

Aplicación de un controlador difuso para un sistema de ventilación

Alexis Horacio López Fragoso

Universidad Autónoma de México,
Facultad de Estudios Superiores Cuautitlán,
Cuautitlán, Estado de México, México
guados123@gmail.com

Resumen En este proyecto se desarrollará e implementará un sistema de ventilación HVAC (que significa Heating, Ventilation and Air-conditioning) utilizando un controlador difuso. Con ayuda de MATLAB y las herramientas que proporciona Fuzzy Logic Toolbox se diseñará el controlador difuso aplicando dos entradas y dos salidas. Para las entradas se utilizará un Joystick, el cual ayudará a que el ventilador gire de manera horizontal. Para la segunda entrada se utilizará un sensor ultrasónico HC-SR04 para obtener información sobre la cercanía de un objeto. Cuando se detecta un elemento muy cerca del ventilador, éste se detiene. Para las salidas se utilizará un servomotor que permitirá controlar el giro del ventilador, y un motor que funcionará como ventilador mediante señales PWM. Se utilizará como interfaz física el Arduino UNO, que será la placa de desarrollo ideal para llevar a cabo nuestro sistema con controlador difuso. Este proyecto demostrará cómo es utilizar un controlador difuso en un sistema HVAC, proporcionando así una alternativa segura en utilizar este tipo de implementación.

Keywords: HVAC, Controlador difuso, MATLAB, Sensor HC-SR04, Joystick, Ventilador

1. Introducción

Los controladores difusos han dado de que hablar en los últimos años, suelen ser amenudo utilizados para una amplia gama de aplicaciones industriales y también para aplicaciones donde el confort para las personas es el protagonista. Es importante enterarse de las capacidades que pueden ofrecer los controladores difusos al abordar estos tipos de tareas, como es el control de los sistemas HVAC, como por ejemplo el control de un ventilador.

Los sistemas de calefacción, ventilación y aire acondicionado son fundamentales en garantizar el confort térmico y la calidad del aire en diversos entornos, ya sean residenciales, comerciales o industriales. En este mundo donde la eficiencia y la sostenibilidad son cada vez más importantes, optimizar el funcionamiento de estos sistemas se ha convertido en algo importante [1].

Controlar de manera efectiva los sistemas HVAC presenta varios desafíos. La naturaleza compleja y no lineal de la temperatura, más las impredecibles

condiciones ambientales y las diferentes necesidades de las personas, hacen que los métodos de control actuales a veces resulten insuficientes [2]. Los controladores tradicionales suelen depender de modelos matemáticos precisos, los cuales pueden ser difíciles de obtener o inadecuados para manejar la incertidumbre y la variabilidad de los sistemas físicos.

La lógica difusa, propuesta por Zadeh en 1965 [3], emerge como una alternativa eficaz. Al permitir trabajar con información imprecisa y manejar la incertidumbre, los controladores difusos facilitan la modelación de sistemas complejos a través de reglas lingüísticas y conjuntos difusos. Esto resulta en un control más adaptable y robusto frente a las variaciones del entorno [4].

El presente proyecto tiene como objetivo diseñar e implementar un sistema de ventilación controlado mediante un controlador difuso. Utilizando herramientas como MATLAB y Fuzzy Logic Toolbox, se desarrollará un controlador que integrará dos entradas y dos salidas, haciendo uso de sensores conectados a una placa de arduino UNO.

2. Estado del Arte

El control de sistemas HVAC ha sido un campo de estudio de interés debido a la necesidad de mejorar la eficiencia y el confort de las personas. A lo largo de las últimas décadas, se han explorado diversas técnicas de control, como lo es la lógica difusa, la cual es una de las mejores por su capacidad para manejar la incertidumbre y la no linealidad de distintos sistemas.

2.1. Aplicaciones de la Lógica Difusa en HVAC

La lógica difusa, introducida por Zadeh en 1965 [3], ha sido aplicada exitosamente en el control de sistemas HVAC debido a su habilidad para trabajar con información imprecisa y variables lingüísticas. Mamdani y Assilian [5] fueron pioneros en la aplicación de controladores difusos, demostrando su eficacia en sistemas complejos.

En el contexto de HVAC, Kusiak et al. [6] implementaron un controlador difuso para optimizar el consumo energético en edificios comerciales, logrando reducir significativamente el uso de energía sin comprometer el confort térmico. Asimismo, Ruano et al. [7] desarrollaron un sistema de control difuso para la gestión de la climatización en edificios inteligentes, mostrando mejoras en la adaptabilidad y eficiencia del sistema.

2.2. Comparación con Métodos de Control Convencionales

Los controladores PID y los modelos basados en ecuaciones diferenciales han sido utilizados en sistemas HVAC. Sin embargo, estos enfoques requieren modelos matemáticos precisos y pueden ser ineficaces ante cambios en las condiciones del entorno [8]. En contraste, los controladores difusos pueden adaptarse dinámicamente y manejar la no linealidad y las incertidumbres inherentes a los sistemas HVAC.

Wang y Ma [9] realizaron una comparación entre controladores convencionales y difusos en sistemas de aire acondicionado, concluyendo que la lógica difusa ofrece un mejor desempeño en términos de estabilidad y respuesta a perturbaciones.

2.3. Herramientas y Tecnologías para Implementación

La disponibilidad de herramientas como MATLAB y su Fuzzy Logic Toolbox ha facilitado el diseño y simulación de controladores difusos. Además, plataformas de hardware como Arduino permiten la implementación física de estos controladores de manera accesible y económica [10].

En estudios recientes, se ha explorado la integración de sensores inteligentes y redes de comunicación para mejorar el control y monitorización de sistemas HVAC basados en lógica difusa [11].

2.4. Desafíos y Tendencias Actuales

A pesar de sus ventajas, la implementación de controladores difusos en sistemas HVAC presenta desafíos, como la definición adecuada de las funciones de membresía y reglas difusas. La tendencia actual se orienta hacia el uso de algoritmos de aprendizaje automático y optimización para automatizar y mejorar el diseño de los controladores [12].

La integración de sistemas HVAC con tecnologías de Internet de las Cosas (IoT) y el enfoque en edificios inteligentes también está impulsando el desarrollo de soluciones más avanzadas y eficientes.

3. Conocimientos Previos

Para el diseño de un controlador difuso es necesario adquirir conocimientos esenciales sobre la lógica difusa, para así poder aplicar a la perfección estos conocimientos en un sistema de control difuso. A continuación, se presentan fundamentos teóricos y matemáticos que serán útiles para este proyecto.

3.1. Lógica Difusa

La lógica difusa, introducida por Zadeh en 1965 [3], es una generalización de la lógica clásica que permite manejar conceptos de verdad parcial. Mientras que en la lógica clásica una proposición es verdadera o falsa (valor 1 o 0), en la lógica difusa puede tomar cualquier valor real entre 0 y 1, representando grados de pertenencia.

Un conjunto difuso A en un universo de discurso X se define mediante una función de membresía $\mu_A : X \rightarrow [0, 1]$, donde $\mu_A(x)$ indica el grado en que el elemento x pertenece al conjunto difuso A [13].

3.2. Controladores Difusos

Un controlador difuso es un sistema basado en reglas lingüísticas que utiliza la lógica difusa para mapear entradas a salidas, emulando el razonamiento humano [14]. La estructura general de un controlador difuso consta de cuatro componentes principales:

1. **Fuzzificación:** Transforma las variables de entrada *crisp* (precisas) en valores difusos mediante funciones de membresía.
2. **Base de Conocimiento:** Consiste en una base de datos (funciones de membresía) y una base de reglas (reglas difusas) que describen el comportamiento del sistema.
3. **Motor de Inferencia Difusa:** Procesa las reglas difusas y combina las condiciones de entrada para producir una salida difusa.
4. **Defuzzificación:** Convierte la salida difusa en un valor *crisp* para actuar sobre el sistema controlado.

3.3. Fuzzificación

La fuzzificación es el proceso de convertir las variables de entrada *crisp* en valores difusos. Esto se logra mediante funciones de membresía, que pueden ser de diferentes tipos: triangulares, trapezoidales, gaussianas, entre otras [?].

Por ejemplo, para una variable de entrada e , la función de membresía triangular se define como:

$$\mu_A(e) = \begin{cases} 0, & e \leq a \\ \frac{e-a}{b-a}, & a < e \leq b \\ \frac{c-e}{c-b}, & b < e \leq c \\ 0, & e > c \end{cases}$$

donde a , b y c son los puntos que definen la forma triangular.

3.4. Base de Reglas Difusas

La base de reglas consiste en un conjunto de reglas lingüísticas del tipo “*SI condicion ENTONCES accin*”. Cada regla relaciona las variables de entrada con las de salida mediante operadores lógicos difusos [15].

Por ejemplo:

Regla 1: *SI el error es positivo grande Y la derivada del error es positiva, ENTONCES el control es negativo grande.*

3.5. Motor de Inferencia Difusa

El motor de inferencia combina las reglas difusas para producir una salida difusa. Los métodos más comunes de inferencia son el de Mamdani y el de Takagi-Sugeno [16].

En el método de Mamdani, se utilizan operadores mínimos y máximos para las operaciones *AND* y *OR*, y el resultado es una agregación de las salidas de todas las reglas:

$$\mu_{\text{salida}}(y) = \max_i [\min(\mu_{\text{premisa}_i}(x), \mu_{\text{conclusión}_i}(y))]$$

donde i indica la i -ésima regla.

3.6. Defuzzificación

La defuzzificación es el proceso de convertir la salida difusa en un valor *crisp*. Existen varios métodos, siendo el más común el del Centroide o Centro de Gravedad (*CoG*):

$$y^* = \frac{\int_Y y \cdot \mu_{\text{salida}}(y) dy}{\int_Y \mu_{\text{salida}}(y) dy}$$

donde Y es el universo de discurso de la variable de salida [17].

4. Metodología

En este proyecto, se siguieron varias etapas para desarrollar e implementar el sistema de ventilación controlado mediante un controlador difuso. La metodología se divide en dos partes: el diseño del controlador difuso y la implementación del sistema físico utilizando arduino. A continuación, se describen detalladamente cada una de estas etapas.

4.1. Diseño del Controlador Difuso

El diseño del controlador difuso se realizó utilizando MATLAB y su herramienta *Fuzzy Logic Toolbox*. El proceso se muestra a continuación:

Definición de Variables de Entrada y Salida Se establecieron dos variables de entrada y dos variables de salida para el controlador difuso:

- **Entradas:**
- Posición del Joystick: Es la dirección en la que se mueve el ventilador horizontalmente.
- Distancia al Objeto: Con el sensor ultrasónico HC-SR04, indica la cercanía de un objeto al ventilador.

- **Salidas:**
- *Ángulo del Servomotor:* Controla la orientación horizontal del ventilador.
- *Velocidad del Ventilador:* Controla la velocidad del motor del ventilador mediante señales PWM.

Definición de Funciones de Membresía Para cada variable, se definieron funciones de membresía, a continuación se presentan las **funciones de membresía:**

- **Posición del Joystick:** Izquierda, Centro y Derecha.
- **Distancia al Objeto:** Cerca, Media y Lejos.
- **Ángulo del Servomotor:** Izquierda, Centro y Derecha.
- **Velocidad del Ventilador:** Baja, Media y Alta.

Creación de la Base de Reglas Difusas Se estableció una base de reglas que relaciona las entradas con las salidas. Por ejemplo:

- Regla 1: SI la Distancia del Objeto es **Cerca** ENTONCES la Velocidad del Ventilador es **Baja**.
- Regla 2: SI la Distancia del Objeto es **Medio** ENTONCES la Velocidad del Ventilador es **Media**.
- Regla 3: SI la Distancia del Objeto es **Lejos** ENTONCES la Velocidad del Ventilador es **Alta**.
- Regla 4: SI la Posicin del Joystick es **Izquierda** ENTONCES el Ángulo del Servomotor es **Izquierda**.
- Regla 5: SI la Posicin del Joystick es **Centro** ENTONCES el Ángulo del Servomotor es **Centro**.
- Regla 6: SI la Posicin del Joystick es **Derecha** ENTONCES el Ángulo del Servomotor es **Derecha**.

En total, se definieron 6 reglas para cubrir todas las combinaciones relevantes de entradas y salidas.

Implementación y Simulación en MATLAB Utilizando el Fuzzy Logic Toolbox, se implementó el sistema tipo Mamdani, para nuestro controlador difuso. Se realizaron pruebas para verificar el comportamiento del controlador probando las distintas entradas. En la sección de análisis de resultados se discutirá más sobre esto.

4.2. Implementación con el arduino

Se utilizó arduino para llevar a prueba en la vida real nuestro sistema de ventilación con el controlador difuso.

Componentes Utilizados

- **Arduino UNO:** Microcontrolador que usamos para usar los sensores.
- **Sensor Ultrasónico HC-SR04:** Para la medición de la distancia al objeto.
- **Joystick:** Para controlar el eje horizontal del servomotor.
- **Servomotor:** Para controlar la orientación horizontal del ventilador.
- **Motor DC:** Funciona como ventilador, controlado mediante señales PWM.
- **Shield arduino:** Para facilitar la conexión entre los distintos componentes.

Configuración y Conexión de los Componentes Se conectaron los componentes al Arduino UNO siguiendo las especificaciones técnicas:

- El sensor HC-SR04 se conectó a los pines digitales para generar y recibir señales de ultrasonido, trigger conectado en el pin D7 y el echo en el pin D8.
- El joystick se conecta a la entrada analógica A0 para leer la posición en el eje X.
- El servomotor se conecta al pin D9, generando señales PWM para controlar su posición.
- El motor DC se controla mediante el pin D3.

Integración entre MATLAB y Arduino Se utilizó una toolbox de MATLAB para conectar el software con el hardware de Arduino [18], esto nos permitió la comunicación entre el controlador difuso y el arduino. Con MATLAB pudimos leer los valores de los sensores, procesar las entradas con el controlador difuso y enviar las señales de control al servomotor y al motor DC.

Pruebas del sistema Se realizaron pruebas en diferentes escenarios para comprobar el funcionamiento del sistema:

- **Prueba 1:** Moviendo el joystick hacia la izquierda y derecha, se verificó que el servomotor se moviera de izquierda a derecha.
- **Prueba 2:** Acercando y alejando un objeto al ventilador, con el fin de comprobar que la velocidad del ventilador disminuyera.
- **Prueba 3:** Se evaluó la respuesta conjunta del sistema, con el sensor ultrasónico HC-SR04 y el joystick.

5. Análisis de Resultados

En esta sección de análisis de resultados se presentan los resultados obtenidos tras la implementación del controlador difuso en el sistema de ventilación. Se incluye la imagen del armado del circuito, una captura de pantalla del código ejecutándose en MATLAB, la interfaz del *Fuzzy Logic Toolbox* y gráficas que ilustran las funciones de membresía de las entradas y salidas.

5.1. Montaje del Circuito

El armado del sistema se realizó con el hardware de arduino y los componentes que se especificaron en la metodología. En la Figura 1, se muestra una imagen del circuito armado, donde se pueden observar los componentes: el Arduino UNO, el sensor ultrasónico HC-SR04, el joystick, el servomotor y el motor DC.

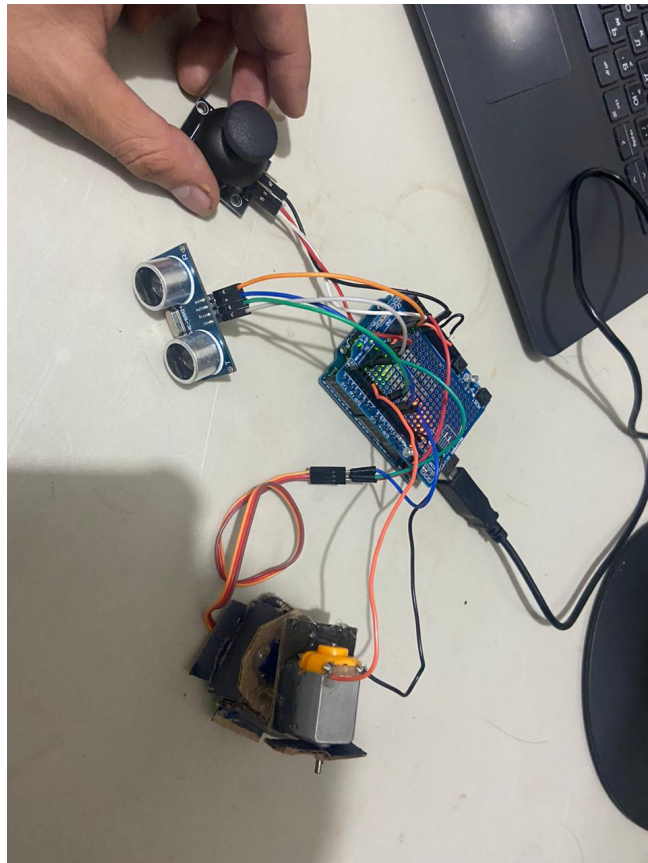


Figura 1: Armado del circuito del sistema de ventilación.

5.2. Ejecución del Código en MATLAB

El código se elaboró en MATLAB, nos ayudó a mantener la comunicación con el hardware arduino UNO y la implementación del controlador difuso. En la Figura 2, se presenta una captura de pantalla del código ejecutándose, donde podemos ver la lectura de los sensores.

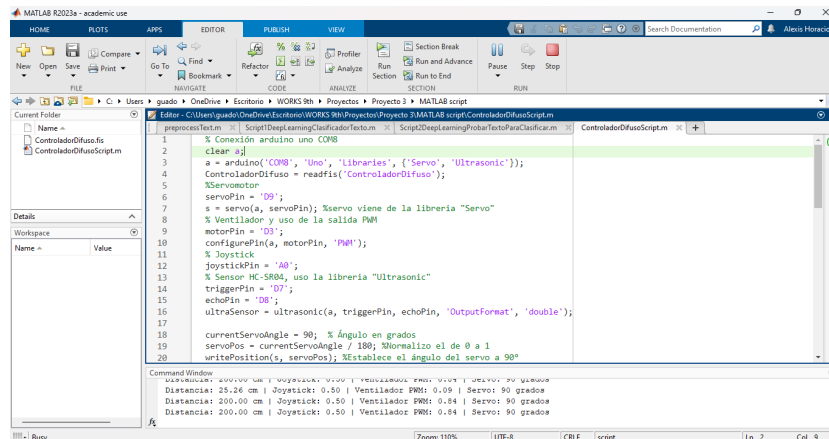


Figura 2: Código en MATLAB en ejecución.

5.3. Interfaz del Fuzzy Logic Toolbox

La configuración del controlador difuso se realizó utilizando el Fuzzy Logic Toolbox de MATLAB. En la Figura 3, se muestra la ventana del fuzzy logic donde se definieron las variables de entrada y salida.

5.4. Gráficas de las Funciones de Membresía

A continuación, se muestran las funciones de membresía para cada una de las variables:

Variable de Entrada: Distancia La Figura 4 muestra las funciones de membresía definidas para la variable de entrada *Distancia*. Se utilizaron tres conjuntos difusos: Cerca, Media y Lejos.

Variable de Entrada: Joystick En la Figura 5, se presentan las funciones de membresía para la variable de entrada *Joystick*. Los conjuntos difusos definidos son: Izquierda, Centro y Derecha.

Variable de Salida: Ángulo del Servomotor La Figura 6 muestra las funciones de membresía para la variable de salida *Ángulo del Servomotor*. Los conjuntos difusos utilizados son: Izquierda, Centro y Derecha, correspondientes a los ángulos de 0, 90 y 180 respectivamente.

Variable de Salida: Velocidad del Ventilador En la Figura 7, se ilustran las funciones de membresía para la variable de salida *Velocidad del Ventilador*. Se definieron los conjuntos difusos: Baja, Media y Alta.

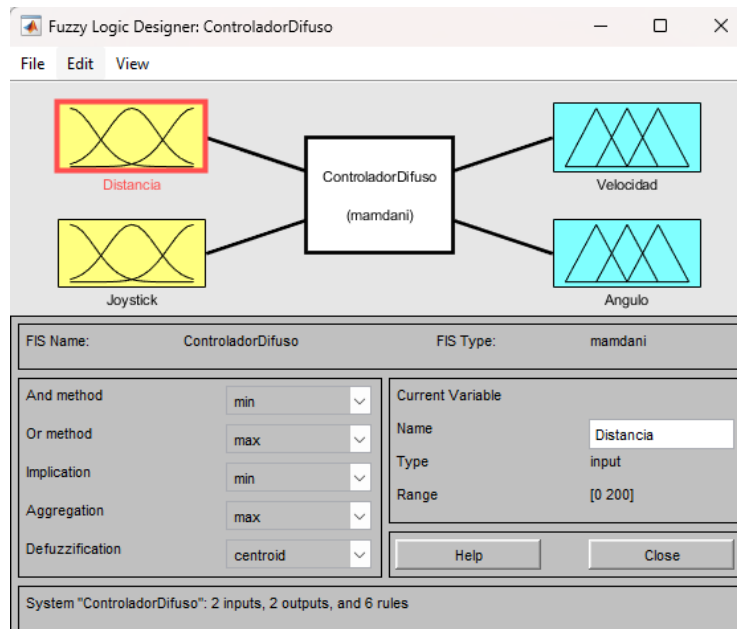


Figura 3: Interfaz del *Fuzzy Logic Toolbox* con el controlador difuso diseñado.

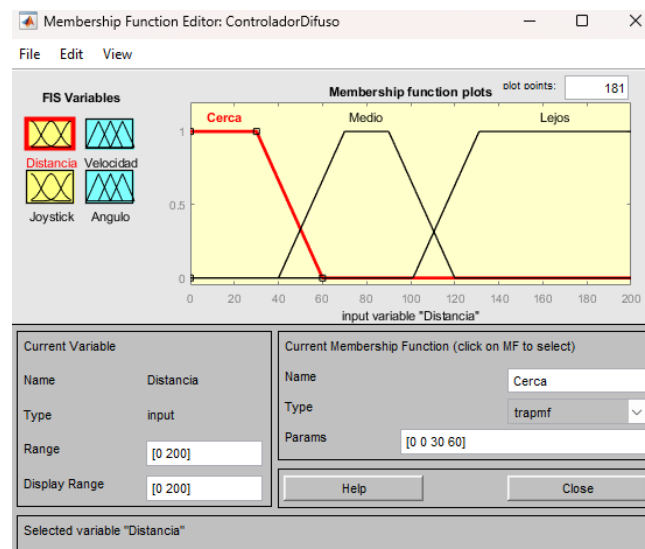


Figura 4: Funciones de membresía para la variable de entrada Distancia.

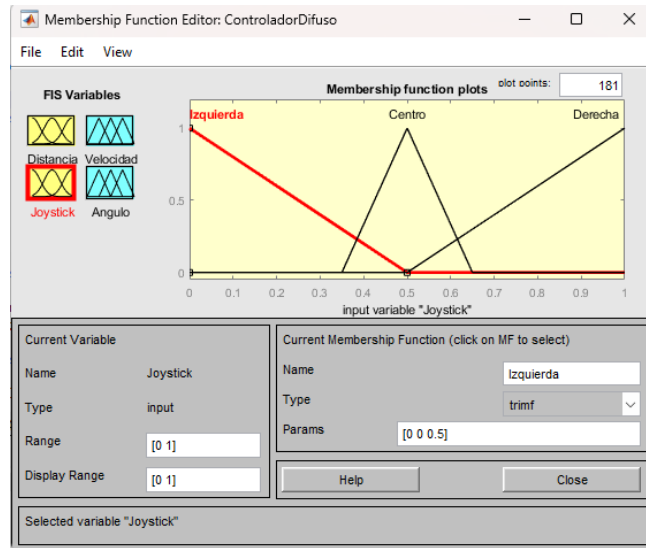


Figura 5: Funciones de membresía para la variable de entrada Joystick.

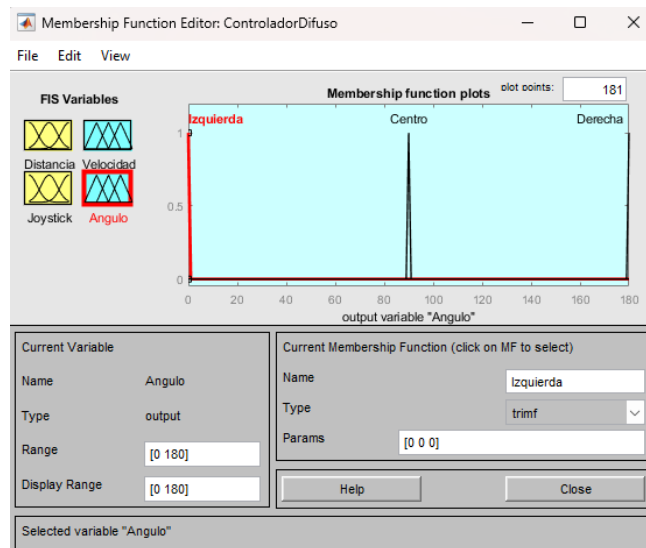


Figura 6: Funciones de membresía para la variable de salida *Ángulo del Servomotor*.

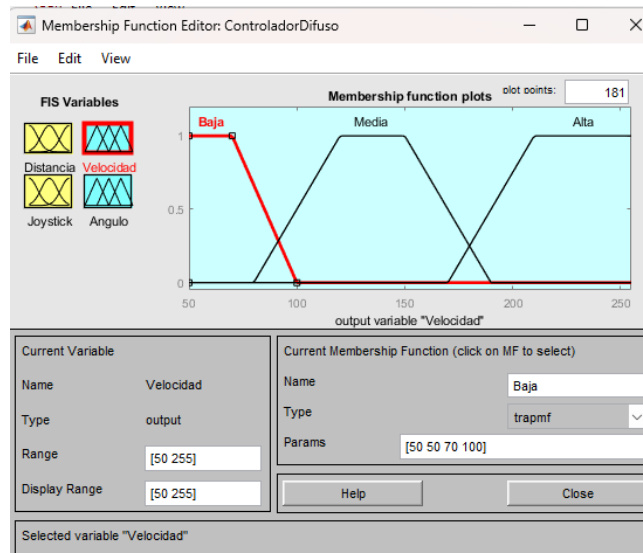


Figura 7: Funciones de membresía para la variable de salida Velocidad del Ventilador.

Análisis de las Funciones de Membresía Las funciones de membresía fueron diseñadas para reflejar el comportamiento esperado del sistema: Para la **Distancia**, se espera que cuando un objeto esté cerca, la velocidad del ventilador disminuya por seguridad. Las funciones de membresía permiten una transición suave.

En el caso del **Joystick**, las funciones de membresía triangulares permiten determinar la dirección en la que el usuario desea orientar el ventilador.

El **Ángulo del Servomotor** está relacionado con la posición del joystick, ajustándose a tres posiciones principales.

La **Velocidad del Ventilador** varía entre baja, media y alta, controlada tanto por la distancia al objeto.

Estas funciones de membresía hacen que el controlador difuso interprete correctamente las entradas y genere las salidas adecuadas, proporcionando un control eficiente del sistema de ventilación.

Análisis de las reglas difusas Las reglas difusas fueron establecidas para poder ser aplicadas en nuestro sistema de ventilación, usando únicamente 6 reglas para controlar nuestro sistema. La cantidad de reglas fueron suficientes para evaluar el funcionamiento de nuestro sistema. En la sección de metodología se establecieron las reglas difusas.

A continuación, en la Figura 8 se presentan las reglas difusas establecidas:

Las reglas mostradas son adecuadas para que el funcionamiento del sistema de ventilación pueda operar correctamente.

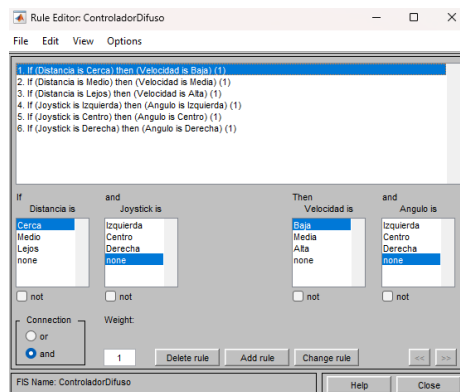


Figura 8: Reglas difusas para el Controlador Difuso

6. Conclusión

En este trabajo se implemento un sistema de ventilación usando un controlador difuso. Donde se implementaron diversas reglas y funciones de membresía para poder diseñar así nuestro controlador difuso de la mejor manera posible. Evaluando las dificultades que se tuvieron, fueron más que nada en el entendimiento de los conceptos teóricos aunados para la implementación de nuestro sistema, una vez comprendidos los conceptos, el diseño y la implementación resulto bastante sencilla.

Se utilizo Arduino UNO como interfaz física para poner a prueba nuestro sistema con controlador difuso. Para las entradas se usaron: un Joystick y un Sensor Ultrasónico HC-SR04. Para las salidas usamos un servomotor, el cual controlaría la dirección de movimiento horizontal de nuestro ventilador. Para la segunda salida, se implemento un motor de CD, que funcionaria como nuestro ventilador.

Se utilizaron 6 reglas difusas, las cuales se consideraron suficientes para evaluar el funcionamiento del sistema de ventilación. El código hecho en MATLAB nos permitió, además, mostrar los valores en tiempo real de nuestros componentes, utilizando la función *fprinf*. El sistema fue armado para evaluar su funcionamiento, paulatinamente se planteo en diseñar una base física en la que se encontraría el sistema con el controlador, pero por la falta de recursos y tiempo. Optamos por armar el sistema de manera rudimentaria, enfocándonos exclusivamente en la funcionalidad.

Este trabajo puede suponer un avance en la utilización de los sistemas difusos para ayudar a la mejora en ciertas áreas de la vida, como lo es la climatización, como lo pudimos ver en este trabajo. Los sistemas difusos representan un gran avance y junto con la combinación de la inteligencia artificial, en un futuro cercano conseguiremos ver esta aplicación en nuestra vida.

7. Anexos

A continuación, se incluyen el código utilizado en MATLAB para la importación del archivo con extensión `.fis` del controlador difuso generado al ser creado con la herramienta de fuzzy y la interacción con el Arduino UNO.

7.1. Código en MATLAB

```
1 % Conectar a la placa Arduino en el puerto COM8
2 clear a;
3 a = arduino('COM8', 'Uno', 'Libraries', {'Servo', 'Ultrasonic'
4     '});
5
6 % Configurar el servomotor
7 servoPin = 'D9';
8 s = servo(a, servoPin);
9
10 % Configurar el ventilador (motor) en un pin PWM (por ejemplo
11     , D3)
12 motorPin = 'D3';
13 configurePin(a, motorPin, 'PWM');
14
15 % Configurar el joystick en el pin analógico A0
16 joystickPin = 'A0';
17
18 % Configurar el sensor HC-SR04
19 triggerPin = 'D7';
20 echoPin = 'D8';
21 ultraSensor = ultrasonic(a, triggerPin, echoPin, '
22     OutputFormat', 'double');
23
24 % Inicializar el servomotor en 90 grados (posición media)
25 currentServoAngle = 90; % ángulo en grados
26 servoPos = currentServoAngle / 180;
27 writePosition(s, servoPos);
28
29 disp('Iniciando control difuso...');
30
31 while true
32     % Leer la distancia del HC-SR04 en centímetros
33     distance = readDistance(ultraSensor) * 100; % Convertir
34         a cm
35     distance = min(distance, 200); % Limitar a 200 cm
36         máximo
37
38     % Leer el valor del joystick (normalizado de 0 a 1)
```

```

36     joystickValue = readVoltage(a, joystickPin) / 5; %
        Arduino de 5V
37
38     % Crear vector de entrada para el FIS
39     inputs = [distance, joystickValue];
40
41     % Obtener las salidas difusas
42     outputs = evalfis(ControladorDifuso, inputs);
43
44     % Extraer las salidas
45     fanSpeed = outputs(1); % Velocidad del ventilador (50
        a 255)
46     servoAngle = outputs(2); % ngulo del servomotor (0,
        90 o 180)
47
48     % Convertir la velocidad del ventilador a valor PWM (0 a
        1)
49     pwmValue = (fanSpeed - 50) / (255 - 50); % Normalizar
        entre 0 y 1
50     pwmValue = max(0, min(pwmValue, 1)); % Asegurar que
        est entre 0 y 1
51     writePWMDutyCycle(a, motorPin, pwmValue); % Escribir PWM
52
53     % Movimiento en pasos de 10 grados
54     if currentServoAngle ~= servoAngle
55         if currentServoAngle < servoAngle
56             currentServoAngle = min(currentServoAngle + 10,
                servoAngle);
57         else
58             currentServoAngle = max(currentServoAngle - 10,
                servoAngle);
59         end
60
61         % Convertir el ngulo a valor entre 0 y 1 para
            writePosition
62         servoPos = currentServoAngle / 180;
63         writePosition(s, servoPos);
64     end
65
66     fprintf('Distancia: %.2f cm | Joystick: %.2f | Ventilador
        PWM: %.2f | Servo: %.0f grados\n', ...
67         distance, joystickValue, pwmValue, currentServoAngle)
        ;
68     pause(0.1);
69 end

```

Listing 1.1: Cdigo de controlador difuso en MATLAB

Referencias

1. X. Cao, X. Dai, and J. Liu, "Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade," *Energy and Buildings*, vol. 128, pp. 198–213, 2016.
2. S. Wang, *Intelligent Buildings and Building Automation*. Routledge, 2010.
3. L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
4. J. Jantzen, "Design of fuzzy controllers," Technical University of Denmark, Department of Automation, Tech. Rep., 1998.
5. E. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.
6. A. Kusiak, M. Li, and F. Tang, "Control of hvac with data mining and numerical optimization," *Applied Energy*, vol. 87, no. 10, pp. 3273–3285, 2011.
7. A. Ruano, E. Crispim, E. Conceição, and C. López, "Building temperature regulation using a new adaptive fuzzy pi controller," *Energy and Buildings*, vol. 38, no. 2, pp. 121–127, 2006.
8. M. Astrid, S. Khotimah, and S. Kusumadewi, "Comparison between conventional pid and fuzzy logic controller for liquid temperature control system," *International Journal of Computer Applications*, vol. 19, no. 9, pp. 18–24, 2011.
9. S. Wang and Z. Ma, "Building hvac control systemsrole of advanced control strategies," *Energy and Buildings*, vol. 41, no. 1, pp. 16–26, 2008.
10. A. Valdez, J. Ramírez, and O. Mendoza, "Fuzzy logic control for a dc-dc converter using arduino," in *2013 10th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*. IEEE, 2013, pp. 168–173.
11. J. Yang, D. Li, Z. Guo, and I. Gorton, "An energy-efficient heterogeneous multicore-based system for building hvac control," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 228–241, 2014.
12. A. Afshari, D. Friedrich, and A. Bruce, "A novel approach to hvac energy optimization using fuzzy logic controller and genetic algorithms," *Energy and Buildings*, vol. 133, pp. 349–366, 2017.
13. G. J. Klir and B. Yuan, *Fuzzy Set Theory: Foundations and Applications*. Prentice Hall, 1995.
14. K. M. Passino and S. Yurkovich, *Fuzzy Control*. Addison-Wesley Longman Publishing Co., Inc., 1998.
15. E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proceedings of the Institution of Electrical Engineers*, vol. 121, no. 12, pp. 1585–1588, 1974.
16. T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.
17. J. Jantzen, *Foundations of Fuzzy Control*. John Wiley & Sons, 2007.
18. MathWorks, *MATLAB Support Package for Arduino Hardware*, 2020, <https://www.mathworks.com/hardware-support/arduino-matlab.html>.