,[-40,-20,0])
rpm_bajo_n = Triangular(x_rpm,[-10,0,10])
rpm_bajo_p = Triangular(x_rpm,[0,20,40])
rpm_medio_p = Triangular(x_rpm,[10,40,60])
rpm_alto_p = Sigmoide(x_rpm,[0.4,50])
#SALIDA
x_corr = np.linspace(-0.05,0.05,1001)
corr_sin = Sigmoide(x_corr,[-600,-0.035])
corr_alta_n = Triangular(x_corr,[-0.04,-0.03,-0.01])
corr_media_n = Triangular(x_corr,[-0.03,-0.015,-0.005])
corr_baja_n = Triangular(x_corr,[-0.005,0,0.005])
corr_baja_p = Triangular(x_corr,[0.005,0.015,0.03])
corr_media_p = Triangular(x_corr,[0.01,0.03,0.04])
corr_alta_p = Sigmoide(x_corr,[800,0.035])
```

El código del Programa de todas las etapas:

```
e = np.linspace(-100, 100, 1001)
out = np.zeros(e.size)
v = np.linspace(-0.05,0.05,1001)
for i in range(e.size):
    val = float(e[i])
    # Fuzzyficando
    rpm_1 = Sigmoide(val, [-0.4, -50])
    rpm_2 = Triangular(val, [-60, -40, -10])
    rpm_3 = Triangular(val, [-40, -20, 0])
    rpm_4 = Triangular(val, [-10, 0, 10])
    rpm_5 = Triangular(val, [0, 20, 40])
    rpm_6 = Triangular(val, [10, 40, 60])
    rpm_7 = Sigmoide(val, [0.4, 50])
    ## Inferencia
    c_1 = cut(rpm_1, corr_alta_p)
```

```

c_2 = cut(rpm_2, corr_media_p)
c_3 = cut(rpm_3, corr_baja_p)
c_4 = cut(rpm_4, corr_sin)
c_5 = cut(rpm_5, corr_baja_n)
c_6 = cut(rpm_6, corr_media_n)
c_7 = cut(rpm_7, corr_alta_n)
# Se unen
Vp = union([c_1,c_2,c_3,c_4,c_5,c_6,c_7])
# Defuzzyficar
out[i] = defuzz(v, Vp, "centroid")

```

Tabla 1. Clasificaciones difusas para la señal de error de temperatura.

Etiqueta	Descripción	Parámetros
Negativo	Error muy negativo <0	Trapezoidal (-10,-10,-1,1.75)
Mínimo	Error mínimo cercano con un promedio 0	Triangular (0,1.75,3)
Intermedio Positivo	Error positivo con un promedio de 5	Triangular (0.5,5,7)
Máximo Positivo	Error muy positivo >7.5	Sigmoidal (2,6.5)

Tabla 2. Clasificaciones difusas para la señal de tiempo de encendido.

Etiqueta	Descripción	Parámetros
Apagado	No hay tiempo de encendido	Singleton (0)
Alto	tiempo de encendido con un promedio de 1 seg	Triangular (0,2,4)
Alto Medio	tiempo de encendido con un promedio de 2.5 seg	Triangular (1,3,5)
Medio	tiempo de encendido con un promedio de 5 seg	Triangular (3,5,7)
Bajo Medio	tiempo de encendido con un promedio de 7 seg	Triangular (5,7,9)
Bajo	tiempo de encendido con un promedio >8 seg	Sigmoidal (3.4,8.1)

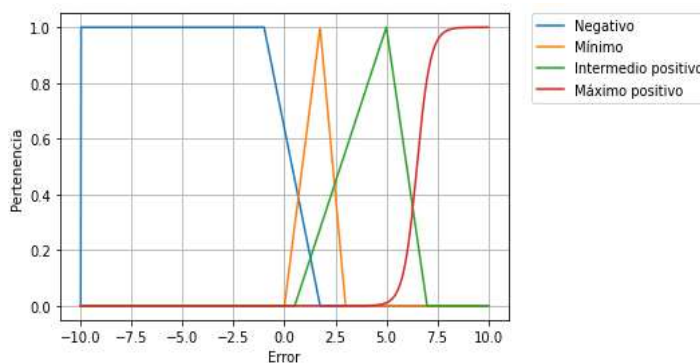


Fig. 10. Funciones de membresía de la variable de entrada “Error” del sistema difuso para el control de temperatura.

Tabla 4. Clasificaciones difusas para la señal de Voltaje PWM.

Etiqueta	Descripción	Parámetros
Vol Bajo	PWM con un promedio <80	Zeta (-0.6,85)
Vol Medio Bajo	PWM con un promedio de 100	Triangular (80,100,120)
Vol Regular	PWM con un promedio de 130	Triangular (110,130,160)
Vol Medio	PWM con un promedio de 160	Triangular (145,160,190)
Vol Medio Alto	PWM con un promedio de 210	Triangular (180,210,240)
Vol Muy Alto	PWM con un promedio 230	Sigmoidal (0.5,235)

Tabla 5. Clasificaciones difusas para la señal de error de la velocidad de agitación.

Etiqueta	Descripción	Parámetros
Alto Negativo	error con un promedio < -80	Zeta (-0.4,-50)
Medio Negativo	error con un promedio de -35	Triangular (-60,-40,-10)
Bajo Negativo	error con un promedio -20	Triangular (-40,-20,0)
Cero	error con un promedio 0	Triangular (-10,0,10)
Bajo Positivo	error con un promedio 25	Triangular (0,20,40)
Medio Positivo	error con un promedio 40	Triangular (10,40,60)
Alto Positivo	error con un promedio > 60	Sigmoidal (0.4,50)

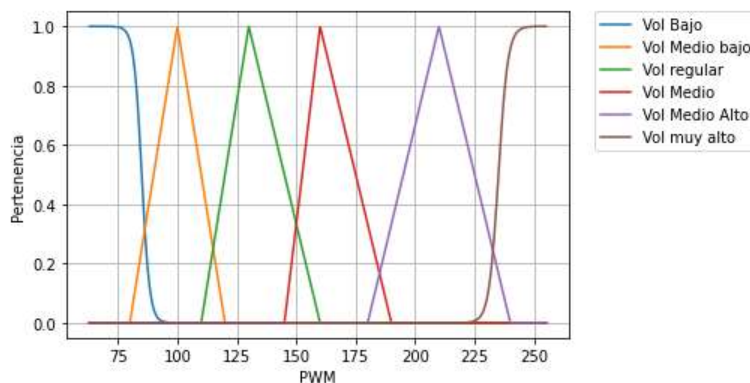


Fig. 13. Funciones de membresía de la variable de salida “Voltaje PWM” del sistema difuso para el control de pH.

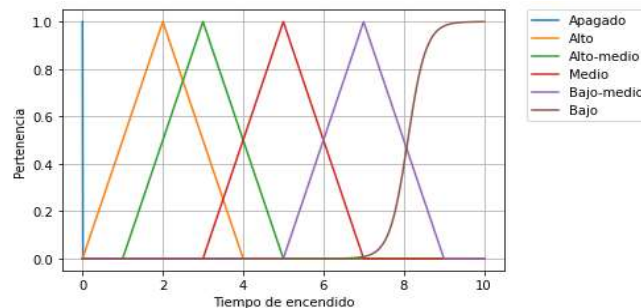


Fig. 11. Funciones de membresía de la variable de salida “Tiempo de encendido” del sistema difuso para el control de temperatura.

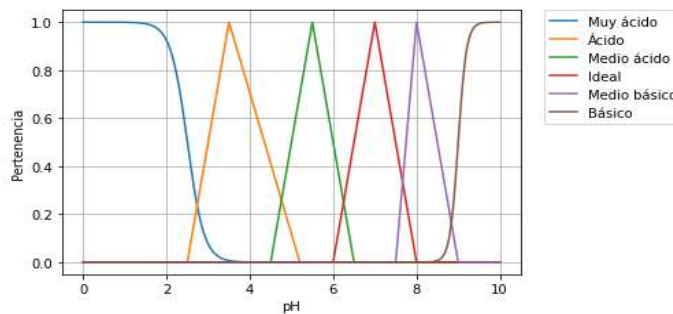


Fig. 12. Funciones de membresía de la variable de entrada “pH” del sistema difuso para el control de pH.

Tabla 3. Clasificaciones difusas para la señal de pH en el medio.

Etiqueta	Descripción	Parámetros
Muy Ácido	pH con un promedio < 2	Zeta (-5,2,5)
Ácido	pH con un promedio de 3	Triangular (2.5,3.5,5.2)
Medio Ácido	pH con un promedio de 5	Triangular (4.5,5.5,6.5)
Ideal	pH con un promedio de 7	Triangular (6,7,8)
Medio Básico	pH con un promedio de 8	Triangular (7.5,8,9)
Básico	pH con un promedio > 9	Sigmoidal (10,9)

2.9. Diseño de control difuso de temperatura

Se estableció un punto de referencia indicando la temperatura deseada ingresada por el usuario. La señal de entrada al sistema es el error calculado mediante la diferencia entre la señal deseada y la señal medida, usando este parámetro se diseñaron las

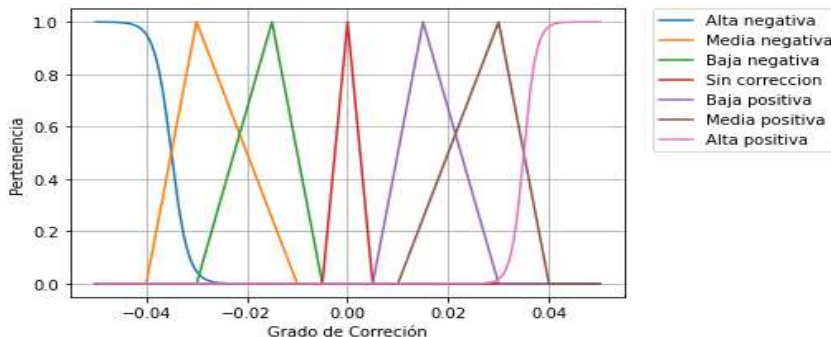


Fig. 15. Funciones de membresía de la variable de salida “Grado de corrección” del sistema difuso para el control de la velocidad de agitación.

Tabla 7. Reglas de inferencia de la variable temperatura.

Reglas
Reglas 1: IF “Error” IS “Negativo” THEN “Tiempo de Encendido” IS “Apagado”
Reglas 2: IF “Error” IS “Mínimo” THEN “Tiempo de Encendido” IS “Alto”

Tabla 8. Reglas de inferencia de la variable pH.

Reglas
Reglas 1: IF “pH” IS “Ideal” THEN “Voltaje PWM” IS “Voltaje-Medio”
Reglas 2: IF “pH” IS “Básico” THEN “Voltaje PWM” IS “Voltaje-Muy-Alto”

Tabla 9. Reglas de inferencia de la variable velocidad de agitación.

Reglas
Reglas 1: IF “Error” IS “Cero” THEN “Tiempo de Encendido” IS “Apagado”
Reglas 2: IF “Error” IS “Bajo-Negativo” THEN “Grado de corrección” IS “Baja-Positiva”

funciones de membresía de entrada para el algoritmo de control difuso cuyo rango de valores se encuentra en intervalo de -10 a 10, los parámetros de diseño se muestran en las Tabla 1.

El diseño de la señal de salida del sistema se contempla el tiempo de encendido de la resistencia para un intervalo de diez segundos, con este parámetro se diseñaron las funciones de membresía de salida para el control difuso, el rango de valores de salida se encuentra en el intervalo de 0 a 10 segundos. los parámetros de diseño se muestran en las Tabla 2.

2.10. Diseño de control difuso para pH

Las funciones de membresía de entrada se diseñaron a partir de la adquisición de la señal de pH del medio que se encuentra en el biorreactor usando este parámetro se

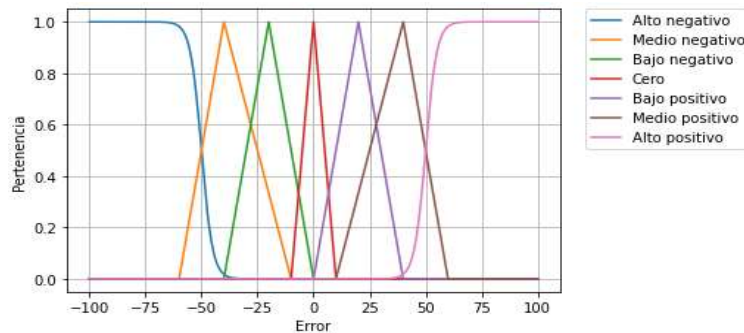


Fig. 14. Funciones de membresía de la variable de entrada “Error” del sistema difuso para el control de la velocidad de agitación.

Tabla 6. Clasificaciones difusas para la señal de grado de corrección.

Etiqueta	Descripción	Parámetros
Alto Negativo	error con un promedio < -80	Zeta (-0.4,-50)
Medio Negativo	error con un promedio de -35	Triangular (-60,-40,-10)
Bajo Negativo	error con un promedio -20	Triangular (-40,-20,0)
Cero	error con un promedio 0	Triangular (-10,0,10)
Bajo Positivo	error con un promedio 25	Triangular (0,20,40)
Medio Positivo	error con un promedio 40	Triangular (10,40,60)
Alto Positivo	error con un promedio > 60	Sigmoidal (0.4,50)

diseñaron las funciones clasificando la señal de pH cuyo rango de entrada se encuentra en el intervalo de 0 a 10 pH en el medio. los parámetros de diseño se muestran en las Tabla 3.

El control para la señal de salida es determinado por una bomba peristáltica regulando la Modulación de Ancho de Pulso (PWM), con este parámetro se diseña las funciones de membresía de salida del sistema difuso, clasificando la señal de modulación de ancho de pulso en seis variables lingüísticas, el rango de valores de salida se encuentra en el intervalo de 60 a 255 voltaje de modulación de ancho de pulso. Los parámetros de diseño se muestran en las Tabla 4.

2.11. Diseño de control difuso para velocidad de agitación

Se indica el punto de referencia indicando las revoluciones por minuto deseadas, se calcula el error mediante la diferencia entre la señal deseada y la señal adquirida. Se utilizó este parámetro para el diseño de las funciones de membresía de entrada clasificándolo en siete etiquetas, cuyo rango de entrada se encuentra en el intervalo de -100 a 100. Los parámetros de diseño se muestran en las Tabla 5.

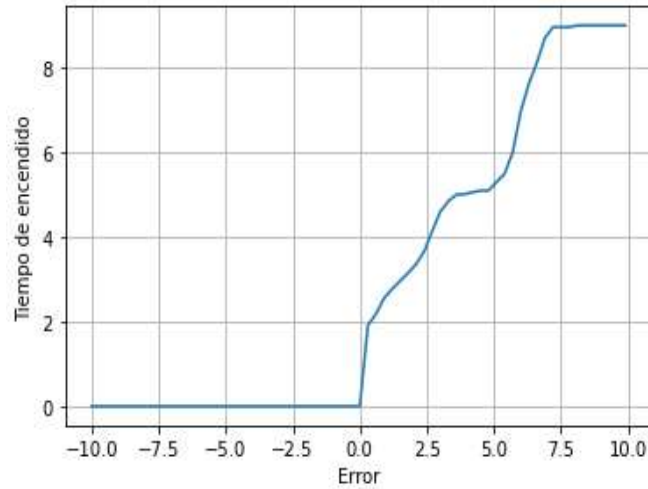


Fig. 16. Curva de respuesta del control difuso para la variable temperatura.

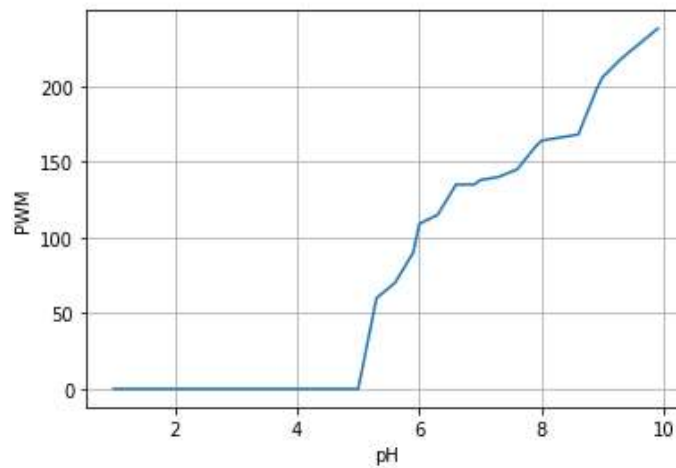


Fig. 17. Curva de respuesta del control difuso para el pH.

El diseño de las funciones de membresía para la señal de salida se estableció un grado de corrección de la Modulación del Ancho de Pulso (PWM), a partir de este parámetro de clasifica la señal de salida en siete variables lingüísticas, el rango de valores de salida se encuentra en el intervalo de -0.05 a 0.05 de grado de corrección. Los parámetros de diseño se muestran en las Tabla 6.

2.12. Diseño de reglas de inferencia

A continuación, se presenta las tablas de algunas reglas de inferencia desarrolladas.

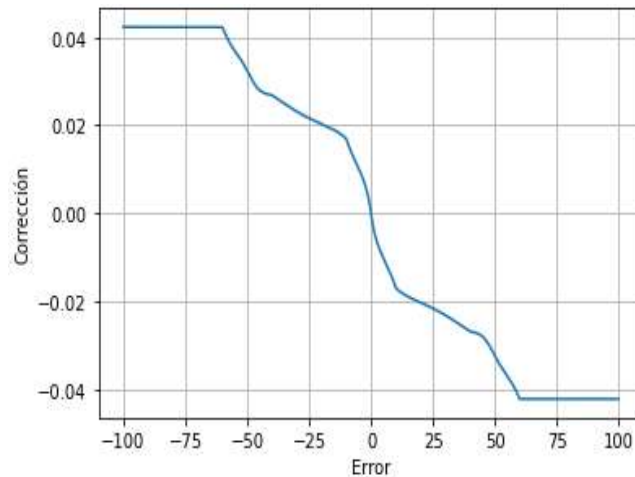


Fig. 18. Curva de respuesta del control difuso para velocidad de agitación.

3. Resultados

La respuesta obtenida para cada uno de los sistemas difusos se obtuvo mediante la relación establecida por las reglas de inferencia, se observa la curva de respuesta para la variable Temperatura en la Fig. 16.

En la Fig. 16. se muestra el funcionamiento del sistema difuso para el control de temperatura. Si el error de temperatura es negativo, implica que la medición de la temperatura sobrepasa el valor de referencia, por lo cual, la resistencia que permite el cambio de temperatura en el medio se mantiene apagada.

En el momento en el que el error se vuelve positivo, el sistema determina un tiempo de encendido de la resistencia que aumenta conforme aumenta el error para compensar el valor de la temperatura.

La respuesta del control difuso para el pH se observa en la curva de respuesta en la Fig.17.

En la Fig. 17. se muestra el comportamiento del sistema difuso para el control del pH en el medio. Si la señal de entrada permanece con un valor menor a 5, el ancho de modulación de pulso no genera una respuesta en el mecanismo de control de pH. En cambio, cuando se sobrepasa este umbral, aumenta en ciclo de trabajo conforme aumenta la entrada (ver Tabla 8).

En la Fig. 18. se muestra la respuesta del sistema difuso para el control de la velocidad de agitación. Si el error de velocidad es negativo, implica que el punto de referencia sobrepasa al valor medido, por lo cual, el grado de corrección del mecanismo es mayor. En caso contrario, si el error es positivo, el grado de corrección es bajo (ver Tabla 9).

4. Conclusiones

Se aplicó la teoría de lógica difusa a través del lenguaje de programación Python para diseñar un algoritmo que contenga las funciones necesarias para realizar los procesos de fuzzificación, inferencia y desfuzzificación para el diseño de tres sistemas difusos con la finalidad de controlar y monitorizar las variables de temperatura, pH y velocidad de agitación de un biorreactor. Se muestran las curvas de respuesta de los sistemas para mostrar las respuestas a los parámetros analizados en un biorreactor y demostrando que por este medio se pueden obtener las respuestas deseadas para el funcionamiento de un sistema control para un biorreactor. En este trabajo, se muestra la versatilidad del lenguaje Python para modelar sistemas de control difusos y proporciona las bases para desarrollar el modelado de manera física a través de tarjetas de adquisición, microprocesadores, microcontroladores que sean compatibles con el lenguaje Python.

Referencias

1. DisGonzález-Morcillo, C.: *Lógica Difusa, una introducción práctica*. Técnicas de Softcomputing (2011)
2. Fullér, R., Zimmermann, H.J.: On Zadeh's compositional rule of inference. In: *Fuzzy Logic*, pp. 193–200 (1993)
3. Kallestinova, E.: How to write your first research paper. *Yale Journal of Biology and Medicine*, 84(3), pp. 181–190 (2011)
4. Lee, K.D.: *Introduction. Python programming fundamentals*. Topics in Computer Science. Springer, London (2014)
5. Ruíz-Leza, H.A., Rodríguez-Jasso, R.M., Rodríguez-Herrera, R., Contreras-Esquivel, J.C., Aguilar, C.N.: Diseño de biorreactores para fermentación en medio sólido. *Revista Mexicana de Ingeniería Química*, pp. 33–40 (2007)
6. Jantzen, J.: *Foundations of fuzzy control*. John Wiley & Sons (2007)
7. Raich, V., Hooda, D.: *Fuzzy logic models and fuzzy control: An introduction*. Alpha Science International Ltd., pp. 2.1–2.3 (2017)
8. Hunt, J.: *A beginners guide to python 3 programming*. Springer Publishing (2019).
9. Silos-Chincoya, G., García-Estrada, H., Ramírez-Sotelo, M.G., Cabrera-Llanos, A.I.: Comparación de la variable de temperatura en un biorreactor: PID y lógica difusa. XVII Encuentro, Participación de La Mujer En La Ciencia (2020)