Revista Causalidad

Artículo de Investigación https://doi.org/10.70929/caui3.v2i1.0015

Implementación de un enfoque basado en aprendizaje por refuerzo para el desplazamiento autónomo y eficiente de robots móviles de ruedas

Implementation of a reinforcement learning-based approach for autonomous and efficient locomotion of wheeled mobile robots

Recibido: 26, enero 2026

Aceptado: 05, febrero 2026

Publicado online: 12, febrero 2026

Como Citar: Mieles-Cárdenas et al., "Implementación de un enfoque basado en aprendizaje por refuerzo para el desplazamiento autónomo y eficiente de robots móviles de ruedas", *Revista Digital Científica Causalidad (caui3)*, vol. 2, no. 1, pp. 1–9, doi: 10.70929/caui3.v2i1.0015.

ISSN: 3073-1518

Carlos Mieles-Cárdenas[1], Martín Zavala-Angamarca[2]

Resumen:

Este estudio tiene como objetivo optimizar la navegación autónoma de un robot móvil de ruedas "Turtlebot" en un entorno de laberinto 2D. El objetivo principal se enfoca en la minimización tanto el número de pasos como el tiempo requerido para alcanzar un objetivo, para lo que se han explotado las bondades de un algoritmo de Aprendizaje por Refuerzo. En específico, se ha seleccionado el algoritmo de aprendizaje Optimización de Políticas Proximales (PPO). Así, el robot demostró una curva de aprendizaje exponencial, alcanzando un alto nivel de efectividad en un período de tiempo reducido. Se ha observado una mejora continua y significativa en la calidad de las trayectorias generadas, así como una reducción progresiva en el número de movimientos necesarios para completar la tarea a lo largo de las sesiones de entrenamiento. Este proceso de entrenamiento ha sido organizado en sesiones consecutivas, compuestas por diez mil movimientos, con la capacidad de guardar el modelo aprendido para continuar su refinamiento en iteraciones posteriores. Los resultados validan que el algoritmo PPO, constituye una metodología altamente prometedora para el control de movimiento autónomo en robots.

Palabras Clave: Aprendizaje por refuerzo, PPO, turtlebot, simulación, robot de ruedas.

Abstract:

This study aims to optimize the autonomous navigation of a Turtlebot wheeled mobile robot in a 2D maze environment. The main objective focuses on minimizing both the number of steps and the time required to reach a target, for which the benefits of a reinforcement learning algorithm have been exploited. Specifically, the Proximal Policy Optimization (PPO) learning algorithm has been selected. Thus, the robot demonstrated an exponential learning curve, achieving a high level of effectiveness in a short period of time. A continuous and significant improvement in the quality of the trajectories generated has been observed, as well as a progressive reduction in the number of movements required to complete the task throughout the training sessions. This training process has been organized into consecutive sessions, consisting of ten thousand movements, with the ability to save the learned model for further refinement in subsequent iterations. The results validate that the PPO algorithm is a highly promising methodology for autonomous motion control in robots.

Keywords: Reinforcement learning, PPO, turtlebot, simulation, wheeled robot.

[1]Facultad de Ingeniería Industrial, Universidad de Guayaquil, Guayaquil 090514, Ecuador.

[2]Carrera de Electrónica en Automatización y Telecomunicaciones, Instituto Superior Tecnológico Carlos Cisneros.

Autor de correspondencia: cmieles196@gmail.com

1 Introducción

La navegación autónoma en entornos no estructurados representa una tarea compleja, además, un área de investigación activa en robótica móvil [1], [2], [3], [4], [5]. Es este sentido, es importante destacar que los robots móviles no se limitan solo a las industrias manufactureras; ahora se utilizan en la agricultura, el espacio, el ejército, la medicina, la educación, el rescate y muchos más [6]. A diferencia de las zonas urbanas con carriles, señales de tráfico y mapas, el entorno que rodea al robot es desconocido y no está estructurado. Dicho entorno requiere un análisis minucioso, ya que es aleatorio y continuo, y el número de percepciones y acciones posibles es infinito [7].

Las principales dificultades de los robots móviles se enfocan en la navegación y la planificación de trayectorias, temas que han sido objeto de investigación durante décadas. Como resultado, se han presentado y aplicado diversas metodologías para abordar el problema de la planificación de trayectorias de robots móviles. Estos estudios han dado lugar a tecnologías de percepción autónoma, que se desarrollan continuamente debido a la competitividad de esta industria [8]. Sin embargo, en comparación con el nivel de desarrollo de los métodos de navegación autónoma en entornos urbanos [9], la navegación autónoma en escenarios todo terreno no se ha estudiado con la misma profundidad, lo que significa que existen importantes oportunidades para nuevas investigaciones en esta área.

Para conseguir una navegación autónoma, un robot debe comprender tanto su postura como el entorno externo, información necesaria para determinar una ruta óptima y transitable que le permita alcanzar de forma segura su objetivo sin asistencia humana en el bucle de control [10], [11]. Sin embargo, la navegación totalmente autónoma en entornos no estructurados aún no se ha logrado, a pesar de los importantes avances en informática e ingeniería.

La navegación autónoma de vehículos en vías urbanas ha suscitado gran interés debido a sus emergentes aplicaciones comerciales en todo el mundo. En este sentido, en M. Nadeem Ahangar et al. se realiza una revisión exhaustiva de tecnologías que utilizan los conceptos de percepción (sensado), comunicación y control, que además permiten la integración de vehículos autónomos en los complejos sistemas de tráfico urbano [12]. En Ankur Sarker et al. se revisa los tres pilares fundamentales para la navegación vehicular, además, los aspectos de control, destacando su sinergia para lograr una conciencia situacional robusta, fusionando datos y la consideración la intervención de usuario en el desarrollo de sistemas de transporte inteligentes seguros y eficientes[13] . Finalmente, diversos estudios destacan que se han propuesto modelos de tráfico, que han evolucionado y se han evaluado, hasta alcanzar una apropiada adecuación en el contexto emergente de los vehículos autónomos [14]. Así, el estudio de la literatura indica que, estos agentes requieren actualización de los paradigmas de modelado existentes, con el objetivo de incrementar la precisión e interacciones, para así predecir su impacto en la dinámica de movimiento autónomo.

En la actualidad, las soluciones existentes, como la programación basada en reglas o ciertos enfoques de inteligencia artificial, a menudo presentan limitaciones, ineficiencias o políticas de control deficientes [15]. Ante esta problemática, surge la necesidad de explorar un nuevo modelo de aprendizaje que se adapte mejor al desplazamiento de robots móviles. Este estudio se centra en el aprendizaje por refuerzo y específicamente en el algoritmo PPO para evaluar su efectividad [16]. La relevancia del tema radica en el impacto teórico y académico que los datos obtenidos podrían tener para futuros entornos reales o estudios similares. Se plantea la hipótesis de que la implementación de PPO como algoritmo de aprendizaje podría resultar en una mejora visible en las trayectorias y en los resultados cuantitativos del desplazamiento.

Finalmente, se destaca la organización del manuscrito, siendo que en la sección 2 se describe la metodología aplicada en el trabajo, la sección 3 describe los resultados alcanzados, finalmente la sección 4 describe las conclusiones y posibles trabajos futuros.

2 Métodos y Materiales

Con el objetivo de alcanzar una navegación autónoma en el vehículo de ruedas "*A*", es importante destacar que el entorno de movimiento "*E*" en el que se desplaza, mantiene una estructura definida, sin obstáculos. En consecuencia, "A" debe partir desde un punto de inicio definida "p_i", realizar un conjunto de movimientos "*M*", hasta alcanzar un punto final "p_f". Entonces, para alcanzar este objetivo, "*A*" debe aprender cual es el conjunto de "*M*" más adecuado, proceso que ha sido determinado por medio de una metodología de aprendizaje basado en recompensas, como se puede ver en la Figura 1.

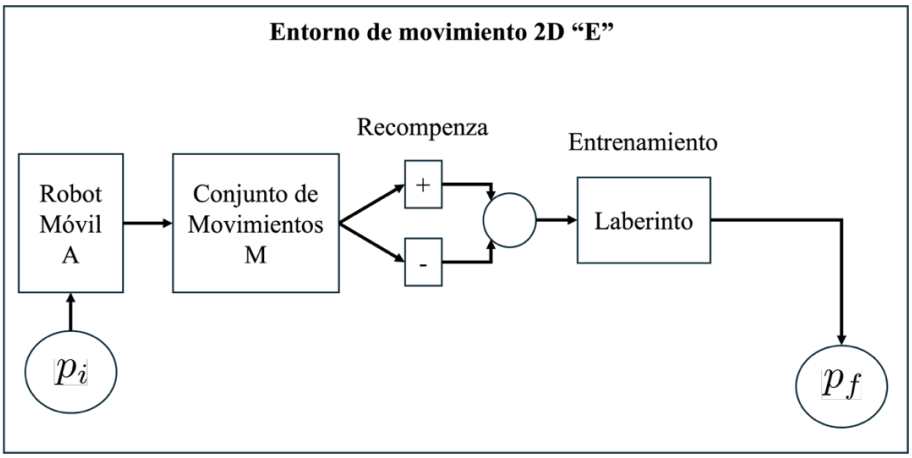


Figura 1. Problema de movimiento de Vehículo Móvil.

El objetivo es alinear el modelo de IA a partir del conjunto de "*M*", más allá de solo definir movimientos en un orden específico. El entrenamiento adecuado modifica (fine-tunear) afina los movimientos hasta maximizar la recompensa total esperada del modelo de recompensa. En ese sentido, a continuación, se describe un conjunto de conceptos importantes para el desarrollo del trabajo.

2.1 Definición de movimientos

Un aspecto de relevancia en el trabajo ha sido la definición de los movimientos iniciales. Así, se han determinado 3 instantes relevantes de movimiento siendo, giro a la izquierda, giro a la derecha y de frente. Posteriormente, se agregaron dos movimientos más para mejorar los desplazamientos hacia la izquierda y hacia la derecha, que han permitido un desplazamiento más preciso del vehículo.

2.2 Optimización de Movimiento y Trayectoria Mejorada

La optimización se ha definido como la reducción del número de pasos entre entrenamientos. Entonces, una trayectoria mejorada se refiere a la reutilización de una trayectoria anterior hasta un punto de decisión, desde el que el robot explorara una nueva trayectoria.

2.3 Asignación de Recompensas

Se han establecido recompensas positivas y negativas para guiar el comportamiento del robot. Por ejemplo, llegar a la meta, se otorga una recompensa positiva, mientras que colisionar tiene una recompensa negativa. Posteriormente, se han agregaron recompensas para mantener la línea recta y por la distancia mínima observada por el LIDAR [17].

A partir de los criterios descritos, se ha definido que, un movimiento se traduce como la ejecución de una orden de movimiento simultaneo de cada rueda, entonces, con el objetivo de alcanzar la meta, el robot móvil puede ejecutar los movimientos descritos en el Cuadro1.

Cuadro 1. Acción y descripción de movimientos.

Acción	Descripción
$\boldsymbol{movimiento_0}$	Recto: Las ruedas giran hacia adelante
$\boldsymbol{movimiento_1}$	Izquierda: la rueda izquierda retrocede y la derecha avanza
$\boldsymbol{movimiento_2}$	Derecha: La rueda derecha retrocede y la izquierda avanza
$\boldsymbol{movimiento_3}$	Diagonal izquierda: rueda derecha avanza más rápido que la izquierda.
$\boldsymbol{movimiento_4}$	Diagonal derecha: rueda izquierda avanza más rápido que la derecha.

Estos movimientos han sido calculados a partir de la ecuación (1)

$$x' = x + v * \cos(\theta) * t$$

$$y' = y + v * sen(\theta) * t \quad (1)$$

$$\theta' = \theta \pm w * t$$

donde;

x= representa la coordenada en x en la que se encuentra el robot antes del movimiento,
y= representa la coordenada en y en la que se encuentra el robot antes del movimiento,
Θ= representa el ángulo en el que el robot está apuntando antes del movimiento,
v= representa la velocidad lineal del robot, es una constante de valor 0.2 m/s,
w= representa la velocidad angular del robot, es una constante de valor 0.25 rad/s,
t= es el tiempo que dura un movimiento.

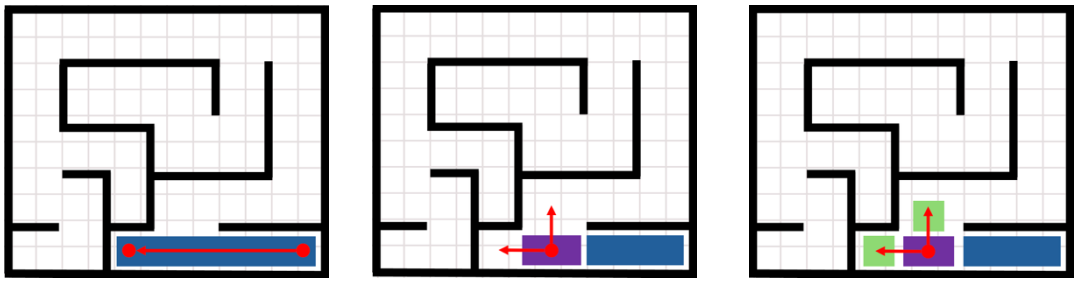
Por otra parte, es importante destacar el Cuadro 2, donde se define el conjunto de recompensas a los diferentes M.

Cuadro 2. Tabla de recompensas.

Acción	Recompensa
Alcanzar el objetivo	+100
Chocar con un objeto	-10
Mantener la línea recta en movimiento	+0.1
Distancia mínima observada por LIDAR	$Recompensa \ +-= \ 0.3 \ * \ min\_distancia$

2.4 Ejemplo práctico

En entrenamiento inicia con la configuración del entorno de movimiento E, la definición del vehículo móvil de ruedas A y los puntos p_i y p_f dentro de E. Así, la Figura 2 muestra situaciones dentro de un E discretizado de movimiento para A. En consecuencia, para que A alcance p_f, debe completar un número de movimientos no definidos desde p_i hasta p_f , además, la condición adicional de cero colisiones.



(a) Trayectoria de ejemplo y primer avance (línea roja con flecha), entre dos puntos (puntos rojos).

(b) Bifurcación en la trayectoria. El cuadro púrpura representa el punto de bifurcación, mientras las flechas rojas indican las opciones de camino.

(c) Decisión.

Figura 2. Entorno de movimiento 2D para A.

En términos de sencillo entendimiento, la trayectoria puede ser representada como una secuencia de pasos, descritos en la ecuación (2). Donde, la Figura 2(a) describe un tramo de trayectoria (en azul), recorrido entre 2 puntos (línea roja).

$$T = paso_1 + paso_2 + paso_3 + \ldots + paso_n \quad (2)$$

donde, el $paso_1$ no necesariamente tiene que pertenecer al $movimiento_1$, es decir, el $paso_1$ puede ser parte del amplio conjunto de los n movimientos, por tanto, $paso_n \subseteq movimiento_n$, a lo largo de la trayectoria.

Es importante destacar que, bajo una situación de decisión considerada como bifurcación (es decir, existencia de más de una opción de camino para avanzar) a lo largo de E, el robot móvil A debe tomar decisiones, traducidas como giros y avances. La Figura 2(b) representa la situación de bifurcación, mientras que la ecuación (3) describe la expresión formal, tal que:

$$TB = paso_1 + paso_2 + paso_3 \ldots bif \quad (3)$$

Así, la decisión de giro o avance de trayectoria frente la bifurcación significa una trayectoria mejorada, determinada a partir de la reutilización de una trayectoria superada hasta su última bifurcación (ver eq. 4), así, el robot móvil A elige el camino diferente y puede explorar el entorno E hasta alcanzar la trayectoria correcta, ver Figura 2(c).

$$TM = TB + paso1 + paso2 + paso3 \ldots pasox \quad (4)$$

Por tanto, A se puede entrenar desde p_i hasta alcanzar p_f en un número no determinado de pasos y movimientos.

3 Resultados

Con el objetivo de completar el modelo descrito, se ha diseñado un conjunto de experimentos, a partir de la siguiente definición de tecnologías. El Sistema Operativo para Robots (ROS 1 Noetic) brindo el apoyo necesario para integrar las herramientas necesarias para el desarrollo del trabajo. El aprendizaje se ha completado a través de las bondades de Deep Learning, en específico en combinación de redes neuronales y aprendizaje por refuerzo, siendo el algoritmo

de uso PPO (Proximal Policy Optimization). Con objeto de completar las simulaciones se utilizó Gazebo debido a su compatibilidad con ROS, su facilidad de comunicación, su estabilidad y la gran comunidad de soporte disponible, finalmente, y el modelo de robot móvil Turtlebot3 compatible con toda la tecnología antes descrita.

La Figura 3, describe un primer escenario de trabajo, donde se puede ver el robot móvil, además, la interfaz del simulador, que indica los parámetros del modelo, de la física, de la atmósfera, etc.

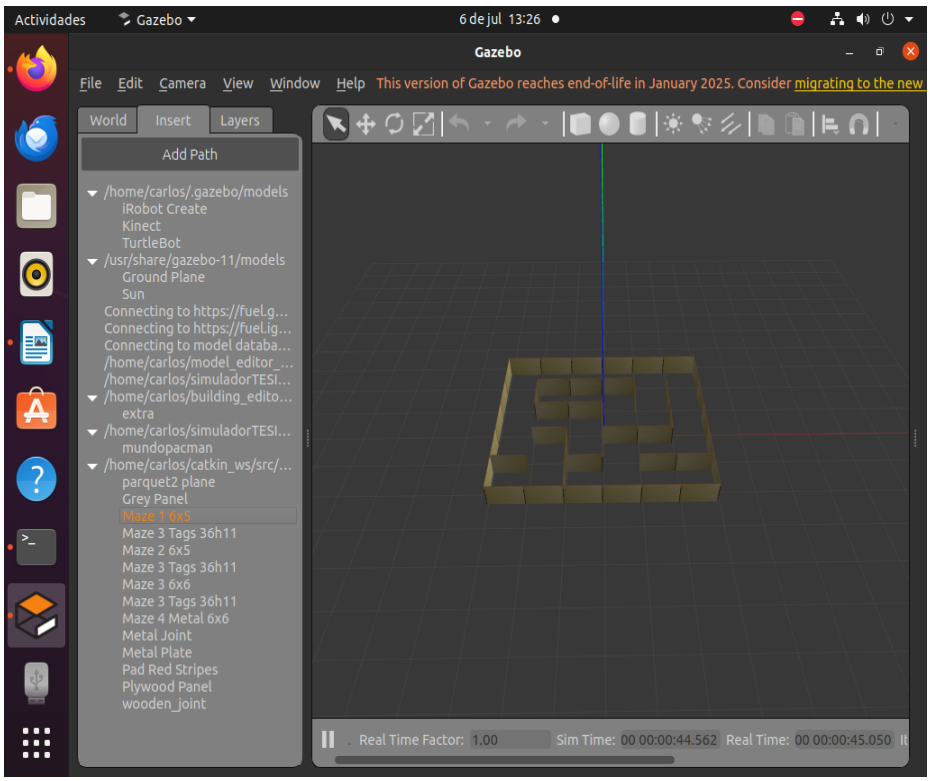


Figura 3. Simulación en funcionamiento.

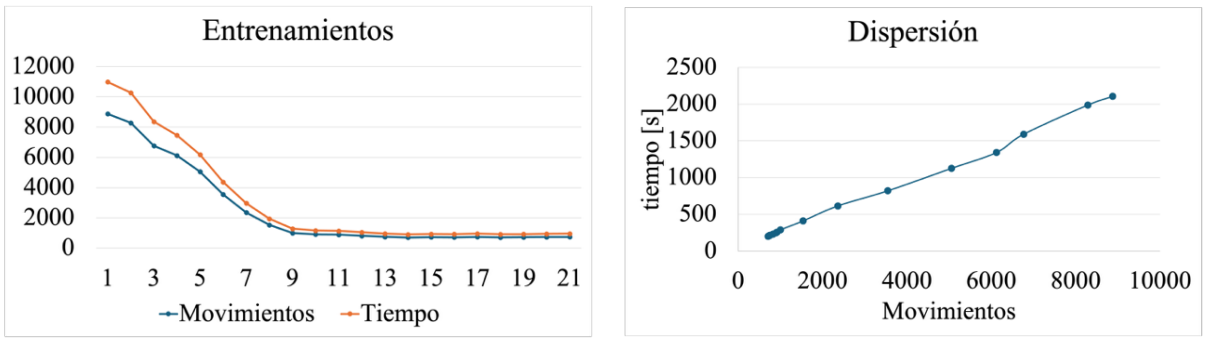
Con el objetivo de alcanzar un entrenamiento adecuado, se configuro un total de 10 mil pasos por sesión de entrenamiento (este número de pasos asegura que el robot realice un entrenamiento mínimo), al mismo tiempo, se asegura la obtención de un conjunto pertinente de datos que puedan ser debidamente analizados. Así, después de varios entrenamientos y ajustes se consiguió resultados satisfactorios, donde, el robot alcanzar la meta en el entrenamiento número 10. Es decir, con un número de 8876 movimientos y con un tiempo de 2106 segundos, por sesión de entrenamiento, el robot disminuye esos pasos considerablemente hasta llegar a la mejor cifra que fue en el entrenamiento número 23 con 708 pasos y un tiempo de 202 segundos, así, el Cuadro 3 realiza une breve descripción de la información de los diferentes entrenamientos (cabe destacar que el Cuadro inicia desde el décimo entrenamiento).

Cuadro 3. Tabla de datos de los entrenamientos.

Datos	Movimientos	Tiempo (s)
Entrenamiento 10	8,876	2,106
Entrenamiento 11	8,284	1,987
Entrenamiento 12	6,767	1,591
Entrenamiento 13	6,122	1,342
Entrenamiento 14	5,053	1,126

Entrenamiento 15	3,545	821
Entrenamiento 16	2,359	615
Entrenamiento 17	1535	411
Entrenamiento 18	1000	290
Entrenamiento 19	915	256
Entrenamiento 20	895	250
Entrenamiento 21	820	230
Entrenamiento 22	750	215
Entrenamiento 23	708	202
Entrenamiento 24	730	208
Entrenamiento 25	721	207
Entrenamiento 26	745	214
Entrenamiento 27	715	205
Entrenamiento 28	723	207
Entrenamiento 29	737	211
Entrenamiento 30	748	215

Por otra parte, la Figura 4 muestra los cambios descritos. La figura 4(a) describe los movimientos y tiempo de entrenamiento, así, se puede apreciar los rangos de estabilidad, es decir, el robot ya no puede seguir optimizando porque ya no encontró trayectorias y por ende ya no puede seguir generando más ni optimizar más el movimiento, además que el tiempo se establece en rangos tan estables, similar a una función lineal constante. Mientras, la Figura 4(b) muestra la dispersión de esta información, información relevante debido a que muestra la relación que hay entre tiempo y movimientos, la información muestra un descenso de derecha a izquierda lo que solo confirma que mientras el robot aprende y optimiza las trayectorias, más bajan los valores del campo movimiento y tiempo, lo que implica mejores resultados en la línea de tendencia.



(a) (b)

Figura 4. Disminución de movimientos y tiempo, durante los entrenamientos.

De esta forma, la curva de la Figura 4 muestra como el robot móvil presenta un aprendizaje exponencial decreciente, pues el número de pasos y el tiempo empleado en cada entrenamiento es menor llegando a una relativa estabilidad a partir del entrenamiento número 21.

Entonces, se puede realizar una predicción de resultados en los siguientes entrenamientos a partir de la ecuación (5).

$$(x) = \begin{cases} 5800e^{-0.25(x-10)} + 3000, & 10 \leq x \leq 15 \\ 1600e^{-0.35(x-16)} + 700, & 16 \leq x \leq 21 \\ 750 + 2\sin(0.5(x-22)), & 22 \leq x \leq 30 \end{cases} \quad (5)$$

Es importante recalcar que se deben cumplir las condiciones expuestas según los rangos de valores.

4 Conclusiones

El PPO (Proximal Policy Optimization) se destaca como una técnica de aprendizaje por refuerzo altamente eficaz para optimizar el movimiento y las trayectorias de robots con ruedas. Esta eficacia se basa en el análisis de los resultados, que demuestran una mejora clara en el aprendizaje, lo que lleva a la estabilización del comportamiento y una optimización progresiva.

Un aspecto fundamental de este proceso es la correcta implementación de un sistema de recompensas y la modificación del código. Ambas acciones deben ir de la mano de una monitorización constante del comportamiento del robot, ya que sin esta observación es imposible determinar las mejoras necesarias. En este sentido, la clave del éxito del aprendizaje por refuerzo con PPO radica en la coherente integración de la programación, el análisis del comportamiento y los ajustes continuos. Esto convierte el entrenamiento de robots en un proceso dinámico e iterativo, donde la calidad de la retroalimentación es vital para alcanzar los resultados deseados.

Para la implementación en entornos reales, es importante considerar las especificaciones técnicas del robot TurtleBot y sus sensores, especialmente el LIDAR. Este sensor, utilizado en el proyecto, proporcionó datos fiables para una navegación precisa y una toma de decisiones informada. Sin embargo, en un contexto real, factores como la resolución del sensor, su rango de detección, las condiciones ambientales y la capacidad de procesamiento del robot deben evaluarse cuidadosamente para asegurar un rendimiento seguro y eficiente.

Finalmente, este proyecto puede ser ampliado en escenarios dinámicos, donde los obstáculos se mueven o las condiciones del entorno cambian. Adaptar el entrenamiento del robot a estas situaciones permitiría evaluar su capacidad de respuesta en contextos no estáticos, algo crucial para aplicaciones reales en áreas como la robótica de servicio, la logística autónoma y la exploración.

Financiación

Los autores agradecen a la Facultad de Ingeniería Industrial, Universidad de Guayaquil, Guayaquil 090514, Ecuador.

Declaración de conflicto de intereses

Los autores no han declarado ningún posible conflicto de intereses en relación con esta investigación, la autoría y/o la publicación de este artículo.

Referencias

[1] M. Häselich, M. Arends, N. Wojke, F. Neuhaus, and D. Paulus, “Probabilistic terrain classification in unstructured environments,” *Rob. Auton. Syst.*, vol. 61, no. 10, pp. 1051–1059, Oct. 2013, doi: 10.1016/j.robot.2012.08.002.

[2] L. Wijayathunga, A. Rassau, and D. Chai, “Challenges and Solutions for Autonomous Ground Robot Scene Understanding and Navigation in Unstructured Outdoor Environments: A Review,” *Applied Sciences*, vol. 13, no. 17, p. 9877, Aug. 2023, doi: 10.3390/app13179877.

[3] F. Qi, “Local probabilistic terrain estimation for navigation in unstructured environments,” in *Third International Conference on Machine Vision, Automatic Identification, and Detection (MVAID 2024)*, R. Jin, Ed., SPIE, Aug. 2024, p. 120. doi: 10.1117/12.3036525.

[4] M. J. Procopio, J. Mulligan, and G. Grudic, “Learning terrain segmentation with classifier ensembles for autonomous robot navigation in unstructured environments,” *J. Field Robot.*, vol. 26, no. 2, pp. 145–175, Feb. 2009, doi: 10.1002/rob.20279.

[5] A. Cantos-Andrade and F. Samaniego, “Diseño y construcción de chasis para un robot móvil de ruedas para la exploración de terrenos irregulares,” *Revista Digital Científica Causalidad (caui3)*, vol. 1, no. 1, pp. 1–1, 2025, doi: 10.70929/caui3.v1i1.2wrkyz89.

[6] B. K. Patle, G. Babu L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, “A review: On path planning strategies for navigation of mobile robot,” *Defence Technology*, vol. 15, no. 4, pp. 582–606, Aug. 2019, doi: 10.1016/j.dt.2019.04.011.

[7] A. N. A. Rafai, N. Adzhar, and N. I. Jaini, “A Review on Path Planning and Obstacle Avoidance Algorithms for Autonomous Mobile Robots,” *Journal of Robotics*, vol. 2022, pp. 1–14, Dec. 2022, doi: 10.1155/2022/2538220.

[8] L. G. Galvao, M. Abbod, T. Kalganova, V. Palade, and M. N. Huda, “Pedestrian and Vehicle Detection in Autonomous Vehicle Perception Systems—A Review,” *Sensors*, vol. 21, no. 21, p. 7267, Oct. 2021, doi: 10.3390/s21217267.

[9] J. Crespo, J. C. Castillo, O. M. Mozos, and R. Barber, “Semantic Information for Robot Navigation: A Survey,” *Applied Sciences*, vol. 10, no. 2, p. 497, Jan. 2020, doi: 10.3390/app10020497.

[10] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection,” *Int. J. Rob. Res.*, vol. 37, no. 4–5, pp. 421–436, Apr. 2018, doi: 10.1177/0278364917710318.

[11] S. Benford, C. Magerkurth, and P. Ljungstrand, “Bridging the physical and digital in pervasive gaming,” *Commun. ACM*, vol. 48, no. 3, pp. 54–57, Mar. 2005, doi: 10.1145/1047671.1047704.

[12] M. N. Ahangar, Q. Z. Ahmed, F. A. Khan, and M. Hafeez, “A Survey of Autonomous Vehicles: Enabling Communication Technologies and Challenges,” *Sensors*, vol. 21, no. 3, p. 706, Jan. 2021, doi: 10.3390/s21030706.

[13] A. Sarker *et al.*, “A Review of Sensing and Communication, Human Factors, and Controller Aspects for Information-Aware Connected and Automated Vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 7–29, Jan. 2020, doi: 10.1109/TITS.2019.2892399.

[14] Ł. Łach and D. Svyetlichnyy, “Comprehensive Review of Traffic Modeling: Towards Autonomous Vehicles,” *Applied Sciences*, vol. 14, no. 18, p. 8456, Sep. 2024, doi: 10.3390/app14188456.

[15] W. Qiang, L. Yang, and H. Jin, “Efficient and Robust Malware Detection Based on Control Flow Traces Using Deep Neural Networks,” *Comput. Secur.*, vol. 122, p. 102871, Nov. 2022, doi: 10.1016/j.cose.2022.102871.

[16] Y. Gu, Y. Cheng, C. L. P. Chen, and X. Wang, “Proximal Policy Optimization With Policy Feedback,” *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 52, no. 7, pp. 4600–4610, Jul. 2022, doi: 10.1109/TSMC.2021.3098451.

[17] N. Li *et al.*, “A Progress Review on Solid-State LiDAR and Nanophotonics-Based LiDAR Sensors,” *Laser Photon. Rev.*, vol. 16, no. 11, Nov. 2022, doi: 10.1002/lpor.202100511.