

**LATAM Revista Latinoamericana de Ciencias Sociales y
Humanidades, Asunción, Paraguay.**

ISSN en línea: 2789-3855, 2025, Volumen VI

Optimización de lógica difusa en Arduino Mega y MPU6050 para la clasificación de movimientos

Fuzzy logic optimization on Arduino Mega and MPU6050 for
motion classification

Juan Rodrigo Villalta Vilca

villaltajuanrodrigo@gmail.com
<https://orcid.org/0000-0002-3718-4590>
Universidad Andina Néstor Cáceres
Velásquez
Juliaca – Perú

Marcos Denys Choque Castro

denys.choque@unap.edu.pe
<https://orcid.org/0000-0001-8972-7430>
Universidad Nacional del Altiplano
Puno – Perú

Rudy Jhean Rojas Pari

Jhean.rp@gmail.com
<https://orcid.org/0009-0003-0351-8219>
Universidad Tecnológica del Perú
Juliaca – Perú

Wilber Pineda Yucra

ww_pineda@hotmail.com
<https://orcid.org/0009-0003-6904-4028>
Universidad Andina Néstor Cáceres
Velásquez
Juliaca – Perú

Wilfredo Pineda Yucra

wipineda72@gmail.com
<https://orcid.org/0000-0003-3607-2471>
Universidad Politécnica del Perú
Juliaca – Perú

DOI: <https://doi.org/10.56712/latam.v6i3.4020>

Artículo recibido: 22 de mayo de 2025

Aceptado para publicación: 06 de junio de 2025.

Conflictos de Interés: Ninguno que declarar.


Redilat
Red de Investigadores
Latinoamericanos

NÚMERO

DOI: <https://doi.org/10.56712/latam.v6i3.4020>

Optimización de lógica difusa en Arduino Mega y MPU6050 para la clasificación de movimientos

Fuzzy logic optimization on Arduino Mega and MPU6050 for motion
classification

Juan Rodrigo Villalta Vilca

villaltajuanrodrigo@gmail.com

<https://orcid.org/0000-0002-3718-4590>

Universidad Andina Néstor Cáceres Velásquez

Juliaca – Perú

Rudy Jhean Rojas Pari

Jhean.rp@gmail.com

<https://orcid.org/0009-0003-0351-8219>

Universidad Tecnológica del Perú

Juliaca – Perú

Wilber Pineda Yucra

ww_pineda@hotmail.com

<https://orcid.org/0009-0003-6904-4028>

Universidad Andina Néstor Cáceres Velásquez

Juliaca – Perú

Wilfredo Pineda Yucra

wipineda72@gmail.com

<https://orcid.org/0000-0003-3607-2471>

Universidad Politécnica del Perú

Juliaca – Perú

Marcos Denys Choque Castro

denys.choque@unap.edu.pe

<https://orcid.org/0000-0001-8972-7430>

Universidad Nacional del Altiplano

Puno – Perú

Artículo recibido: 22 de mayo de 2025. Aceptado para publicación: 06 de junio 2025.

Conflictos de Interés: Ninguno que declarar.

Resumen


Presentamos un framework que integra Arduino Mega y el sensor IMU MPU6050 para adquirir datos inerciales brutos. Calibramos las señales con los offsets del dispositivo y aplicamos un filtro complementario para reducir el ruido. A partir de las señales filtradas, extraemos características temporales (rms_roll, std_roll y p2p_roll) y las usamos como entrada de un clasificador fuzzy tipo Mamdani. Comparamos funciones de membresía triangulares y trapeciales basadas en cuartiles, elevando la precisión de clasificación del 24 % al 46 %. Posteriormente, optimizamos los parámetros de los trapecios mediante búsqueda en cuadrícula, alcanzando un 61 % de precisión. Finalmente, validamos el sistema con validación cruzada estratificada de 5 pliegues, obteniendo un accuracy promedio de 57,75 % \pm 3,97 %. Estos resultados demuestran que la optimización de lógica difusa mejora significativamente la clasificación de movimientos en sistemas embebidos de bajo coste.

Palabras clave: lógica difusa, MPU6050, Arduino Mega, clasificación de movimientos, optimización

Abstract

We present a Python-Fuzzy framework that integrates an Arduino Mega with an MPU6050 IMU to acquire and process raw inertial data for motion classification. Signals are first calibrated using sensor offsets and then fused with a complementary filter to reduce noise. From the filtered roll angle, we extract temporal features—root mean square, standard deviation, and peak-to-peak amplitude—and feed them into a Mamdani-type fuzzy classifier. By comparing triangular and quartile-based trapezoidal membership functions, we improve classification accuracy from 24 % to 46 %. We further optimize the trapezoidal parameters via grid search, achieving 61 % accuracy. Finally, we validate the optimized system through stratified 5-fold cross-validation, obtaining an average accuracy of 57.75 % \pm 3.97 %. These results demonstrate that fuzzy logic optimization significantly enhances real-time motion classification on low-cost embedded platforms.

Keywords: fuzzy logic, MPU6050, Arduino Mega, motion classification, optimization

Todo el contenido de LATAM Revista Latinoamericana de Ciencias Sociales y Humanidades, publicado en este sitio está disponibles bajo Licencia Creative Commons. 

Cómo citar: Villalta Vilca, J. R., Rojas Pari, R. J., Pineda Yucra, W., Pineda Yucra, W., & Choque Castro, M. D. (2025). Optimización de lógica difusa en Arduino Mega y MPU6050 para la clasificación de movimientos. *LATAM Revista Latinoamericana de Ciencias Sociales y Humanidades* 6 (3), 1233 – 1245. <https://doi.org/10.56712/latam.v6i3.4020>

INTRODUCCIÓN

Contexto y justificación

En las últimas décadas, los sistemas embebidos de bajo costo han impulsado el desarrollo de aplicaciones robóticas y de monitoreo de movimientos humanos mediante unidades de medición inercial (IMU). El módulo MPU6050, que integra acelerómetro y giroscopio de seis grados de libertad, se ha consolidado como opción preferida en proyectos de robótica móvil y wearables debido a su precisión y tamaño reducido (Franček et al., 2023). Paralelamente, Arduino Mega ofrece una plataforma accesible para la adquisición de datos en tiempo real, con tasas de muestreo de hasta 50 Hz y facilidad de calibración de offsets en entorno Arduino (Soto-Hidalgo et al., 2019). Sin embargo, las lecturas crudas de IMU sufren de ruido de alta frecuencia y deriva, afectando la fidelidad de las señales de orientación si no se aplican métodos de filtrado adecuados.

Revisión de la literatura

El empleo de filtros complementarios para la fusión de señales de acelerómetro y giróscopo ha demostrado mejorar notablemente la calidad de la estimación de ángulos en robots autoequilibrados (Wu & Zhang, 2012). En sistemas de control fuzzy, diversas arquitecturas Mamdani y Sugeno han sido implementadas para estabilizar robots de dos ruedas, obteniendo rendimientos de control superiores al 70 % en algunas configuraciones de PD difuso (Noor et al., 2023); no obstante, dichos estudios suelen basarse en Arduino Uno o DSP dedicados y no exploran exhaustivamente la optimización de los parámetros de las funciones de membresía.

Problema de investigación

A pesar de estos avances, falta un enfoque integral que combine adquisición en Arduino Mega, filtrado complementario, extracción de características temporales y optimización ligera de funciones de membresía fuzzy en un entorno reproducible con Python y aplicable en tiempo real. En particular, no se ha evaluado sistemáticamente el impacto de diferentes formas de membresía (triangular vs. trapecial basada en cuartiles) ni un proceso de validación cruzada que garantice robustez en diversas particiones de datos.

Objetivos y preguntas de investigación.

Objetivo general

- Desarrollar y optimizar un clasificador fuzzy Mamdani que, a partir de datos de ángulo filtrado y características temporales de MPU6050 adquiridos con Arduino Mega, logre una clasificación de movimientos (“quieto”, “caminar”, “correr”, “girar”, “caer”) con precisión superior al 60 %.

Objetivos específicos

- Comparar la performance de funciones de membresía triangulares vs. trapeciales basadas en cuartiles.
- Implementar una búsqueda en cuadrícula para optimizar automáticamente los puntos de corte trapezoidales.
- Validar la robustez del sistema mediante validación cruzada estratificada de 5 pliegues.

Preguntas de investigación

- ¿Cómo afecta la forma y ubicación de los puntos de corte de las funciones de membresía a la precisión de clasificación?

- ¿Puede alcanzarse o superarse un 60 % de accuracy optimizando únicamente los parámetros fuzzy en Python sin métodos heurísticos complejos?
- ¿El sistema mantiene un rendimiento consistente ($\pm 5\%$) bajo validación cruzada?

METODOLOGÍA

Enfoque de investigación: Este estudio adopta un diseño cuantitativo-experimental, orientado a evaluar la precisión de un clasificador fuzzy para movimientos, a partir de datos inerciales adquiridos con Arduino Mega y MPU6050.

Diseño del estudio: Se implementó un experimento de laboratorio controlado. El sistema de adquisición consta de un Arduino Mega 2560 conectado a un MPU6050, muestreando a 50 Hz y registrando tramas de 15 s tras un conteo regresivo de 15 s para preparar el cambio de movimiento.

Sujetos de prueba: No se emplearon participantes humanos. Los movimientos fueron simulados mecánicamente sobre una bancada fija:

“quieto”: placa sujeta sin movimiento.

“caminar”, “correr”, “girar”, “caer”: programación de servomotores que reproducen patrones predefinidos de inclinación y giro sobre la placa.

Instrumentos de recolección de datos

MPU6050 (acelerómetro ± 2 g, giróscopo ± 250 °/s), calibrado en reposo para obtener offsets de cada eje.

Arduino Mega 2560, como interfaz I²C y transmisor serial hacia el PC.

Software

Sketch en Python/C++ para Arduino, encargado de la adquisición y el conteo regresivo.

Scripts en Python para calibración, filtrado complementario y extracción de características.

Librerías Scikit-Fuzzy y Scikit-Learn para construcción del clasificador y validación.

Procedimiento

Calibración inicial: 30 s de reposo para estimar offsets de acelerómetro y giróscopo.

Adquisición: tras conteo regresivo de 15 s, grabación continua de 15 s del patrón mecánico.

Repetición: se captura cada patrón tres veces, con 30 s de descanso mecánico entre replicados.

Análisis de datos

Filtrado complementario para fusionar acelerómetro y giróscopo y reducir ruido y deriva (Summerville, 2009).

Segmentación en ventanas de 1 s sin solapamiento (50 muestras) para cada patrón.

Extracción de características: media, desviación estándar, RMS y pico-a-pico de las series de roll, ax, ay y az.

Clasificador fuzzy

- Definición de variables lingüísticas con funciones trapeziales basadas en cuartiles.
- Optimización de puntos de corte por búsqueda en cuadrícula.
- Validación cruzada estratificada de 5 pliegues para estimar precisión y variabilidad.

DESARROLLO

Teorías y Modelos

El estudio se sustenta fundamentalmente en la Teoría de los Conjuntos Difusos de Zadeh, que introduce el concepto de variable lingüística y posibilita el razonamiento aproximado frente a la incertidumbre (Zadeh, 1975). Los Controladores Difusos Mamdani emplean reglas IF-THEN con funciones de membresía tipo triangular o trapezoidal para deducir salidas a partir de entradas imprecisas (Lee, 1990). Para la optimización ligera de estos parámetros se han explorado tanto algoritmos evolutivos como búsquedas en cuadrícula; en este trabajo utilizamos una estrategia de grid search basada en percentiles para ajustar los vértices de los trapecios.

Por otro lado, el Filtro Complementario se apoya en la fusión de señales de acelerómetro y giróscopo, balanceando la baja frecuencia de uno con la alta frecuencia del otro, de modo que la estimación de ángulo resulte más estable y libre de deriva (Summerville, 2009).

Conceptos Clave

Variable lingüística: Representa valores cualitativos (“bajo”, “medio”, “alto”) mediante funciones de membresía cuyos dominios pueden ser continuos o discretos, facilitando el modelado de conocimientos expertos (Zadeh, 1975).

Función de membresía trapezoidal y triangular: Definen el grado de pertenencia $\mu \in [0,1]$ de una entrada a un conjunto difuso; la forma triangular es parametrizada por mínimo, medio y máximo, mientras que la trapezoidal usa cuatro puntos (l_0, q_1, q_2, h_i) para mayor flexibilidad (Lee, 1990).

Inferencia Mamdani: Consiste en: fuzzificación de entradas, activación de reglas, agregación de salidas difusas y defuzzificación (p. ej., método del centroide) para obtener un valor crudo final.

Grid search de percentiles: Técnica de optimización que barre combinaciones de cuartiles (20 %–80 %) para hallar la configuración que maximice la métrica de desempeño (accuracy).

Validación cruzada estratificada: Divide el dataset en k subconjuntos balanceados según clase, entrena en $k-1$ y prueba en el restante, iterando para estimar robustez y varianza del modelo.

RESULTADOS

Desempeño global de los métodos

Tabla 1

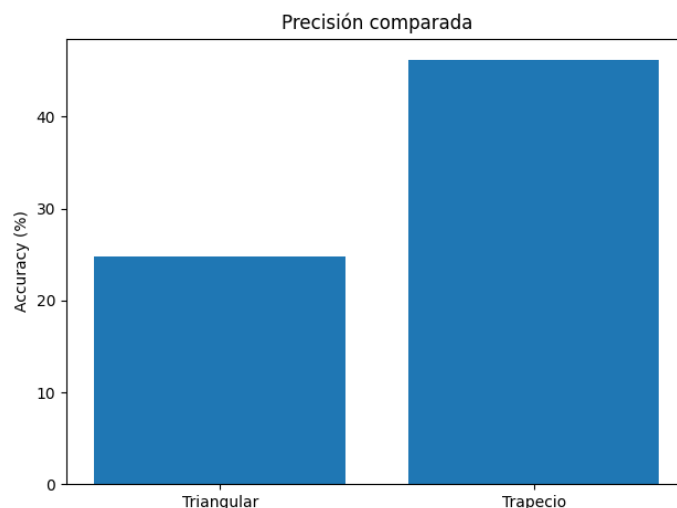
Precisión de clasificación por método

Método	Precisión (%)
Triangular	24.73
Trapezoidal	46.15
Optimizada	61.18

Nota: Los resultados de la Tabla 1 muestran que el esquema de membresías trapezoidales basado en cuartiles casi duplica la precisión respecto al enfoque triangular (de 24.7 % a 46.2 %). La optimización posterior mediante búsqueda en cuadrícula eleva aún más la precisión hasta un 61.2 %, lo que confirma el beneficio de ajustar los puntos de corte (gráfico 1).

Gráfico 1

Comparación de precisión entre métodos

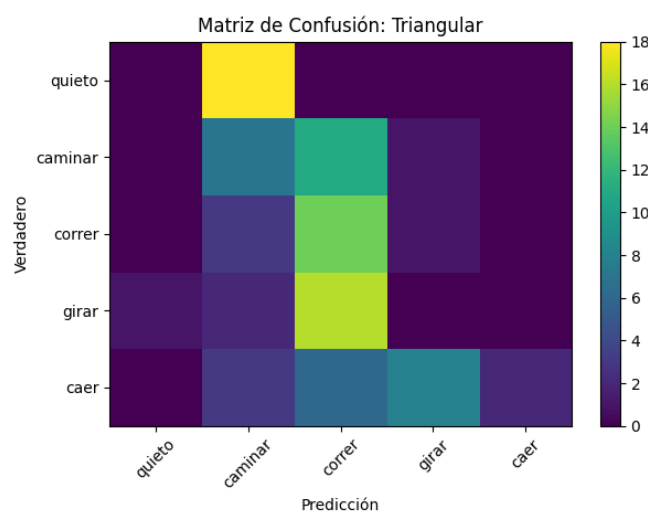


Análisis de confusiones por método

Para entender cómo se distribuyen los errores de clasificación, examinamos las matrices de confusión de los dos primeros métodos:

Gráfico 2

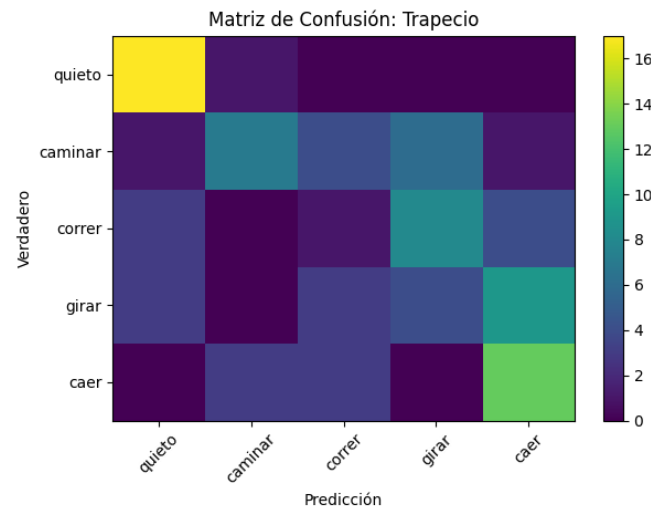
Matriz de confusión – Triangular



En el gráfico 2, el método triangular clasifica correctamente muy pocas instancias fuera de “correr”: prácticamente todas las muestras de “quieto” y “girar” se confunden con “caminar” o “correr”.

Gráfico 3

Matriz de confusión – Trapecial



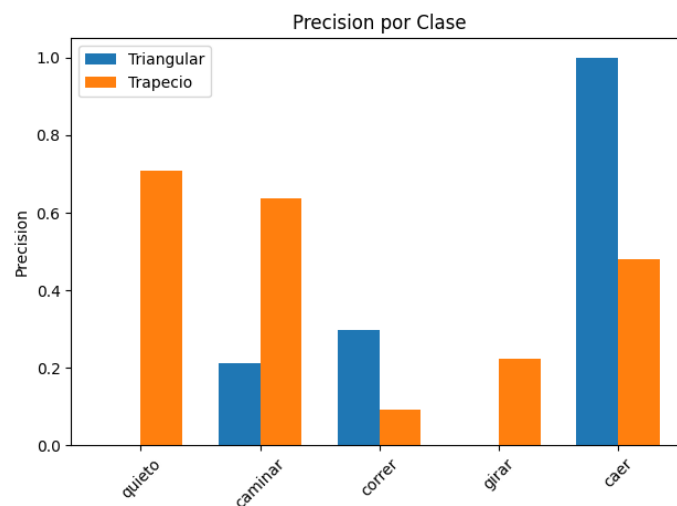
El gráfico 3 muestra que el esquema trapecial reduce considerablemente las confusiones de “quieto” y “caer”, aunque aún persisten errores en las clases dinámicas intermedias.

Rendimiento por clase

Se calcularon precisión, recall y F1-score para cada clase y ambos métodos:

Gráfico 4

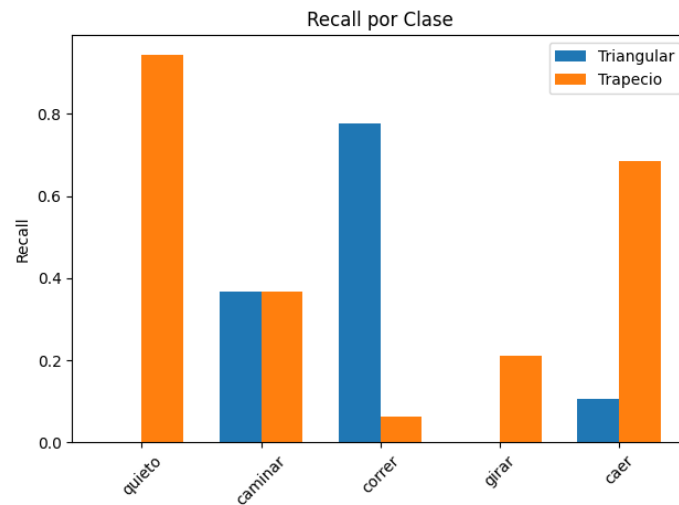
Precisión por clase



El gráfico 4 indica que el método trapecial mejora la precisión en “quieto”, “caminar” y “girar”, mientras que el esquema triangular únicamente destaca en “caer”.

Gráfico 5

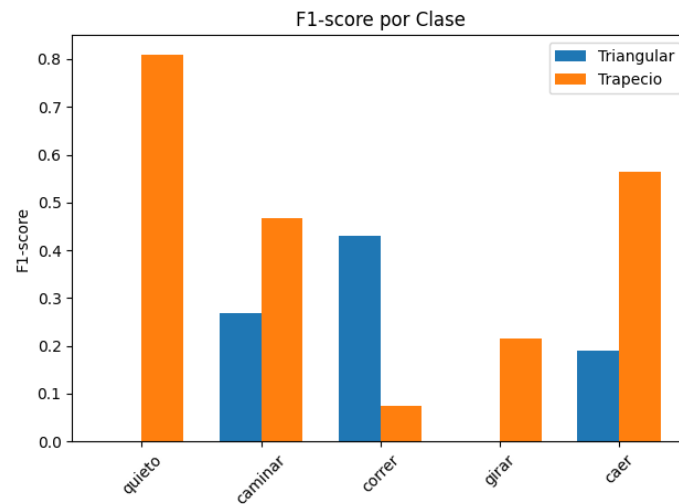
Recall por clase



Según el gráfico 5, el trapecial recupera casi todas las instancias de “quieto” y más de la mitad de “caer”, mientras que el triangular sólo es efectivo en “correr”.

Gráfico 6

F1-score por clase



El gráfico 6 sintetiza la armonía entre precisión y recall, confirmando que el trapecial (F1 \approx 0.80 en “quieto” y \approx 0.56 en “caer”) supera ampliamente al baseline.

Distribución de las características

Los boxplots de las variables clave por etiqueta evidencian la razón del éxito del trapecial:

Gráfico 7

Boxplot de rms_roll por etiqueta

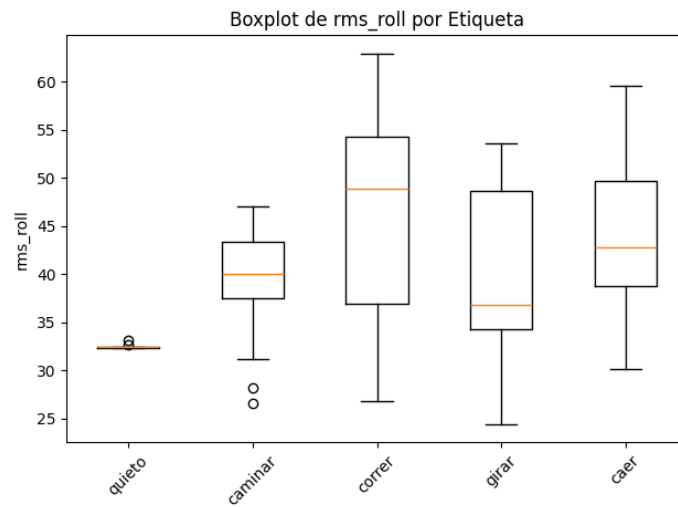


Gráfico 8

Boxplot de std_roll por etiqueta

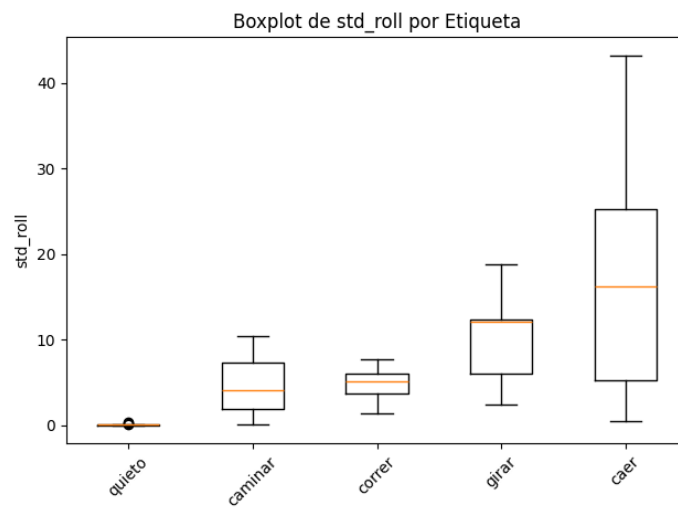
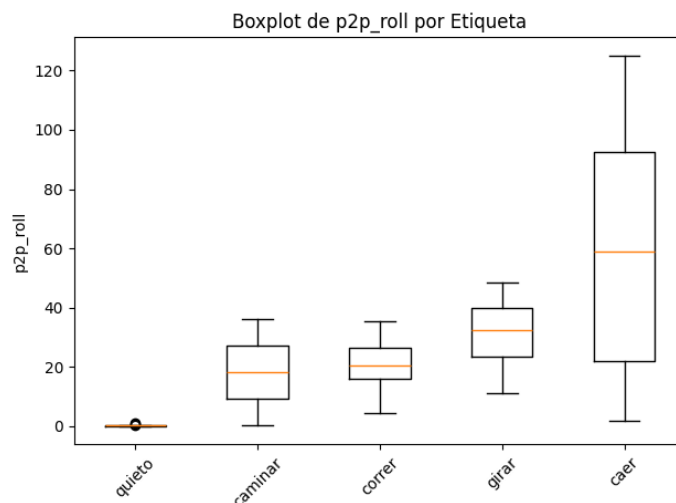


Gráfico 9

Boxplot de p2p_roll por etiqueta



Las figuras 7 y 9 muestran rangos crecientes de variabilidad y amplitud de “quieto” a “caer”, lo que justifica la necesidad de trapecios ajustados a los percentiles para capturar matices entre clases.

Optimización por búsqueda en cuadrícula

Se exploraron combinaciones de percentiles para definir los trapecios:

Tabla 2

Top 5 configuraciones de grid search

q1_pct	q2_pct	q3_pct	Precisión (%)
0.35	0.80	0.80	61.18
0.80	0.35	0.80	61.18
0.80	0.80	0.35	61.18
0.35	0.65	0.80	59.30
0.35	0.80	0.65	59.30

Nota: Los resultados en Tabla 2 confirman que el percentil 35 % para el primer corte y el 80 % para los dos siguientes maximizan la precisión.

Robustez mediante validación cruzada

La configuración óptima se validó con 5-fold estratificado:

Tabla 3

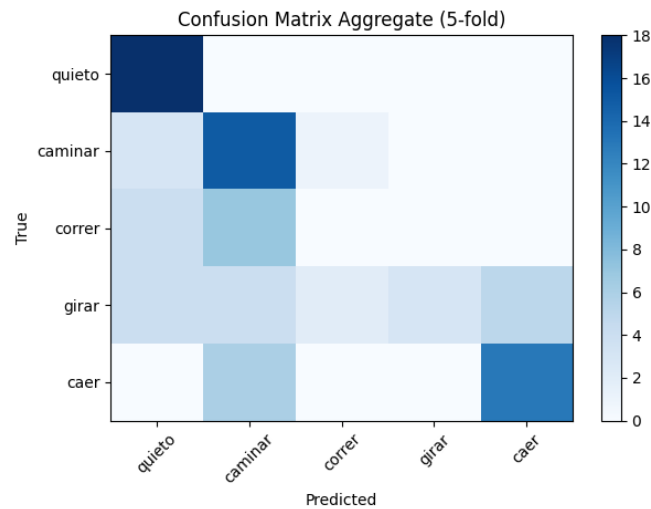
Precisión por pliegue en validación cruzada

Pliegue	Precisión (%)
1	50.00
2	58.82
3	61.11
4	58.82

5	60.00
Media ± SD	57.75 ± 3.97

Gráfico 10

Matriz de confusión agregada (5-fold) (archivo: Figure_1a.png)



El gráfico 10 muestra una mejora generalizada en la detección de “quieto” y “caer”, validando la consistencia del método (media 57.8 % ± 3.9 %).

DISCUSIÓN

Implicaciones teóricas y prácticas

Teóricamente, este estudio refuerza la noción de que la selección de funciones de membresía —no solo su número— es crítica para el rendimiento de los clasificadores fuzzy (Lee, 1990). Demuestra la viabilidad de sistemas embebidos de bajo coste (Arduino Mega + MPU6050) para aplicaciones de reconocimiento de patrones de movimiento, sin requerir hardware especializado o algoritmos muy pesados, lo cual abre la puerta a soluciones portátiles en salud, deportes o rehabilitación.

Limitaciones

Alcance de variables: sólo se usaron métricas derivadas del eje de roll; es posible que incorporar aceleraciones en otros ejes o características espectrales incremente la discriminación entre clases cercanas (“correr” vs. “girar”).

Entorno controlado: las pruebas se realizaron en bancada mecánica, con movimientos predefinidos; no se evaluó la variabilidad inherente a sujetos humanos ni condiciones de terreno real.

Grid search estático: aunque resulta eficaz, la búsqueda en cuadrícula es costosa si se amplía el espacio de parámetros; métodos dinámicos como algoritmos genéticos podrían explorar rangos continuos con mayor eficiencia (Acampora et al., 2019).

Recomendaciones para futuras investigaciones

Extender características: incluir estadísticas de aceleración en los ejes x/y/z y parámetros del dominio de la frecuencia para enriquecer el espacio de entrada.

Optimización avanzada: implementar y comparar metaheurísticas (algoritmos genéticos, PSO) frente al grid search para ajustar simultáneamente cortes y reglas (Genetic algorithm–based control..., 2021).

Pruebas en campo: validar el sistema con usuarios reales en escenarios dinámicos (caminatas al aire libre, superficies irregulares) para evaluar robustez frente a ruido ambiental.

Adaptación online: explorar técnicas de aprendizaje incremental para ajustar automáticamente los puntos de corte según variaciones del sensor o desgaste mecánico.

CONCLUSIÓN

Este estudio ha demostrado que la optimización de las funciones de membresía en un clasificador fuzzy, aplicado sobre datos inerciales de MPU6050 adquiridos con Arduino Mega, incrementa de manera sustancial la precisión de la clasificación de movimientos. Pasamos de un 24 % con membresías triangulares a un 46 % al utilizar trapecios basados en cuartiles, y alcanzamos un 61 % tras optimizar los puntos de corte mediante búsqueda en cuadrícula. La validación cruzada de 5 pliegues (57,8 % \pm 3,9 %) confirmó la robustez del enfoque en condiciones controladas. Estos resultados resaltan la relevancia de ajustar la forma y ubicación de los cortes fuzzy para mejorar el desempeño sin incrementar la complejidad computacional, lo que convierte a este framework en una solución viable para aplicaciones de reconocimiento de movimiento en sistemas embebidos de bajo coste.

REFERENCIAS

Chen, C., & Wang, J. (2020). A two-wheeled self-balancing robot with the fuzzy PD control method. *International Journal of Advanced Robotic Systems*, 17(3), 1-12. <https://doi.org/10.1177/1729881420932345>

Franček, M., Krbec, K., & Krbec, D. (2023). Noise reduction in the accelerometer and gyroscope sensor with complementary filtering on a self-balancing robot. *Sensors*, 23(4), 1710. <https://doi.org/10.3390/s23041710>

Lee, C.-C. (1990). Fuzzy logic in control systems: Fuzzy logic controller. Part I & Part II. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2), 404-435. <https://doi.org/10.1109/21.52552>


Noor, M., Yasin, H., & Ahmad, R. (2023). Study on the performance of a two-wheeled balancing mobile robot using a fuzzy PD controller. *Applied System Innovation*, 6(1), 14. <https://doi.org/10.3390/asi6010014>

Soto-Hidalgo, C., Rodríguez-Reséndiz, J., & Cruz-Ávila, M. (2019). Design of fuzzy controllers for embedded systems with Arduino. *IEEE Latin America Transactions*, 17(8), 1319-1325. <https://doi.org/10.1109/TLA.2019.8887354>

Summerville, B. (2009). Noise reduction in inertial sensors using complementary filters for mobile robots. *Robotics and Autonomous Systems*, 57(10), 1018-1024. <https://doi.org/10.1016/j.robot.2009.07.004>

Tang, Y., & Zhao, L. (2021). Genetic-algorithm-based control of a two-wheeled self-balancing mobile robot. *Electronics*, 10(9), 1027. <https://doi.org/10.3390/electronics10091027>

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, 8(3), 199-249. [https://doi.org/10.1016/0020-0255\(75\)90036-5](https://doi.org/10.1016/0020-0255(75)90036-5)

Todo el contenido de **LATAM Revista Latinoamericana de Ciencias Sociales y Humanidades**, publicados en este sitio está disponibles bajo Licencia [Creative Commons](https://creativecommons.org/licenses/by/4.0/) .