Memorias del XX Congreso Latinoamericano de Control Automático (CLCA 2025)
13-17 de Octubre, 2025. Cancún, Quintana Roo, México

Hybrid Control Based on Deep Reinforcement Learning and Dynamic Window Approach for Robotic Navigation

C. Vasquez-Jalpa*, M. Nakano*, M. Velasco-Villa**

** Graduate Section, Mechanical and Electrical Engineering School, Instituto Politecnico Nacional, Av. Santa Ana 1000, Mexico City 04440 (cvasquezj2400@alumno.ipn.mx, mnakano@ipn.mx)*
*** Center for Research and Advanced Studies of Instituto Polytechnic Nacional, Av. Instituto Politecnico Nacional 2508, San Pedro Zacatenco, Mexico City (velasco@cinvestav.mx)*

Abstract: This work presents a hybrid mobile robot navigation system for simulated environments. The system integrates the Dynamic Window Approach (DWA) with a deep reinforcement learning (DRL) using an actor-critic architecture. Initially, the robot uses DWA, while concurrently, a DRL agent generates alternative control speeds using actor and critic networks. An entropy-based mechanism dynamically weights the DWA and DRL speeds, transitioning gradually to DRL control. The system's performance is evaluated using success rate and trajectory length, comparing the hybrid approach to DWA and DRL alone. Results demonstrate improved robustness and performance, particularly in complex scenarios.

Keywords: Robotic navigation, Deep reinforcement learning, Entropy, Dynamic window approach, Hybrid control, Mobile robot.

1. INTRODUCCIÓN

La navegación autónoma de robots móviles es un desafío crucial en robótica. Estos entornos, caracterizados por la imprevisibilidad de la geometría y la dinámica del entorno, exigen sistemas de control robustos y adaptativos. Los algoritmos de planificación de trayectorias clásicos, que requieren mapas precisos del entorno, presentan limitaciones significativas en escenarios donde la información del entorno es incompleta o dinámica. El Algoritmo de Ventana Dinámica (DWA) es efectivo para la planificación local en entornos parcialmente conocidos; sin embargo, Bodong y Kim (2024) señalan que puede carecer de la capacidad de adaptación necesaria para optimizar el rendimiento en situaciones complejas y cambiantes. Además, esto se ha demostrado en diversos trabajos que exploran alternativas basadas en aprendizaje por refuerzo profundo (DRL) para mejorar la eficiencia y robustez en entornos con obstáculos estáticos y dinámicos (Laiyi et al., 2022; Haisen et al., 2023).

El aprendizaje por refuerzo profundo (DRL) ha surgido como una técnica prometedora para la navegación robótica en entornos no estructurados. Algoritmos de DRL, como los métodos actor-crítico (Liang et al., 2020; Xiaoyu et al., 2022), permiten a los robots aprender políticas de control óptimas a través de la interacción con el entorno. Sin embargo, el entrenamiento de estos agentes puede ser computacionalmente costoso y requerir un gran número de iteraciones, especialmente en entornos de alta dimensionalidad o con restricciones de tiempo real (Laiyi et al., 2022; Haisen et al., 2023). Además, la eficiencia del entrenamiento puede verse afectada por la complejidad del espacio de estados y la formulación de la función de recompensa (An et al., 2024; Fanfan et al., 2024).

Este trabajo propone un enfoque híbrido que combina la robustez del algoritmo DWA con la capacidad de aprendizaje adaptativo del DRL para superar las limitaciones de los métodos de navegación tradicionales. A su vez cuenta con un mecanismo de selección basado en la entropía que permite una transición gradual del control, desde el algoritmo DWA hasta el control aprendido por el agente. Este enfoque híbrido busca aprovechar las ventajas de ambos métodos, logrando un sistema de navegación eficiente en los entornos.

El resto del artículo se estructura de la siguiente manera: la Sección 2 describe el problema abordado y el tipo de robot utilizado; la Sección 3 presenta los fundamentos teóricos del algoritmo propuesto; la Sección 4 detalla el algoritmo de navegación híbrido, que combina DWA y DRL; la Sección 5 evalúa el rendimiento del algoritmo mediante simulaciones, analizando las velocidades generadas y el comportamiento del parámetro α; finalmente, la Sección 6 presenta las conclusiones y el trabajo futuro.

2. PROBLEMA DE NAVEGACIÓN

En este trabajo se considera el problema de navegación de un robot móvil del tipo diferencial el cual se desplaza en un ambiente congestionado. Se considera un robot móvil descrito por el modelo cinemático,

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\,sen\theta \\ \dot{\theta} &= w \end{aligned} \tag{1}$$

donde (x, y) representan la posición del punto medio de las ruedas en el plano (X-Y), θ describe la orientación del vehículo con respecto al eje X. v corresponde a la velocidad lineal y w a la velocidad rotacional. El vehículo se describe en la Fig. 1.

El objetivo planteado es lograr que el robot móvil, a partir de una condición inicial $[x(0), y(0), \theta(0)]$, alcance la posición final establecida por una meta (x_m, y_m, θ_m) esto llevado a cabo en un ambiente congestionado de obstáculos estáticos.

Sin pérdida de generalidad se considera un robot navegando en un plano cartesiano X-Y con origen (0,0) desde el instante de tiempo cero hasta que llega a la meta. En una aplicación

https://doi.org/10.58571/CNCA.AMCA.2025.038

real, un sensor Light Detection and Ranging (LiDAR) puede definir una ventana dinámica (entorno) en la cual se tomaron en cuenta los obstáculos existentes. El robot utilizará el obstáculo más cercano como punto de referencia para su desplazamiento, hasta hallar otro más cercano. El mecanismo de entropía y el sistema de recompensas del DRL guían la exploración del robot, minimizando la probabilidad de ciclos repetitivos alrededor de los obstáculos. Si la meta se detecta dentro del rango del LIDAR, se prioriza como punto de referencia para optimizar la ruta.

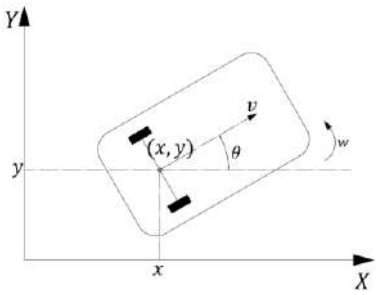

Figura 1. Robot móvil tipo diferencial considerado.

3. PRELIMINARES

En esta sección, describimos brevemente los conceptos básicos utilizados en el algoritmo propuesto, que son el Aprendizaje por Refuerzo Profundo y el Algoritmo de Ventana Dinámica.

3.1 Aprendizaje por refuerzo profundo (DRL)

El DRL es un subcampo del aprendizaje automático que se centra en el entrenamiento de agentes para tomar decisiones óptimas en un entorno dado. Un agente aprende a través de la interacción con el entorno, recibiendo recompensas o penalizaciones por sus acciones, ver Fig. 2. El objetivo del agente es maximizar la recompensa acumulada a lo largo del tiempo. DRL combina técnicas de aprendizaje por refuerzo con redes neuronales profundas para representar funciones de valor y políticas complejas. Los métodos actor-crítico son una clase popular de algoritmos DRL que utilizan dos redes neuronales: una red actor que define la política (es decir, la forma en que el agente selecciona acciones) y una red crítica que evalúa el valor de los estados y las acciones. El entrenamiento implica iterativamente actualizar ambas redes para mejorar la política y la estimación del valor. (Ebrahim et al., 2024) proporciona una revisión exhaustiva de los algoritmos DRL.

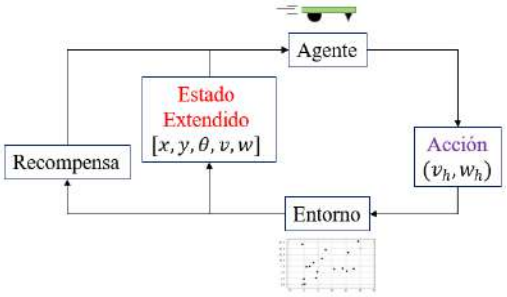

Figura 2. Ciclo de interacción en el aprendizaje por refuerzo profundo.

En la Fig. 2, el Agente recibe el estado extendido del entorno y ejecuta una acción. Como resultado, el Agente evoluciona y el entorno devuelve una nueva observación del estado junto con una señal de recompensa. En la Sección 4, se detalla el algoritmo que le permite al robot pasar del estado extendido a la acción híbrida (v_h, w_h).

3.2 Algoritmo de Ventana Dinámica (DWA)

El DWA es un algoritmo de planificación de movimiento local que considera las limitaciones dinámicas del robot, como la velocidad máxima, la aceleración máxima y el radio de giro mínimo. A diferencia de los algoritmos de planificación global que requieren un mapa completo del entorno, DWA opera en una ventana de tiempo y espacio limitada, utilizando información sensorial local para generar velocidades de control. Para cada velocidad dentro de la ventana dinámica, DWA simula la trayectoria del robot durante un corto periodo de tiempo y evalúa la trayectoria utilizando una función de costo que considera la distancia al objetivo, la proximidad a los obstáculos y la velocidad. La velocidad que produce la trayectoria con el menor costo se selecciona y se aplica al robot. (Yanjie y Norzalilah, 2024) presenta una descripción detallada del algoritmo DWA.

4. DESCRIPCIÓN DEL ALGORITMO DE NAVEGACIÓN

Esta sección describe el algoritmo propuesto para la navegación autónoma de un robot móvil en entornos no controlados. El algoritmo combina el Algoritmo de Ventana Dinámica con un modelo de aprendizaje por refuerzo profundo con arquitectura actor-crítico. El robot móvil considerado puede verse en la Fig. 1 y su evolución en el plano cartesiano puede representarse mediante el modelo cinemático de (1).

4.1 Representación del Estado (s)

Para efectos del algoritmo de navegación, se considera un estado extendido del robot formado por:

$$s = [x, y, \theta, v, w] \tag{2}$$

4.2 Espacio de Acciones

El espacio de acciones consiste en pares de velocidades de control: $[v, w]$.

En el DRL, la acción predicha está dada por:

$$a_p = \pi_\varphi(s) \tag{3}$$

donde:

- a_p: acción predicha (v_p, w_p).
- π: política de la red neuronal (distribuciones de probabilidades de las acciones).
- φ: pesos de conexión de la red actor.

4.3 Función de Recompensa (r)

La función de recompensa se diseña para guiar al robot hacia el objetivo mientras evita obstáculos, en la forma,

 https://doi.org/10.58571/CNCA.AMCA.2025.038

$$r = \begin{cases} r_{arrive} = 10 \\ Si\ dit_{obj(i)} < dist_{obj(i-1)} = 2 \\ -1; de\ lo\ contrario \end{cases} \quad (4)$$

donde:

- $dit_{obj(i)}$: Distancia al objetivo en el instante *i*.
- $dit_{obj(i-1)}$: Distancia al objetivo en el instante *i-1*.

Esta función fomenta que el agente reduzca progresivamente la distancia al objetivo y penaliza trayectorias que no representen un avance real, garantizando un aprendizaje orientado alcanzar meta.

4.4 Calidad Q

La calidad Q representa cuan eficiente fue la acción realizada a_p iniciando en el estado *s,* dicho valor es calculado con los valores Q de cada una de las redes críticas, como se tienen 2 redes resulta:

$$Q = r + \gamma * \min\left(Q_1(s, a_p; \theta_1), Q_2(s, a_p; \theta_2)\right) \quad (5)$$

donde:

- γ: Factor de descuento con valor de 0.9.
- Q_1: Valor Q de la primera red crítica cuyas entradas son: s, a_p ; con parámetros θ_1.
- Q_2: Valor Q de la segunda red crítica cuyas entradas son: s, a_p ; con parámetros θ_2.

4.5 Arquitectura de las Redes Neuronales (Actor y Crítico)

Este trabajo emplea una arquitectura de redes neuronales basada en el algoritmo Soft Actor-Critic (SAC) (Shuhuan et al., 2025), modificada para incluir una segunda red crítica. Esta modificación mejora la política del agente, en comparación con la arquitectura SAC original que emplea una sola red crítica (Husam y Oscar, 2024), además se emplea la función de activación *Rectified Linear Unit (ReLU)*, definida como $f(x) = \max(0, x)$. Esta función no fue seleccionada arbitrariamente, sino que se adopta siguiendo el diseño propuesto en la investigación de Husam y Oscar (2024), quienes demostraron su efectividad en problemas de navegación autónoma con aprendizaje por refuerzo profundo. La arquitectura neuronal se ilustra en la Fig. 3 y está compuesta por cuatro capas:

- Capas completamente conectadas (FC): Dos capas FC, cada una con 256 neuronas, procesan la información del vector de estado. Estas capas realizan transformaciones no lineales de la entrada mediante ReLU, extrayendo características relevantes para la toma de decisiones.
- Capas de salida específicas: La red actor genera la distribución de probabilidad de las acciones (v, w) utilizando softmax para limitar las velocidades. Y en el caso de las redes críticas gemelas, en su salida, se estima el valor de la calidad de la acción realizada (Q) de los estados-acción utilizando una función sigmoidal.

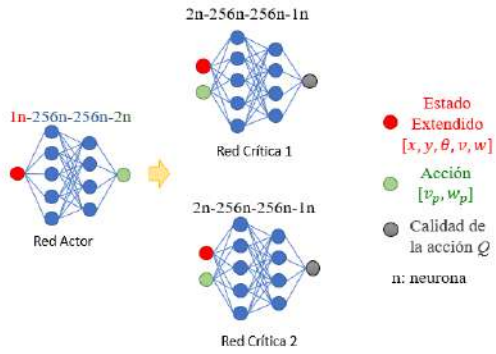

Figura 3. Representación gráfica de la arquitectura de DRL.

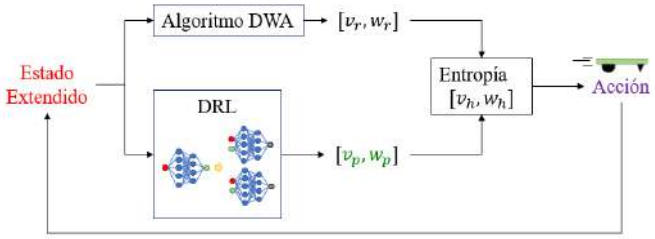

Figura 4. Esquema del sistema híbrido propuesto basado en DRL y DWA.

La Fig. 4 ilustra el sistema de navegación híbrido propuesto. Tanto el algoritmo DWA como el DRL generan velocidades reales (v_r, w_r) y predichas (v_p, w_p), respectivamente. Luego, un mecanismo basado en la entropía combina estas salidas para producir velocidades híbridas (v_h, w_h) que guía al robot al siguiente estado.

4.6 Incorporación de la Entropía para la Acción Híbrida

Para mejorar la exploración del espacio de acciones y mitigar el riesgo de convergencia prematura a óptimos locales, se introduce un mecanismo de transición basado en la entropía. Este mecanismo pondera las acciones sugeridas por el algoritmo de aprendizaje por refuerzo DRL y el DWA, utilizando un parámetro α que se ajusta dinámicamente en función de la entropía del sistema.

La entropía *H* se calcula a partir de la distribución de probabilidades de las acciones propuestas por la red actor del agente RL, mostrada en la siguiente ecuación. (Shuhuan et al., 2025)

$$H = -(v_p * \log(v_p) + w_p * \log(w_p)) \quad (6)$$

El parámetro α se encarga de hacer la transición entre los valores predichos por la red y los valores dados por el control DWA. α se actualiza mediante una función sigmoide que depende de *H*, un umbral de entropía (H_{umbral}), y una constante de velocidad de transición (*k=0.5*). (Shuhuan et al., 2025)

$$\alpha = 1 / (1 + exp(-k * (H_{umbral} - H))) \quad (7)$$

Considerando que la entropía de una distribución de probabilidad siempre es positiva y, para el caso de una distribución discreta con *n* posibles resultados, su valor

 https://doi.org/10.58571/CNCA.AMCA.2025.038

máximo es $log_2(n)$ (Jinding et al., 2024). En este caso, se tienen dos salidas (velocidad lineal y angular), así que el valor máximo de entropía es $log_2(2) = 1$. Sin embargo, dado que se usa el logaritmo natural (ln) en lugar del logaritmo en base 2, el valor máximo de nuestra entropía será $ln(2) \approx 0.693$.

Dado esto, se experimentó con valores de H_{umbral} dentro del rango *0.1 a 0.6*. Este rango cubre una gama de situaciones:

- *Valores bajos (0.1 - 0.3):* Indican que la red necesita tener una alta confianza (baja entropía) para tomar el control. Esto es útil si la tarea es compleja o si requiere una transición muy gradual.
- *Valores medios (0.3 - 0.5):* Representan un equilibrio entre la confianza de la red y la velocidad de la transición.
- *Valores altos (0.5 - 0.6):* Indican que la red puede tomar el control incluso con una incertidumbre moderada. Esto es útil si la tarea es simple o si se quiere una transición más rápida.

Para nuestro caso, el valor usado fue de $H_{umbral} = 0.3$. Tomando en cuenta lo anterior, es posible obtener las velocidades híbridas con (8) y (9).

$$v_h = \alpha(v_p) + (1 - \alpha)(v_r) \quad (8)$$

$$w_h = \alpha(w_p) + (1 - \alpha)(w_r) \quad (9)$$

El parámetro α se define como un coeficiente de transición dinámico que pondera las velocidades propuestas por el DWA y el DRL. Su valor depende de la entropía del sistema, permitiendo una transición gradual: cuando α→0, el control se asemeja al DWA, y cuando α→1, se aproxima al DRL. De esta manera, α regula el grado de influencia del controlador a lo largo del proceso de navegación. En términos prácticos, α funciona como un regulador adaptativo: al inicio otorga mayor peso al control clásico del DWA para estabilizar la navegación, y conforme el agente gana confianza (menor entropía), transfiere progresivamente el control hacia el DRL.

5. EVALUACIÓN DEL ALGORITMO PROPUESTO

Para evaluar el rendimiento del algoritmo propuesto, se realizaron experimentos de navegación autónoma en entornos simulados donde se proyectan obstáculos de manera aleatoria para cada episodio, es decir, ningún entorno se parece a otro. Un ejemplo de episodio se muestra en la Fig. 5.

En la Fig. 5 se muestra la posición de los obstáculos marcados con puntos negros, mientras que la "x", es decir, el objetivo se mantiene en la misma posición en cada simulación, al igual que el agente representado por el rectángulo cuyo punto de inicio es (0,0) y su trayectoria local está dada por la línea naranja.

La trayectoria realizada por el agente en el entorno se muestra en la Fig. 6. Enfaticemos que, durante la simulación, el DWA genera trayectorias locales las cuales se van modificando en cada paso. Sin embargo, en la Fig. 6 sólo se muestra la trayectoria efectiva seguida hasta alcanzar la meta.

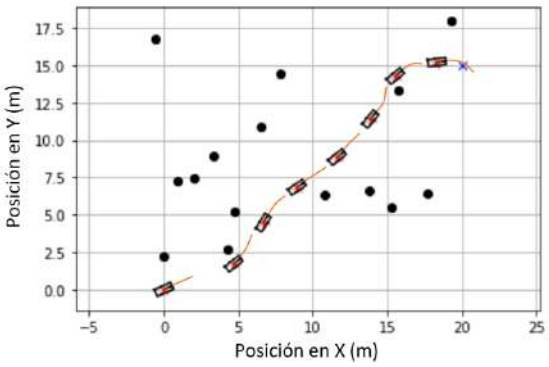

Figura 5. Ejemplo de entorno de navegación autónoma.

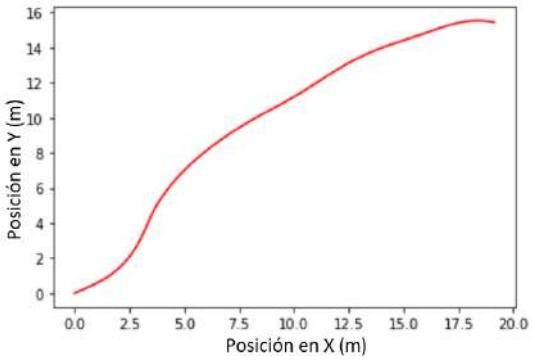

Figura 6. Trayectoria final recorrida por el agente desde (0,0) hasta la coordenada (20,15).

En la Fig. 7 se observa la evolución de la velocidad lineal a lo largo de los pasos del robot para tres componentes: v_r (velocidad del DWA), v_p (velocidad del DRL) y v_h (velocidad híbrida). A medida que avanza el tiempo, todas las velocidades presentan una ligera disminución, esto puede deberse a que el robot se acerca gradualmente al objetivo, lo que requiere una reducción de la velocidad para evitar sobrepasarlo. Además, v_h se sitúa consistentemente entre las velocidades generadas (v_r y v_p), lo que confirma su naturaleza como una combinación de ambos enfoques. Esta tendencia intermedia permite una navegación más suave y adaptable, evitando los cambios bruscos de velocidad que podrían ocurrir si se dependiera exclusivamente de uno de los algoritmos.

Por otra parte, el término *paso* se refiere al cambio de estado del agente definido por (2).

La Fig. 8 muestra la evolución de la velocidad angular en función de los pasos del robot para los tres métodos. Se observa que la velocidad w_r comienza con valores negativos considerables y decrece aún más con el tiempo, lo que indica una tendencia del DWA a realizar giros amplios. En contraste, la velocidad w_p, correspondiente al DRL, se incrementa gradualmente, reflejando una estrategia que favorece giros más controlados y estables conforme avanza el aprendizaje. La velocidad angular híbrida w_h se mantiene en un rango más moderado, equilibrando el comportamiento del DWA y la adaptación progresiva del DRL.

 https://doi.org/10.58571/CNCA.AMCA.2025.038

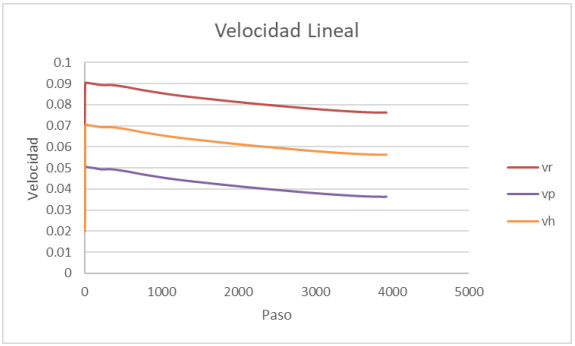

Figura 7. Evolución de las velocidades lineales generadas por el DWA (v_r), el DRL (v_p) y la combinación híbrida (v_h) a lo largo de los pasos del robot.

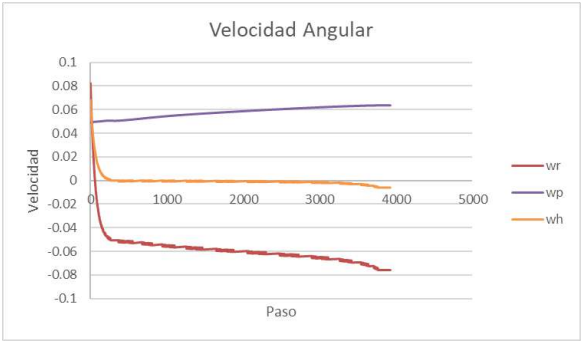

Figura 8. Evolución de las velocidades angulares generadas por el DWA (w_r), el DRL (w_p) y la combinación híbrida (w_h) a lo largo de los pasos del robot.

Para evaluar el desempeño del método híbrido se utilizaron dos métricas principales:

- *Tasa de éxito:* Porcentaje de pruebas en las que el robot alcanzó el objetivo dentro de un tiempo límite predefinido (10000 pasos de tiempo). Esta métrica cuantifica la capacidad del algoritmo para guiar al robot de manera eficiente hacia el objetivo, incluso en presencia de obstáculos. Se registraron las tasas de éxito tanto para el algoritmo híbrido como para el DWA y DRL utilizado de forma independiente, permitiendo una comparación directa de su rendimiento. Mostrado en la Tabla 1.
- *Tiempo de planificación de ruta:* El tiempo de planificación de ruta también es un indicador importante para medir la capacidad de navegación autónoma. Registramos el número de pasos planificados para 100 episodios, como se muestra en la Fig. 9.

Tabla 1. Tasa de éxito de los métodos de navegación

DWA	DRL	HÍBRIDO
91%	47%	96%

La Tabla 1 compara la tasa de éxito de tres enfoques distintos de navegación: DWA, DRL y el método HÍBRIDO, que integra ambos.

El método HÍBRIDO alcanza el mejor desempeño con una tasa de éxito del 96%, superando tanto al enfoque tradicional DWA (91%) como al basado únicamente en DRL (47%). Este resultado evidencia que, si bien DRL por sí solo aún no iguala la fiabilidad de DWA, su integración dentro de un esquema híbrido mejora sustancialmente el rendimiento.

Esto sugiere que el enfoque híbrido se beneficia de las fortalezas de ambos métodos: la capacidad reactiva y robusta de DWA frente a obstáculos, y la capacidad de adaptación y aprendizaje de políticas óptimas de DRL. La fusión a través de un mecanismo de selección basado en entropía permite obtener decisiones más eficaces, resultando en una mayor tasa de éxito.

En la Fig. 9, se observa que el método DWA presenta los menores tiempos de planificación, con una gran cantidad de episodios finalizados en menos de 500 pasos, lo que indica trayectorias cortas y eficientes. Por otro lado, el método DRL muestra un alto número de episodios que alcanzan el límite máximo de 10,000 pasos, lo que implica que en muchos casos el agente no logró llegar al objetivo, evidenciando inestabilidad y falta de confiabilidad en su comportamiento autónomo. En contraste, el enfoque híbrido logra reducir significativamente la cantidad de episodios fallidos respecto al DRL puro, manteniéndose en una zona intermedia, con una mayor proporción de episodios exitosos que no exceden los 5000 pasos.

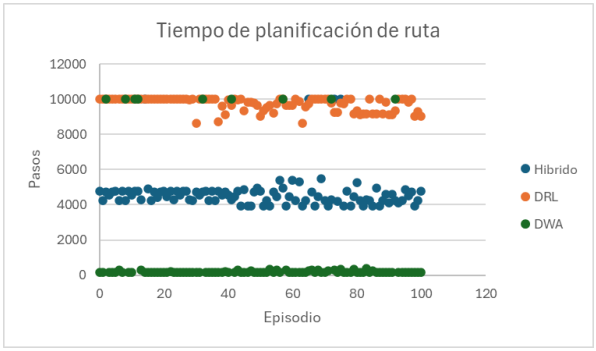

Figura 9. Comparación del tiempo de planificación de ruta, medido en pasos por episodio, entre los métodos DWA, DRL y el enfoque híbrido propuesto.

Adicionalmente, se analizó el comportamiento del mecanismo de transición basado en entropía, monitoreando la evolución del parámetro α a lo largo de las pruebas, ilustrado en Fig. 10. Esto permitió evaluar la efectividad del mecanismo para equilibrar la exploración y la explotación, y su influencia en la toma de decisiones del robot.

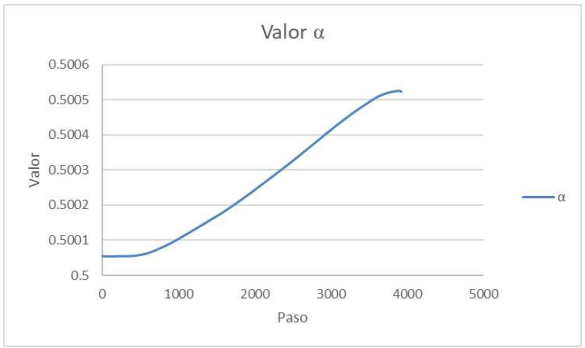

Figura 10. Evolución del parámetro α en la trayectoria de la Fig. 6.

 https://doi.org/10.58571/CNCA.AMCA.2025.038

Como se muestra en la Fig. 10, el valor de α se incrementa progresivamente a medida que avanzan los pasos del robot. En otras palabras, las velocidades propuestas por el algoritmo DWA van perdiendo peso, mientras que las generadas por el DRL adquieren mayor relevancia en el cálculo de las velocidades híbridas.

6. CONCLUSIÓN Y TRABAJO FUTURO

Este trabajo presentó un sistema de navegación híbrido para robots móviles que combina la eficiencia y robustez del Algoritmo de Ventana Dinámica (DWA) con la capacidad adaptativa del Aprendizaje por Refuerzo Profundo (DRL) mediante una arquitectura actor-crítico. Los resultados de las simulaciones mostraron que el enfoque híbrido propuesto, basado en la combinación del algoritmo DWA y el aprendizaje por refuerzo profundo (DRL), supera tanto al método DWA como al DRL por separado en términos de tasa de éxito y eficiencia. Con una tasa de éxito del 96%, el enfoque híbrido demuestra una mayor robustez en la navegación autónoma, reduciendo considerablemente la cantidad de episodios fallidos que se observan en el DRL puro (47%). Aunque el algoritmo DWA mostró los tiempos de planificación más bajos, también presentó limitaciones en la adaptabilidad a entornos más complejos. En contraste, el enfoque híbrido logró un equilibrio entre eficiencia y adaptabilidad, manteniendo tiempos de planificación aceptables y evitando el estancamiento observado en el DRL.

Si bien nuestros resultados demuestran la eficacia del enfoque híbrido en entornos simulados, es importante reconocer que existen ciertas limitaciones. Por ejemplo, no hemos considerado la influencia de factores como el ruido en los sensores o las imperfecciones en el control del robot, como siguiente paso, se considera explorar cómo estos factores afectan el rendimiento del sistema y cómo se podrían mitigar sus efectos para la validación del mismo en un robot físico, lo que permitirá evaluar su desempeño en un entorno real y considerar las limitaciones y desafíos que surgen al interactuar con el mundo físico. Esto abrirá la puerta a futuras investigaciones enfocadas en la robustez del sistema ante imprevistos y la adaptación a las particularidades del robot y su entorno.

ACKNOWLEDGEMENTS

Los autores agradecen al Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCyT) de México por el apoyo financiero brindado durante la realización de esta investigación.

REFERENCIAS

An, Z., Weixiang, W., Wenhao, B., Zhanjun, B. (2024). A path planing method based on deep reinforcement learning for AUV in complex marine environment. *Ocean Engineering*, 313, 119354. https://doi.org/10.1016/j.oceaneng.2024.119354

Bodong, T. and Jae-Hoon, K. (2024). Deep reinforcement learning-based local path planning in dynamic environments for mobile robot. *Journal of King Saud University - Computer and Information Sciences*, 36, https://doi.org/10.1016/j.jksuci.2024.102254

Ebrahim, S., Said, A., Hitham, A., Safwan, A., Alawi, A., Mohammed, R., Suliman, F. (2024). Deep deterministic policy gradient algorithm: A systematic review. *Heliyon*, 10. https://doi.org/10.1016/j.heliyon.2024.e30697

Fanfan, S., Bofan, Y., Jun, Z., Chao, X., Yong, C., Yanxiag, H. (2024). TD3-based trajectory optimization for energy consumption minimization in UAV-assisted MEC system. *Computer Networks*, 255, 110882. https://doi.org/10.1016/j.comnet.2024.110882

Haisen, G., Zhigang, R., Jialun, L., Zongze, W., Shengli, X. (2023). Optimal navigation for AGVs: A soft actor -critic based reinforcement learning approach with composite auxiliary rewards. *Engineering Applications of Artificial Intelligence*, 124, 106613. https://doi.org/10.1016/j.engappai.2023.106613

Husman, A., Oscar, A. (2024). Optimized TD3 algorithm for robust autonomous navigation in crowded and dynamic human-interaction environments. *Results in Engineering*, 24, 102874. https://doi.org/10.1016/j.rineng.2024.102874

Jinding, Z., Kai, Z., Zhongzheng, W., Wensheng, Z., Chen, L., Liming, Z., Xiaopeng, M., Piyang, L., Ziwei, B., Jinzheng, K., Yongfei, Y., Jun, Y. (2024). A latent space method with maximum entropy deep reinforcement learning for data assimilation. *Geoenergy Science and Engineering,* 243. https://doi.org/10.1016/j.geoen.2024.213275

Laiyi, Y., Jing, B., Haitao, Y. (2022). Dynamic path planning for mobile robots with deep reinforcement learning. *IFAC PapersOnLine*, 55-11. 10.1016/j.ifacol.2022.08.042

Liang, G., Te, S., Xudong, L., Ke, L., Natalia, D., David, F., Zhengfeng, Z., Junping, Z. (2020). Demostration Guided Acctor-Critic Deep Reinforcement Learning for Fast Teaching of Robots in Dynamic Environments. *IFAC PapersOnLine*, 53-5, 271-278. 10.1016/j.ifacol.2021.04.227

Shuhuan, W., Yili, S., Ahmad, R., Zeteng W., Zhengzheng, G., Simeng, G. (2025). A deep residual reinforcement learning algorithm based on Soft Actor-Critic for autonomous navigation. *Expert Systems Whit Applications*, 259. https://doi.org/10.1016/j.eswa.2024.125238

Xiaoyu, G., Jiayu, Y., Shuai, L., Hengwei, L. (2022). Actor-critic with familiarity-based trajectory experience replay. *Information Sciences*, 582, 633-647. https://doi.org/10.1016/j.ins.2021.10.031

Yanjie, C., Norzalilah, N. (2024). An improved dynamic window approach algorithm for dynamic obstacle avoidance in mobile robot formation. *Decision Analytics Journal,* 11. https://doi.org/10.1016/j.dajour.2024.100471

 https://doi.org/10.58571/CNCA.AMCA.2025.038