



Interfases
ISSN: 1993-4912
Universidad de Lima

Tejada Muñoz, Guillermo
Lógica difusa para regular la velocidad de un motor de corriente continua (CC)
Interfases, núm. 17, 2023, Enero-Junio, pp. 79-95
Universidad de Lima

DOI: <https://doi.org/10.26439/interfases2023.n017.6320>

Disponible en: <https://www.redalyc.org/articulo.oa?id=730178910005>

- ▶ [Cómo citar el artículo](#)
- ▶ [Número completo](#)
- ▶ [Más información del artículo](#)
- ▶ [Página de la revista en redalyc.org](#)



Sistema de Información Científica Redalyc
Red de Revistas Científicas de América Latina y el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso
abierto

LÓGICA DIFUSA PARA REGULAR LA VELOCIDAD DE UN MOTOR DE CORRIENTE CONTINUA (CC)

GUILLERMO TEJADA MUÑOZ

gtejadam@unmsm.edu.pe

<https://orcid.org/0000-0003-2799-6488>

Universidad Nacional Mayor de San Marcos, Perú

RESUMEN

Este estudio se enfoca en diseñar y describir el funcionamiento de un controlador difuso para regular la velocidad de un motor de corriente continua (CC), el cual es ampliamente utilizado en diversas áreas, como naves espaciales, maquinaria de defensa, robótica, trenes eléctricos, vehículos eléctricos y grúas. Los resultados del controlador difuso se compararon con los de un controlador proporcional integral derivativo (PID), que se sintonizó utilizando herramientas de MATLAB. Se midieron los transitorios de la curva de velocidad del motor, como el tiempo de subida (*rise time*), el tiempo de estabilización (*settling time*) y el sobreimpulso (*overshoot*). En términos de transitorios, el controlador difuso propuesto logró mejores resultados que el controlador PID sintonizado por MATLAB.

PALABRAS CLAVE: motor de corriente continua, controlador difuso, controlador PID, base de reglas, superficie de control

FUZZY LOGIC FOR REGULATING THE SPEED OF A DIRECT CURRENT (DC) MOTOR

ABSTRACT

ABSTRACT. This study focuses on designing and describing the operation of a fuzzy controller to regulate the speed of a direct current (DC) motor, widely used in spacecraft, defense machinery, robotics, electric trains, electric vehicles, and cranes. The article compares the results of the fuzzy controller with those of a proportional integral derivative (PID) controller, which was tuned using MATLAB tools. Transients of the motor speed curve, such as rise time, settling time, and overshoot, were measured. Regarding transients, the proposed fuzzy controller achieved better results than the PID controller tuned by MATLAB.

KEYWORDS: direct current motor, fuzzy controller, PID controller, rule base, control surface

1. INTRODUCCIÓN

El sistema de control difuso es un sistema de control inteligente que no necesita conocer el modelo matemático preciso del objeto controlado. El controlador difuso tiene una gran robustez y adaptabilidad, ya sea que el objeto que se va a controlar sea lineal o no lineal (Zhu et al., 2019).

Los motores, físicamente, consisten en un estator (campo estacionario) y un rotor (campo giratorio o armadura) y funcionan a través de la interacción del flujo magnético y la corriente eléctrica para producir velocidad de rotación y torque (Almatheel & Abdelrahman, 2017). En particular, los motores de corriente continua (CC) son ampliamente utilizados en la industria porque proporcionan un alto torque teniendo un tamaño reducido (Somwanshi et. al, 2019); por ejemplo, se emplean en los molinos de laminación, procesos químicos, trenes eléctricos, manipuladores robóticos y algunos electrodomésticos (Rajagiri et al., 2019).

El control de velocidad de motores de CC ha sido un área de aplicación para muchos algoritmos metaheurísticos durante la última década, ya que proporciona una plataforma de pruebas observable para evaluaciones y comparaciones de rendimiento. El controlador para el motor puede ser de cualquier tipo, por ejemplo, PID, PID de orden fraccional (FOPID), red neuronal (NN), lógica difusa (FL) o sistema de inferencia difuso basado en red adaptativa (ANFIS) (Ekinci et al., 2021).

Se han realizado numerosas investigaciones sobre lógica difusa aplicada al control de motores de CC, como las de Somwanshi et al. (2019), Tang y Cao (2018), y Sharma y Palwalia (2017). Estos estudios diseñan un controlador difuso con dos variables de entrada (error y cambio del error) y tres variables de salida (K_p , K_I y K_D), que representan las tres ganancias de un controlador proporcional integral derivativo (PID). Utilizando estas variables de salida, se genera una señal de control que regula el motor. Estos trabajos difieren entre sí en cuanto al contenido de la base de reglas y al tipo de funciones de pertenencia empleadas para las variables de entrada y de salida. Tienen la ventaja de ser adaptables a los cambios del sistema porque sus ganancias (K_p , K_I y K_D) son autosintonizables, pero la desventaja es que su implementación es compleja.

Por otro lado, existen pocos trabajos que diseñan un controlador difuso con dos variables de entrada (error y cambio del error) y una sola salida. Un ejemplo de esto es el estudio de Almatheel y Abdelrahman (2017), quienes usan dos tipos diferentes de funciones de pertenencia (las gaussianas para las variables de entrada y las triangulares para la variable de salida). Luego, comparan sus resultados con un controlador PID obtenido a través de MATLAB, aunque no se especifican los valores de (K_p , K_I y K_D) utilizados.

En todos los trabajos mencionados anteriormente, los criterios aplicados para la construcción de las bases de reglas no están lo suficientemente detallados.

Por estas razones, en el presente artículo se describe el procedimiento de diseño de un controlador difuso sencillo que tiene como objetivo regular la velocidad de un motor de CC. Este controlador utiliza dos variables de entrada: error y cambio del error, y una variable de salida. Se emplearon funciones de pertenencia triangulares para todas las variables, y se explica detalladamente la metodología para construir la base de reglas del controlador.

Para evaluar el rendimiento del controlador difuso, se lo comparó con un controlador PID cuyos parámetros fueron sintonizados usando las herramientas de *software* de MATLAB. Asimismo, se utilizó el Sistema de Inferencia Difuso (FIS) de MATLAB para editar y probar el controlador difuso. Para demostrar la eficacia del controlador difuso con respecto al controlador PID sintonizado con las herramientas de MATLAB, se empleó el simulador Simulink.

Este artículo está estructurado de la siguiente manera: la sección 2 describe la metodología, la sección 3 presenta los resultados del estudio y la sección 4, las discusiones; finalmente, la sección 5 corresponde a las conclusiones del trabajo.

2. METODOLOGÍA

La Figura 1 muestra el sistema de control diseñado con el propósito de igualar la velocidad del motor (ω_{sal}) a la velocidad de referencia (ω_{ref}). Para lograr este objetivo, se emplea un controlador difuso, del cual se espera que sea eficiente para llevar la velocidad del motor a su valor de referencia en el menor tiempo posible y evitando grandes diferencias sobre impulsos de velocidad. En consecuencia, se busca minimizar los transitorios de tiempo de respuesta (*rise time*), sobreimpulso (*overshoot*) y tiempo de establecimiento (*settling time*), los cuales se miden con relación a la curva que describe la velocidad del motor. La acción de control difuso (SF1) se genera a partir del valor de las entradas, que son el error (E) (la diferencia entre la velocidad del motor y su referencia) y el cambio en el error (dE) (la diferencia entre dos errores consecutivos).

Para mejorar la respuesta de la velocidad del motor (ω_{sal}) y eliminar el error de estado estacionario, es decir, la discrepancia constante en el estado estable entre la velocidad del motor (ω_{sal}) y la velocidad de referencia (ω_{ref}), se ha realizado una adición a SF1 con el valor de SF2, que representa la integral de SF1. Como resultado de esta operación, se obtiene la acción de control SF3. Esta estrategia busca corregir cualquier error persistente en la velocidad del motor y asegurar que se iguale a la velocidad de referencia después de que los transitorios (*rise time*, *overshoot* y *settling time*) hayan ocurrido.

La necesidad de incorporar la integración a la salida, como se mencionó previamente, se basa en el hecho de que un controlador difuso de dos entradas (E, dE) y una salida tiene una respuesta aproximada a un controlador proporcional derivativo (PD), el cual resulta

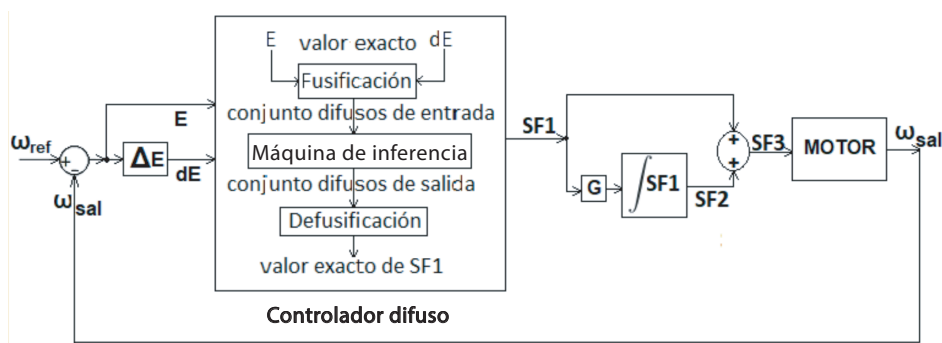
insuficiente para eliminar el error en estado estacionario. Por esta razón, al integrar la salida (al integrar a un PD), se genera una aproximación a un controlador proporcional integral (PI), capaz de eliminar el error en estado estacionario. Luego, al combinar ambos resultados (PD + PI), se consigue una aproximación de un controlador PID difuso. Es importante tener en cuenta que esta aproximación se debe a que la salida de un controlador difuso es no lineal, a diferencia de un controlador PID, que es lineal (Jantzen, 1998).

Posteriormente, mediante la variación de la ganancia G, se ha optimizado la curva de la velocidad del motor (ω_{sal}) en términos de los transitorios: *rise time*, *settling time* y *overshoot*. Esto se ha logrado con el objetivo de reducir al mínimo dichos transitorios en la respuesta del sistema.

Asimismo, se ha realizado una comparación entre el controlador difuso y un controlador PID cuando ambos manejan el mismo motor. Con el objetivo de garantizar que la comparación sea rigurosa, el controlador PID se ha diseñado empleando las herramientas de sintonización de MATLAB. La comparación entre ambos controladores se basa en determinar cuál de ellos genera en la curva de la velocidad (ω_{sal}) los valores más pequeños de los transitorios: *rise time*, *settling time* y *overshoot*.

Figura 1

Sistema de control difuso para controlar la velocidad de un motor de CC

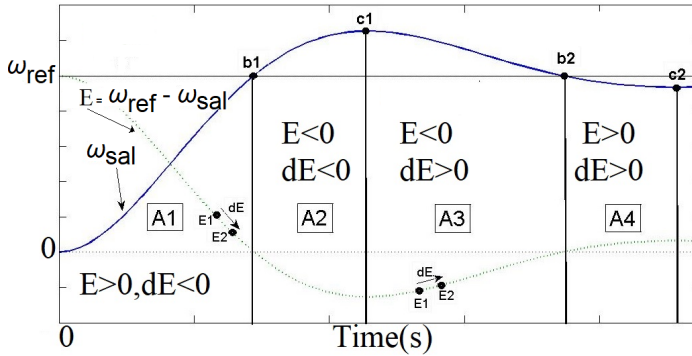


2.1 Zonas de control en un sistema de lazo cerrado

En nuestro proceso de diseño del controlador difuso, ha sido necesario tener en cuenta la respuesta típica de un sistema de control como el de la Figura 1. Esta respuesta típica se muestra en la Figura 2, donde se observa que entre el valor de la velocidad de referencia (ω_{ref}) y la curva de velocidad del motor (ω_{sal}) se pueden distinguir cuatro áreas (A1, A2, A3 y A4) delimitadas por los puntos de cruce (b1 y b2) y los puntos de pico y valle (c1, c2) (Li & Gatland, 1995). Dentro de las áreas se indican los signos que toman las variables E y dE, respectivamente.

Figura 2

Señales de ω_{sal} y E



Nota. Adaptado de "A new methodology for designing a fuzzy logic controller" (p. 508), por H. X. Li y H. B. Gatland, 1995, *IEEE Transactions on Systems, Man, and Cybernetics*, 25(3).

2.2 Modelo del motor

Un controlador basado en lógica difusa no requiere el modelo matemático preciso del objeto por controlar, en este caso, el motor. Sin embargo, fue necesario el modelo matemático del motor para simular la respuesta del controlador difuso y calcular los parámetros K_p , K_i y K_D de un controlador clásico PID, que también se ha utilizado para controlar el motor y servirá como punto de comparación.

La función de transferencia en Laplace, que relaciona la velocidad del motor $\omega(s)$ con la tensión de armadura $E_a(s)$ se muestra en (1) (Hekimoğlu, 2019):

$$PS = \frac{\omega(s)}{E_a(s)} = \frac{K}{JL_a s^2 + (BL_a + JR_a)s + (BR_a + KK_b)} \quad (1)$$

La tensión de armadura $E_a(s)$ es la generada por SF3 (véase la Figura 1) y es la que controla o regula la velocidad del motor. Para propósitos de simulación, se ha utilizado un motor de CC, como el mostrado en la Figura 3, de 620 RPM (64,93 rad/s) a 6 V. Los parámetros del motor son los siguientes:

R_a : resistencia de armadura, 1,9 (Ω)

L_a : inductancia de armadura, 0,00141859 (H)

J : momento inercia del rotor, 0,0001631 ($\text{kg} \cdot \text{m}^2$)

K : constante del torque del motor, 0,0880227 (Nm/A)

K_b : constante de fuerza contraelectromotriz, 0,0880227 (Vs/rad)

B : coeficiente fricción viscosa del motor, 0,00002711 (Nms/rad)

Figura 3

Motor de CC



Por lo tanto, reemplazando estos valores en (1), se tiene:

$$PS = \frac{\omega(s)}{E_a(s)} = \frac{0,08802}{0,0000002314 s^2 + 0,0003099 s + 0,0078} \quad (2)$$

2.3 Controlador PID sintonizado

Se ha utilizado un controlador PID para controlar la velocidad del motor y compararlo con el controlador difuso; la salida del controlador PID, representada por $u(t)$, es la que controla o regula la velocidad del motor. El controlador PID, en su forma de constante de tiempo y paralela, se expresa por (3) y (4), respectivamente.

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de}{dt} \right) \quad (3)$$

$$u(t) = K_p e(t) + K_I \int_0^t e(t) dt + K_D \frac{de}{dt} \quad (4)$$

Donde: K_p , K_I , K_D son las ganancias proporcional, integral y derivativa, mientras que T_i y T_d son los tiempos integral y derivativo, respectivamente. Siendo: $K_I = \frac{K_p}{T_i}$, $K_D = K_p * T_d$.

El error del sistema en el tiempo t , $e(t)$, se calcula restando de la velocidad de referencia (ω_{ref}) el valor de la velocidad $\omega_{sal}(t)$. La salida del controlador $U(s)$ en el dominio de Laplace está expresada por (5) (Johnson & Moradi, 2005):

$$U(s) = K_p + \frac{K_I}{S} + K_D S \quad (5)$$

Los valores, K_p , K_I y K_D pueden ser encontrados por varios métodos; por ejemplo, uno de los más utilizados es el de Ziegler-Nichols (Z-N). Sin embargo, no necesariamente

los valores encontrados con Z-N son los mejores, ya que necesitarán ser sintonizados. En este sentido, en este trabajo se ha preferido encontrar los parámetros del PID mediante las herramientas de sintonización de MATLAB (versión R2021a) mediante las siguientes líneas de comando:

- $C0 = \text{pidstd}(1,1,1)$; %Coloca valores iniciales $K_P=K_I=K_D=1$
- $U = \text{pidtune}(PS, C0)$; %Calcula K_P, K_I, K_D en función de la función de transferencia del motor (PS)
- $[K_P, K_I, K_D] = \text{piddata}(U)$; %Retorna los valores de K_P, K_I y K_D

Los valores encontrados son los siguientes:

$$K_P = 0,1210 \quad (6)$$

$$K_I = 6,1294 \quad (7)$$

$$K_D = 2,6531e-07 \quad (8)$$

Estos valores fueron utilizados en el controlador PID con el objeto de contrastar los resultados con los obtenidos por el controlador difuso.

2.4 Controlador difuso

El primer paso para diseñar el controlador difuso consiste en definir con precisión las variables lingüísticas, los conjuntos difusos y las funciones de pertenencia, como se describe detalladamente más adelante.

El controlador difuso, representado por el bloque "Controlador difuso" en la Figura 1, recibe del mundo real valores numéricos correspondientes a las variables lingüísticas de entrada (E, dE). Estos valores numéricos, también conocidos como *valores exactos*, son procesados dentro del controlador difuso, que a su vez genera un valor exacto en su salida, el cual tiene un impacto directo en el funcionamiento del motor. Las etapas del proceso llevadas a cabo dentro del controlador difuso (véase el bloque "Controlador difuso" de la Figura 1) son las siguientes:

1. Fusificación: en esta etapa, se toma el valor exacto de la entrada E y el valor exacto de la entrada dE; se generan los conjuntos difusos de entrada.
2. Máquina de inferencia: en esta etapa, se utilizan los conjuntos difusos de entrada para realizar inferencias y se generan los conjuntos difusos de salida.
3. Defusificación: en esta etapa, se toman los conjuntos difusos de salida y se genera un valor exacto que actúa sobre el motor.

Los términos como *valor exacto*, *conjuntos difusos de entrada* y *conjuntos difusos de salida* son comúnmente utilizados en el campo de la lógica difusa.

2.4.1 Variables lingüísticas, conjuntos difusos y funciones de pertenencia

El controlador difuso del presente trabajo tiene dos variables lingüísticas de entrada, que son el error (E) y el cambio del error (dE), y una variable lingüística de salida (SF1) (véase la Figura 1).

Para cada una de las variables lingüísticas, o simplemente variables (E, dE y SF1), se han considerado siete conjuntos difusos, que son los siguientes:

- N3: Muy negativo
- N2: Negativo
- N1: Ligeramente negativo
- Z: Cero
- P1: Ligeramente positivo
- P2: Positivo
- P3: Muy positivo

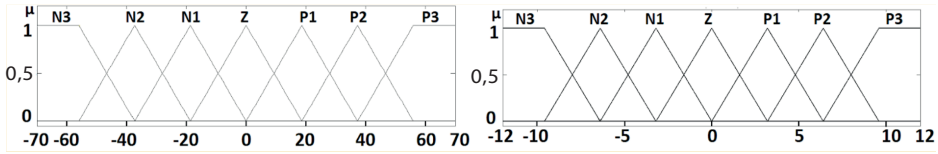
Las funciones de pertenencia de cada conjunto son triangulares, excepto en los extremos donde adoptan una forma trapezoidal. En la parte izquierda de la Figura 4, se presentan las funciones de pertenencia para los conjuntos de las variables E o dE (ambas son iguales), mientras que en la parte derecha se exhiben las funciones de pertenencia de los conjuntos de la variable SF1.

Se ha establecido un rango de -70 a 70 para el universo de discurso de las variables E (error) y dE (cambio del error). Esta elección se basa en reflejar una cifra cercana al error máximo cuando se establece la referencia de velocidad en su valor máximo ($64,93$ rad/s). Dado que la variable dE puede oscilar a un valor menor que E, se ha considerado apropiado utilizar el mismo rango de -70 a 70 para ambas variables. Esta decisión permite abarcar un amplio rango de posibles valores y asegurar que el controlador difuso pueda capturar las variaciones necesarias para un adecuado control del sistema.

En cuanto a la variable SF1, se ha utilizado un universo de discurso que abarca desde -12 V hasta 12 V. Esta elección representa el doble del intervalo de funcionamiento del motor, que va desde -6 V hasta 6 V. La razón detrás de esta decisión es obtener mejores resultados, ya que proporciona un rango más amplio para llevar a cabo los cálculos necesarios durante el proceso de defusificación de SF1. Al tener un rango más amplio, el controlador difuso puede realizar una mayor variedad de ajustes y adaptaciones en la salida, permitiendo un control más preciso y eficiente del sistema.

Figura 4

Funciones de pertenencia para las variables E y dE (izquierda) y la variable SF1 (derecha)



2.4.2 Fusificación

En la etapa de fusificación, se ha calculado el grado de pertenencia que tiene un valor exacto de la variable de entrada con una o varias de sus funciones de pertenencia, generando los conjuntos difusos de entrada. Así, por ejemplo, si la variable E presenta un valor exacto de 12 y la variable dE un valor exacto de -6 , entonces, de la Figura 5 y la Figura 6, respectivamente, se obtienen los siguientes los resultados, a los cuales se les denomina *conjuntos difusos de entrada*:

- E es Z con un grado de verdad de 0,35
- E es P1 con un grado de verdad de 0,65
- dE es N1 con un grado de verdad de 0,32
- dE es Z con un grado de verdad de 0,68

Entonces, solo están involucrados para la variable E los conjuntos difusos: Z y P1. Mientras que para la variable dE están involucrados los conjuntos difusos: N1 y Z.

Figura 5

Fusificación para $E = 12$ (rad/s)

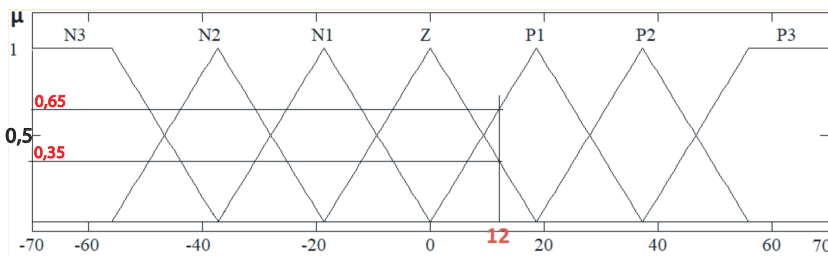
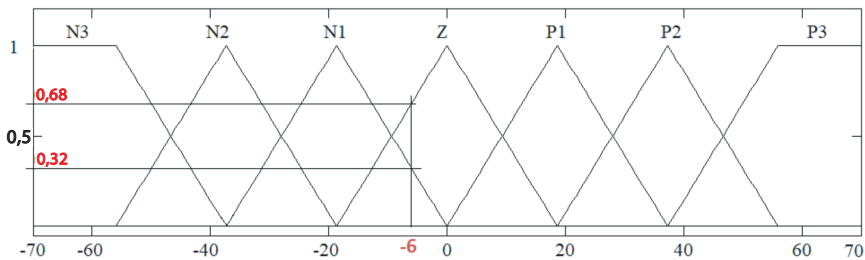


Figura 6

Fusificación para $dE = -6 \text{ (rad/s}^2\text{)}$



2.4.3 Máquina de inferencia

La máquina de inferencia está ligada a las reglas y, particularmente, a las reglas que se activan de acuerdo a la etapa de fusificación. Por eso, es necesario describir antes la metodología que se ha seguido para construir la base de reglas.

2.4.3.1 Base de reglas

La metodología para crear la base de reglas se fundamenta en el conocimiento que se tenga del proceso. La base de reglas es un arreglo cuyas filas y columnas son los conjuntos de las variables E y dE , y en cuyas celdas se escribe uno de los conjuntos de la variable de $SF1$.

El número total de reglas utilizadas en este trabajo es de 49 porque hay 7 conjuntos para cada variable (7×7). En la Figura 7, se muestra la base de reglas y cada regla son sentencias IF-THEN (si-entonces), por ejemplo:

IF E es $N1$ AND dE es $P2$ THEN $SF1$ es $P1$

Donde: AND es un operador de la lógica difusa, los conjuntos $N1$ y $P2$ son los antecedentes de la regla, mientras que el conjunto $P1$ es el consecuente de la regla.

El procedimiento para construir la base de reglas ha sido el de lograr una distribución de los consecuentes en las celdas, de tal manera que, al graficar E , dE y $SF1$ en un gráfico 3D, la superficie del gráfico (llamada *superficie de control*) muestre cambios graduales.

El primer paso del proceso fue completar la diagonal de la base de reglas (celdas sombreadas en la Figura 7). Esta diagonal atraviesa las áreas en la base de reglas, donde $E < 0$ ($N3$, $N2$ y $N1$), $dE > 0$ ($P3$, $P2$ y $P1$), $E > 0$ ($P1$, $P2$ y $P3$) y $dE < 0$ ($N1$, $N2$ y $N3$), que corresponden a las áreas $A1$ y $A3$ de la Figura 2, donde la salida (ω_{sal}) se está corrigiendo

en dirección a la referencia. Además, esta corrección se realiza con una velocidad (dE) proporcional al error (E), como se puede ver en la base de reglas. Por tanto, se ha creído conveniente colocar a lo largo de esta diagonal el conjunto Z .

Luego, a partir de la diagonal, en cada columna verticalmente por encima y debajo de Z , se han colocado los conjuntos difusos contiguos correspondientes, y las celdas que han quedado vacías se han completado con el último conjunto escrito. Esta distribución en la base de reglas ha generado una superficie de control con variaciones suaves de la variable $SF1$, que es lo que se deseaba (véase la Figura 8). La base de reglas resultante tiene un lógico sentido de funcionamiento. Así, por ejemplo, cuando en la base de reglas $E < 0$ y $dE < 0$, significa que se está en el sector A2 de la Figura 2, donde la velocidad del motor está alejándose por encima de la referencia; por tanto, la acción de control es $SF1 < 0$. En cambio, cuando en la base de reglas se tenga $E > 0$ y $dE > 0$, implica estar en el sector A4 de la Figura 2, donde la velocidad del motor está alejándose por debajo de la referencia; por tanto, la acción de control es $SF1 > 0$.

Figura 7

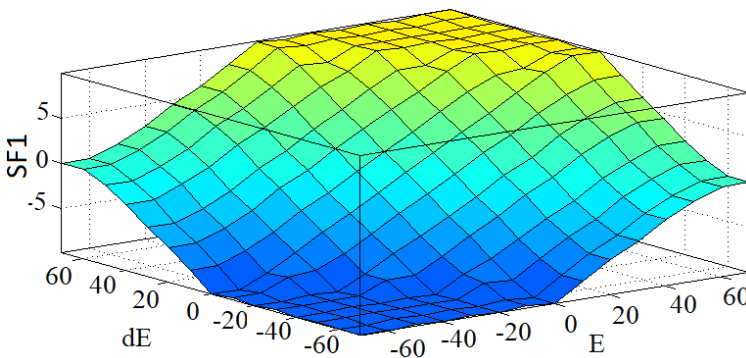
Base de reglas

		E						
		N3	N2	N1	Z	P1	P2	P3
dE	N3	N3	N3	N3	N3	N2	N1	Z
	N2	N3	N3	N3	N2	N1	Z	P1
	N1	N3	N3	N2	N1	Z	P1	P2
	Z	N3	N2	N1	Z	P1	P2	P3
	P1	N2	N1	Z	P1	P2	P3	P3
	P2	N1	Z	P1	P2	P3	P3	P3
	P3	Z	P1	P2	P3	P3	P3	P3

P1, P2 y P3 > 0 N1, N2 y N3 < 0

Figura 8

Gráfico 3D de E, dE y SF1 (superficie de control)



2.4.3.2 Inferencia

Las reglas se han evaluado con el método de mín-máx de Mamdani, que toma el mínimo de los antecedentes para determinar la fuerza de cada regla y toma la regla más fuerte de cada consecuente.

Para seguir con el ejemplo planteado, del total de las 49 reglas posibles que se tiene en la base de reglas y de acuerdo al resultado de la etapa de fusificación, solo poseen un valor numérico significativo (no nulos) aquellas reglas que tengan por antecedentes para la variable E a los conjuntos Z y P1, y para la variable dE, a los conjuntos Z y N1. Por tanto, las reglas que tendrán valor numérico significativo son las cuatro que están sombreadas en la Figura 9 y están escritas en la Tabla 1.

Figura 9

Reglas que solo tienen valor numérico para el ejemplo

		E						
		N3	N2	N1	Z	P1	P2	P3
dE	N3	N3	N3	N3	N3	N2	N1	Z
	N2	N3	N3	N3	N2	N1	Z	P1
	N1	N3	N3	N2	N1	Z	P1	P2
	Z	N3	N2	N1	Z	P1	P2	P3
	P1	N2	N1	Z	P1	P2	P3	P3
	P2	N1	Z	P1	P2	P3	P3	P3
	P3	Z	P1	P2	P3	P3	P3	P3

Tabla 1

Cuatro de las 49 reglas con valor numérico significativo

N.º	Regla
1	IF el error es Z (0,35) AND dE es N1 (0,32), THEN la SF1 es N1
2	IF el error es Z (0,35) AND dE es Z (0,68), THEN la SF1 es Z
3	IF el error es P1 (0,65) AND dE es N1 (0,32), THEN la SF1 es Z
4	IF el error es P1 (0,65) AND dE es Z (0,68), THEN la SF1 es P1

Como los antecedentes están conectados con el operador AND, las reglas asumen como resultado el valor más pequeño de los antecedentes, como se observa en la última columna de la Tabla 1. El resultado total o la salida difusa se encuentra comparando las fuerzas de todas las reglas que especifican el mismo resultado (salida). Así, se puede observar que las reglas 2 y 3 tienen por salida a Z; por tanto, la salida para Z es determinada por la regla de mayor fuerza (es más verdadera). En consecuencia, los conjuntos difusos de salida son estos:

- SF1 es N1 con un grado de verdad de 0,32
- SF1 es Z con un grado de verdad de 0,35
- SF1 es P1 con un grado de verdad de 0,65

2.4.4 Defusificación

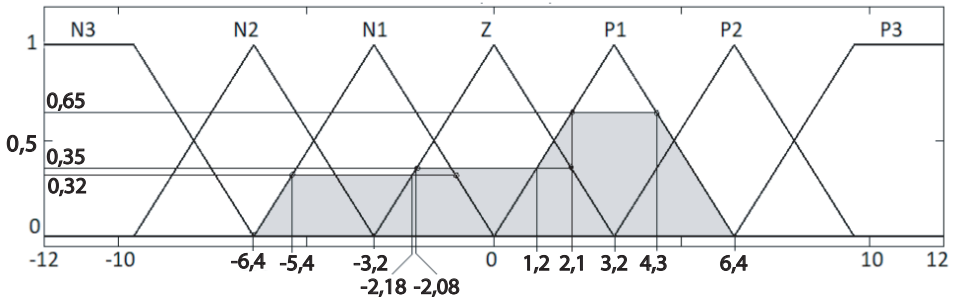
Es el proceso inverso de la fusificación. Existen varios métodos para la defusificación. En este trabajo, se ha optado por el método de centro de masa (centro de gravedad o centroide); el cálculo del centroide se realiza sobre el área determinada por la intersección de los valores de los conjuntos de salida con sus respectivas funciones de pertenencia. En (9) se muestra la fórmula aplicada:

$$SF1 = \frac{\int \mu_c(x) * x dx}{\int \mu_c(x) dx} \quad (9)$$

Continuando con el ejemplo, con los conjuntos difusos de salida del paso anterior, se intercepta a las funciones de pertenencia de salida, de manera que queda la figura geométrica que se observa en la Figura 10.

Figura 10

Defusificación para el caso del ejemplo



Luego, aplicando la fórmula (9):

$$SF1 = \frac{\int_{-5,4}^{-2,18} \left(\frac{x+6,4}{3,2}\right) x dx + \int_{-5,4}^{-2,18} 0,32x + \int_{-2,18}^{-2,08} \left(\frac{x+3,2}{3,2}\right) x dx + \int_{-2,08}^{1,2} 0,35x \dots + \int_{4,3}^{6,4} \left(\frac{6,4-x}{3,2}\right) x dx}{\int_{-5,4}^{-2,18} \left(\frac{x+6,4}{3,2}\right) dx + \int_{-5,4}^{-2,18} 0,32 + \int_{-2,18}^{-2,08} \left(\frac{x+3,2}{3,2}\right) dx + \int_{-2,08}^{1,2} 0,35 \dots + \int_{4,3}^{6,4} \left(\frac{6,4-x}{3,2}\right) dx}$$

SF1 = 0,68

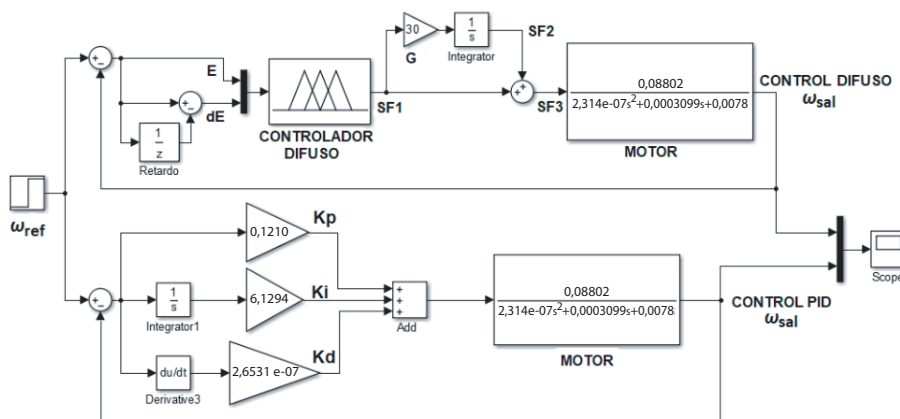
Es decir, cuando el error del sistema (E) es 12 y el cambio del error (dE) es de -6, entonces, SF1 genera una salida de 0,68 (es decir, en términos de lógica difusa, un valor exacto de 0,68). Sin embargo, según se vio en la Figura 1, SF1 por sí solo no alimenta al motor, sino que también hay una contribución de la integral de SF1; esta integración se produce continuamente, ajustando el valor resultante de SF3 y, por tanto, a la velocidad del motor (ω_{sal}); de esta manera, se mejoran los transitorios y el error de estado estacionario. El nuevo valor de la velocidad del motor (ω_{sal}) se compara otra vez con la velocidad de referencia (ω_{ref}), lo que genera un nuevo error (E) y un cambio del error (dE); todo el proceso descrito se repite continuamente.

El editor del Sistema de Inferencia Difuso (FIS) de MATLAB se utilizó para definir las variables de entrada, de salida y de las funciones de pertenencia, así como para editar la base de reglas y evaluar su respuesta de acuerdo a los valores de entrada.

El simulador Simulink de MATLAB se usó para obtener la respuesta del motor con el controlador difuso y también con el controlador PID; los parámetros del PID son los señalados en (6), (7) y (8). En la Figura 11, se muestran los bloques del simulador. La velocidad de referencia (ω_{ref}) fue establecida en 1 rad/s. Ambas respuestas, la del controlador difuso y la del controlador PID, se registraron mediante una matriz en el bloque "Scope". La matriz se exportó al ambiente principal de MATLAB y luego se la graficó mediante los comandos de "Figure" y "Plot". Los transitorios de las señales fueron obtenidos con el comando "Step info". En la Tabla 2, se aprecian los transitorios obtenidos.

Figura 11

Vista del sistema de control difuso y control PID



3. RESULTADOS

La Tabla 2 presenta una comparación entre el controlador difuso y el controlador PID sintonizado con las herramientas de MATLAB.

Tabla 2

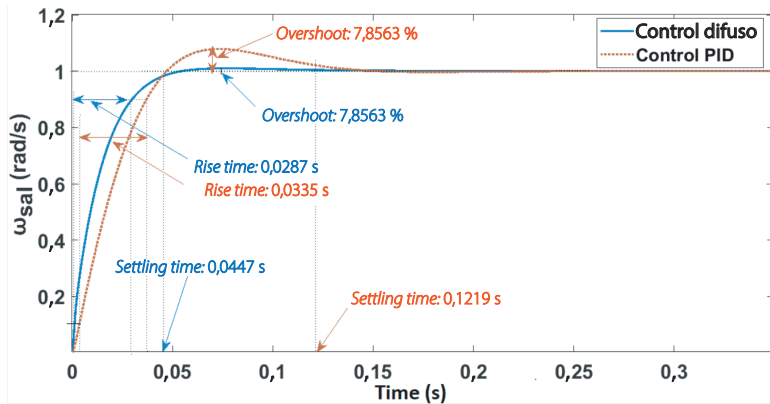
Comparación de resultados

Controlador	Rise time (s)	Setting time (s)	Overshoot (%)
Difuso	0,0287	0,0447	0,9539
PID	0,0335	0,1219	7,8563

En la Figura 12, se observan los gráficos obtenidos de la velocidad del motor para el controlador difuso y para el controlador PID, señalando los transitorios para ambos casos.

Figura 12

Respuesta del motor para el controlador difuso y el controlador PID



4. DISCUSIÓN

Los resultados obtenidos revelan diferencias significativas entre el controlador difuso y el controlador PID:

- En términos de *rise time* (tiempo de respuesta), el controlador difuso demuestra un tiempo más corto en alcanzar la referencia en comparación con el controlador PID. Esto indica que el controlador difuso logra una respuesta más rápida y eficiente.
- El *settling time* (tiempo de asentamiento) para el controlador difuso es casi tres veces menor que para el controlador PID. Esto significa que el controlador difuso logra estabilizar la velocidad del motor de manera mucho más rápida que el controlador PID, lo que implica un mejor rendimiento en cuanto a la estabilidad del sistema.
- En cuanto al *overshoot* (sobreimpulso), el controlador difuso muestra un valor ocho veces menor que el controlador PID. Esto indica que el controlador difuso genera una respuesta más suave y con menor oscilación en la velocidad del motor en comparación con el controlador PID.

Por tanto, los resultados respaldan la eficiencia y superioridad del controlador difuso en la regulación de la velocidad del motor en comparación con el controlador PID utilizado en este estudio. El controlador difuso logra un tiempo de respuesta más rápido, un tiempo de asentamiento más corto y un menor sobreimpulso, lo que demuestra su capacidad para controlar de manera más eficaz la velocidad del motor.

5. CONCLUSIONES

Ha quedado plenamente demostrada la eficiencia del controlador difuso al generar transitorios menores en el motor en comparación con el controlador PID, el cual fue sintonizado utilizando las herramientas de MATLAB.

La adición de un integrador a la salida y la inclusión de un bloque de ganancia en el controlador difuso, ambos debidamente fundamentados, han demostrado ser elementos externos indispensables a partir de los resultados obtenidos.

Las funciones de pertenencia seleccionadas, la base de reglas diseñada y todo el diseño del controlador difuso ha quedado también debidamente justificado por los resultados obtenidos.

Se ha logrado una descripción detallada de la metodología utilizada para construir la base de reglas, abarcando todas las etapas del proceso. Este enfoque exhaustivo ha contribuido de manera significativa a la comprensión y claridad del procedimiento.

REFERENCIAS

- Almatheel, Y. A., & Abdelrahman, A. (2017). Speed control of DC motor using Fuzzy Logic Controller. En *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)* (pp. 1-8). IEEE.
- Ekinci, S., Hekimoğlu, B., & Izci, D. (2021). Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor. *Engineering Science and Technology, an International Journal*, *24*(2), 331-342.
- Hekimoğlu, B. (2019). Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm. *IEEE Access*, *7*, 38100-38114.
- Jantzen, J. (1998). *Tuning of fuzzy PID controllers* [Reporte técnico n.º 98-H 871]. Technical University of Denmark, Department of Automation.
- Johnson, M. A., & Moradi, M. H. (2005). *PID control*. Springer-Verlag.
- Li, H. X., & Gatland, H. B. (1995). A new methodology for designing a fuzzy logic controller. *IEEE Transactions on Systems, Man, and Cybernetics*, *25*(3), 505-512. <https://doi.org/10.1109/21.364863>
- Rajagiri, A. K., Rani, S., & Nawaz, S. S. (2019). Speed control of dc motor using fuzzy logic controller by PCI 6221 with MATLAB. En *E3S Web of Conferences* (vol. 87, art. 01004). EDP Sciences.
- Sharma, K., & Palwalia, D. K. (2017). A modified PID control with adaptive fuzzy controller applied to DC motor. En *2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC)* (pp. 1-6). IEEE.
- Somwanshi, D., Bundele, M., Kumar, G., & Parashar, G. (2019). Comparison of fuzzy-PID and PID controller for speed control of DC motor using LabVIEW. *Procedia Computer Science*, *152*, 252-260.
- Tang, W. J., & Cao, S. Y. (2018). A fast realization method of fuzzy PID control for DC motor. En *2018 37th Chinese Control Conference (CCC)* (pp. 5131-5135). IEEE.
- Zhu, W., Ma, X., Sun, C., & Niu, X. (2019). Design of fuzzy-PID speed controller for brushless DC motor. En *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)* (pp. 1038-1042). IEEE.