CEA Comité Español de Automática
XLVII JORNADAS DE AUTOMÁTICA CÓRDOBA 2026
UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Análisis del tiempo de paso en el aprendizaje por refuerzo para robots móviles

Adrián Bañuls-Arias[a,*], Juan-Antonio Fernández-Madrigal[a], Ana Cruz-Martín[a], Vicente Arévalo-Espejo[a], Cipriano Galindo[a], Juan-Manuel Gandarias[a]

[a] *UnCORE Team, Instituto Universitario de Investigación en Ingeniería Mecatrónica y Sistemas Ciberfísicos (IMECH.UMA). Dpto. de Ingeniería de Sistemas y Automática, Universidad de Málaga, Arquitecto Francisco Peñalosa, nº 6, 29071, Málaga, España.*

To cite this article: Bañuls-Arias, A., Fernández-Madrigal, J.A., Cruz-Martín, A., Arévalo-Espejo, V., Galindo, C., Gandarias, J.M. 2026. Analysis of the Timestep in Reinforcement Learning for Mobile Robots. Jornadas de Automática, 47. https://doi.org/10.17979/ja-cea.2026.47.13780

Resumen

El uso del Aprendizaje por Refuerzo Profundo (DRL) en robótica ha experimentado un gran crecimiento en la última decada, pero el impacto de sus aspectos temporales al aplicarse a sistemas físicos –principalmente la duración del tiempo de paso (*timestep*)– sigue poco explorado. Aunque algunos enfoques lo incluyen en el espacio de acción, ofrecen un análisis limitado, omitiendo elementos clave como la asincronía del software o las latencias del sistema. Estos factores causan desviaciones entre los tiempos de paso nominales y reales, perjudicando el aprendizaje. Este artículo propone tareas de navegación para robots móviles diseñadas para analizar los efectos de la duración del tiempo de paso, respaldadas por más de 2000 horas de simulación. Hasta donde sabemos, esta es la primera evaluación multimétrica de la influencia del tiempo de paso en robótica basada en DRL. Los resultados confirman que existen tiempos de paso óptimos que maximizan la eficiencia del aprendizaje y el éxito en la tarea.

Palabras clave: Aprendizaje por Refuerzo Profundo, Duración del Tiempo de Paso, Robots Móviles, Brecha de Simulación a Realidad, Arquitecturas Asíncronas, Eficiencia de Muestreo, Dinámica Temporal, Robótica Inteligente.

Analysis of the Timestep in Reinforcement Learning for Mobile Robots

Abstract

The use of Deep Reinforcement Learning (DRL) in robotics has gained prominence in the last decade, but the impact of its intrinsic temporal components of DRL when applied to physical systems —generally summarized as the timestep duration —, remains underexplored. While some approaches to this issue include timestep duration as part of action space, they offer a limited analysis, overlooking key elements such as software architecture asynchrony or system-induced latencies. These factors cause deviations between nominal and actual timesteps, negatively impacting learning. This article proposes a set of navigation tasks for mobile robots specifically designed to analyze the effects of timestep duration, supported by over 2000 hours of simulation. To our knowledge, this constitutes the first multimetric evaluation of the influence of step time duration on DRL-based robotic tasks. The findings confirm the existence of optimal step times that maximize learning efficiency and task completion.

Keywords: Deep Reinforcement Learning, Timestep Duration, Mobile Robots, Sim-to-Real Gap, Asynchronous Architectures, Sample Efficiency, Temporal Dynamics, Intelligent Robotics.

1. Introducción

A pesar de los notables progresos del Aprendizaje por Refuerzo Profundo (DRL) en el ámbito de la robótica Le et al. (2024), la mayor parte de la investigación sigue limitada a entornos simulados, con una transferencia limitada a robots reales. Un obstáculo importante es la dependencia de arquitecturas secuenciales y síncronas derivadas del formalismo de los procesos de decisión de Markov (MDP) Puterman (2014), donde el algoritmo de aprendizaje y el bucle de ejecución del

*Autor para correspondencia: adrianb_arias@uma.es

agente se acoplan en un único hilo de ejecución, ignorándose los retrasos estocásticos derivados de observación, actuación, comunicación y computación. Descartar esta dinámica temporal inherente a los sistemas físicos, que también se observa en simuladores físicamente realistas como *CoppeliaSim* Robotics (2024), hace que este tipo de implementaciones de RL resulten poco representativas de las condiciones reales, (i.e. exacerba la brecha entre simulación y realidad —*sim-to-real gap*— Salvato et al. (2021)), introduciendo variabilidad temporal en el bucle de control Ibarz et al. (2021) y aumentando la probabilidad de inestabilidad e ineficiencia en el muestreo.

Aunque trabajos previos han abordado algunos aspectos de este problema, tienden a centrarse en adaptaciones algorítmicas en lugar de abordar el comportamiento en tiempo real de los sistemas desplegados. Por ejemplo, algunas arquitecturas asíncronas introducen un período de toma de decisiones más flexible, pero pasan por alto los retardos del sistema, descuidan el impacto de la duración de la acción en el aprendizaje y requieren modificaciones en el algoritmo de RL subyacente Yuan and Mahmood (2022). Otros autores modelan el tiempo como un proceso continuo Chen and Wang (2022) o incluyen explícitamente la duración del tiempo de acción en el espacio de acciones Wang and Beltrame (2024); Zhou et al. (2024), pero no estudian cómo afecta la duración del tiempo de paso al aprendizaje bajo restricciones del mundo real. Una rara excepción es Mahmood et al. (2018), que evalúa la duración de la acción mediante una arquitectura desacoplada y pruebas de latencia de red; no obstante, su análisis se limita a un conjunto reducido de valores de tiempo de paso y se centra exclusivamente en las recompensas obtenidas.

En este artículo se implementa un entorno de simulación que proporciona un control temporal detallado al desacoplar los bucles de aprendizaje y de ejecución del agente o del simulador. Esto se logra a través de una biblioteca de software ligera y agnóstica al algoritmo de RL, `RL Spin Decoupler` UNCORE-Team (2025), que habilita la comunicación entre ambos bucles. A diferencia de otros trabajos previos, este enfoque no altera el algoritmo de RL, el espacio de acciones ni la estructura de la política, y delimita explícitamente las responsabilidades temporales, facilitando así la monitorización sistemática de los retardos de ejecución. Esta observabilidad resulta crucial para analizar la optimalidad de la política frente a la duración del tiempo de paso y para reducir las discrepancias conductuales al transferir modelos aprendidos entre plataformas, particularmente en escenarios *sim-to-real*. Sobre esta infraestructura se ha diseñado una tarea específica de navegación para robots móviles que expone los efectos de la duración del tiempo de paso en el aprendizaje y en la ejecución posterior de la política, lo que permite llevar a cabo evaluaciones empíricas exhaustivas para identificar las duraciones del tiempo de acción que producen un rendimiento más robusto.

Nuestro estudio se fundamenta en el formalismo estándar de un MDP en tiempo discreto y estado continuo, donde el agente busca aprender una política estocástica $\pi(a_t \mid s_t)$ que maximice la recompensa esperada. Se ha seleccionado el algoritmo *Soft Actor-Critic* (SAC) Haarnoja et al. (2019), utilizado ampliamente en escenarios robóticos Kuo et al. (2025); de Jesus et al. (2021), por su eficiencia en espacios de acción continuos y por su función objetivo regularizada por entropía, lo que asegura tanto exploración como convergencia estable. Nuestros experimentos revelan la existencia de un tiempo de paso óptimo dependiente de la tarea en términos de rendimiento de aprendizaje. Dicho óptimo también depende de propiedades a nivel de sistema, tales como los retardos de procesamiento y la frecuencia de interacción.

En resumen, las contribuciones de este artículo son:

1. Propuesta de un banco de pruebas simuladas con un despliegue desacoplado del RL y del control del agente.
2. Diseño de una tarea de navegación para robots móviles que expone los efectos de diferentes tiempos de paso.
3. Estudio sistemático mediante múltiples métricas para identificar la duración óptima del tiempo de paso.
4. Análisis experimental del impacto de las discrepancias entre tiempos de paso reales y nominales.

El resto del documento se organiza del siguiente modo: la sección 2 ofrece una descripción detallada de la configuración experimental utilizada para analizar los efectos de la duración del tiempo de paso; la sección 3 presenta los resultados obtenidos, incluyendo una comparación de varias métricas de rendimiento; finalmente, en la sección 4 destacamos las principales conclusiones y esbozamos futuras líneas de trabajo.

2. Configuración experimental para el análisis del RL con tratamiento explícito del tiempo

A continuación se describe la tarea de navegación de robots móviles diseñada específicamente para exponer los efectos del uso de diferentes tiempos de paso y la configuración utilizada para su simulación.

2.1. Diseño de la tarea de RL

La tarea diseñada está orientada a poner de manifiesto las diferencias de rendimiento en función de la frecuencia en la toma de decisiones. Involucra a un robot de tracción diferencial que navega hacia un objetivo circular definido con tres anillos concéntricos (i_1, i_2, i_3). Esta configuración del objetivo en forma de diana permitirá, como se verá más adelante, distinguir entre múltiples niveles de éxito dependiendo del tiempo de paso seleccionado.

El objetivo del robot es alcanzar el anillo más interno en el menor tiempo posible, evitando colisionar con cualquier obstáculo del entorno. Para ello, las observaciones –estado– (s_t) recopiladas por el robot comprenden la posición y orientación relativas respecto al objetivo, junto con la distancia mínima a los obstáculos detectados en cada uno de los cuatro sectores resultantes de la partición del campo de visión (FoV) de un sensor LiDAR. El espacio de acciones (a_t) es continuo y está constituido por las velocidades lineal y angular del robot, definidas por los rangos $v \in [0.1, 0.5]$ m/s y $\omega \in [-0.5, 0.5]$ rad/s, respectivamente.

Un episodio finaliza cuando el robot entra en *cualquiera* de los anillos del objetivo, colisiona con un obstáculo, excede la duración máxima del episodio o se desplaza más allá de la distancia permitida respecto al objetivo. Nótese que el robot no puede continuar avanzando hacia los anillos interiores una vez que alcanza uno externo, lo que limita de forma efectiva la recompensa máxima alcanzable en función de la duración

del tiempo de paso seleccionado. Esta configuración define implícitamente tres tareas diferentes, basadas en el anillo que el robot puede alcanzar físicamente.

La definición de recompensa (R_i) es de tipo disperso (*sparse*) e independiente del número de pasos por episodio, lo que asegura comparaciones equitativas. Éstas se basan en umbrales de distancia d_i dentro del anillo i-ésimo del objetivo (siendo d_t la distancia actual al centro de la diana). Asimismo, se aplican penalizaciones por colisión, episodios prolongados ($T_{\max} = 80$ s) y por exceder la distancia máxima al objetivo ($D_{\max} = 3$ m). En conjunto, la función de recompensa se define como:

$$R_i = \begin{cases} -1, & \text{si se detecta una colisión} \\ -0.5, & \text{si } t > T_{\max} \text{ o } d_t > D_{\text{máx}} \\ r_i, & \text{si } d_t < d_i \text{ para algún } i \in \{1,2,3\} \\ 0, & \text{en otro caso} \end{cases} \tag{1}$$

donde los umbrales de distancia y las posibles recompensas máximas para cada anillo del objetivo son: exterior (azul, i_1) con $d_1 = 0.25$ m y $r_1 = 0.25$; medio (rojo, i_2) con $d_2 = 0.125$ m y $r_2 = 0.5$; e interior (amarillo, i_3) con $d_3 = 0.015$ m y $r_3 = 1.0$.

La recompensa efectiva al alcanzar el anillo i-ésimo se calcula mediante la siguiente regla:

$$R(s_t, a_t) = R_i - \frac{R_i}{2} \cdot \min\left(\frac{t}{T_{\max}}, 1\right). \tag{2}$$

Este ajuste penaliza la navegación prolongada al reducir linealmente la recompensa r_i hasta un valor máximo igual a la mitad de la recompensa R_i correspondiente al anillo de parada si el robot emplea $T_{\max}$ segundos en alcanzar el objetivo. Este diseño incentiva al agente no solo a intentar acceder al anillo más interno posible, sino también a hacerlo de forma eficiente, con el objetivo de evidenciar los efectos del tiempo en los resultados.

2.2. *Plataforma experimental de simulación*

Para llevar a cabo un análisis pormenorizado de los efectos del tiempo, se ha desplegado una plataforma de simulación basada en *CoppeliaSim* Robotics (2024), un simulador de física realista ampliamente utilizado por la comunidad. Por un lado, el agente comprende la escena de *CoppeliaSim*, que incluye un robot *TurtleBot2* Lee (2013), un objetivo tipo diana y diez obstáculos cilíndricos; por otro, el algoritmo de RL se ha implementado en Python utilizando un entorno personalizado basado en *Gymnasium* Towers et al. (2024). Ambas partes funcionan en sendos bucles software y se comunican a través de la biblioteca `RL Spin Decoupler` UNCORE-Team (2025). El motor de física del simulador se ejecuta con un período de 0.005 s, y el bucle de ejecución del robot itera cada 0.05 s. El aprendizaje se ha implementado con *Stable Baselines3* Raffin et al. (2021), y las métricas se han registrado mediante *TensorBoard* Abadi et al. (2015).

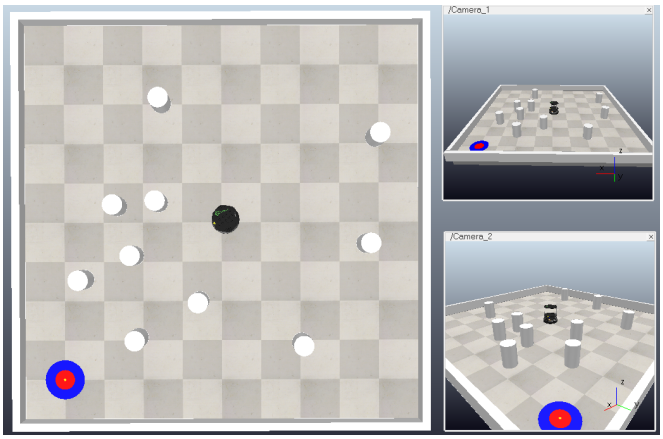

Figura 1: Entorno de *CoppeliaSim* utilizado, que incluye: un robot *TurtleBot2*, un objetivo tipo diana con tres anillos (azul, rojo, amarillo) y diez obstáculos en forma de columnas.

El robot navega en un escenario cerrado de 5×5 m (Fig. 1) ayudado de un LiDAR con un FoV de 270° situado en su parte frontal y configurado con umbrales de colisión mínimos adaptados a su geometría. En cada reinicio de episodio se aleatorizan las posiciones de los obstáculos y del objetivo, así como la orientación del robot, todo ello para asegurar la variabilidad a la vez que se mantiene una distancia mínima entre el robot y el resto de elementos de la escena para evitar inicializaciones no válidas. Si bien no se impone una distancia mínima al objetivo durante la fase de entrenamiento —una distancia inicial próxima al objetivo puede favorecer la convergencia—, estos episodios se descartan en la fase de explotación con el fin de garantizar un análisis más robusto de los resultados.

Se han llevado a cabo más de 30 experimentos, cada uno entrenado con un tiempo de paso fijo diferente, en un rango que va desde 0.2 hasta 7 segundos. En el intervalo $[0.2, 0.75]$ s, donde las dinámicas de aprendizaje son más sensibles y la convergencia más lenta, se ha muestreado el tiempo de paso cada 0.05 s y se ha entrenado cada modelo durante 600.000 pasos; el intervalo $[0.75, 5]$ s se ha muestreado de forma más gruesa, cada 0.25 s y entrenado durante 200.000 pasos; y, finalmente, el intervalo $[5, 7]$ s, ideado para encontrar condiciones más extremas, se ha explorado cada 1 s, utilizando igualmente 200.000 pasos de entrenamiento por modelo.

Dado que la aleatorización en el paso inicial de cada episodio es razonablemente simétrica y se espera que el agente explore una amplia variedad de situaciones, las tareas pueden considerarse ergódicas. En consecuencia, cada experimento se ha entrenado una sola vez, asumiendo que el promedio móvil de un solo experimento es representativo del promedio global de múltiples ejecuciones independientes.

Durante los experimentos, el algoritmo de RL y la simulación del agente compartieron ordenador. Para acelerar la ejecución, los experimentos se distribuyeron en diversas estaciones de trabajo, destacando la contribución de la estación de alto rendimiento del Instituto de Ingeniería Mecatrónica y Sistemas Ciberfísicos IMECH.UMA (2025) de la Universidad de Málaga. Esta estación está equipada con 4 GPUs NVIDIA RTX A6000 (48GB) y un procesador Intel Xeon Gold 5317 (48 hilos).

Por último, mencionar que, para asegurar comparaciones equitativas entre experimentos, los hiperparámetros del algo-

ritmo SAC se mantuvieron constantes, siguiendo la configuración predeterminada en *Stable Baselines3*: tasa de aprendizaje (*learning rate*) de 0.0003, tamaño de lote (*batch size*) de 256, capacidad de la memoria de repetición (*replay buffer*) de 1.000.000 de transiciones y factor de descuento $\gamma = 0.99$.

3. Evaluación y discusión

En esta sección se definen las métricas utilizadas en la evaluación, se presentan los resultados relativos a los efectos de la duración del tiempo de paso nominal, y finalmente se lleva a cabo un análisis temporal de las tareas correspondientes a los diferentes anillos del objetivo.

3.1. *Métricas utilizadas en la evaluación*

Las métricas utilizadas en este trabajo buscan proporcionar una perspectiva integral de los compromisos (*trade-offs*) asociados a la duración del tiempo de paso. Los criterios utilizados son los siguientes: a) la **duración media** del episodio; b) la **recompensa media** por episodio; c) el **número de colisiones**, utilizado como indicador de seguridad; d) el **punto de convergencia** en la curva de recompensa obtenida durante el aprendizaje, que mide la rapidez con la que se alcanza una política estable; e) el **tiempo de la última acción (LAT)**, que mide el retardo real entre la ejecución de una acción y la siguiente; y, por último, f) las **velocidades del robot**, tanto lineal como angular, utilizadas como indicadores de la fluidez y eficiencia al ejecutar la tarea. Nótese que intervalos de tiempo y/o retardos se miden con el reloj de tiempo real (*wall time*).

3.2. *Análisis de las duraciones nominal y real del tiempo de paso*

Para examinar equitativamente los efectos del tiempo, se distinguen las tres tareas descritas en la sección 2.1, definidas por el anillo del objetivo alcanzable dado un tiempo de paso determinado. Esto revela uno de los efectos más sutiles estudiados en este trabajo: la discrepancia entre la duración nominal (prevista por el RL) y la real del tiempo de paso (LAT). Dependiendo del tiempo de paso, si este LAT es significativamente mayor que el nominal, el agente puede ser capaz de acceder a anillos interiores que, en teoría, no serían alcanzables con ese tiempo de paso. La Fig.2 ilustra este fenómeno para un tiempo de paso de 0.2 s, destacando la relevancia de la discrepancia entre el LAT y el valor nominal, así como su impacto estocástico en la tarea.

La Fig.3 muestra un histograma de todos los tiempos de paso evaluados, donde se presenta la probabilidad de alcanzar cada anillo del objetivo. De estas probabilidades se deduce que: a) las duraciones en el intervalo $[0.2, 0.3]$ s conducen mayoritariamente al anillo exterior, en el intervalo $[0.35, 0.6]$ s al anillo medio, y en el intervalo $[0.65, 7]$ s al anillo más interno, respectivamente; y b) las diferencias entre el LAT y el tiempo de paso nominal introducen incertidumbre en la tarea. En la sección 3.3 se utiliza la categorización observada para desglosar los resultados por tarea con el fin de garantizar una comparativa lo más justa posible.

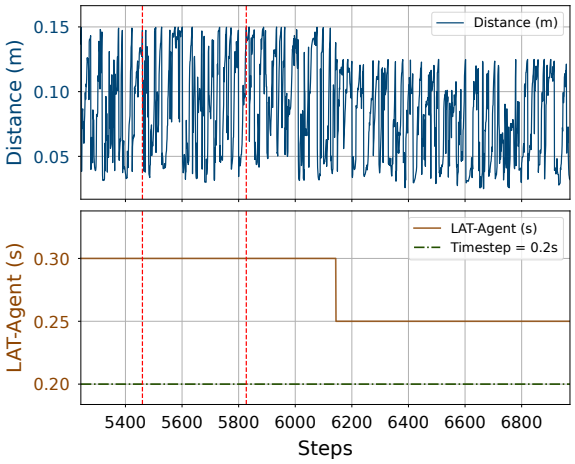

Figura 2: LAT y distancia recorrida por paso (modelo entrenado con tiempo de paso de 0.2 s). Las líneas discontinuas rojas marcan los episodios donde el agente alcanza un anillo que, con el tiempo de paso nominal, sería inalcanzable.

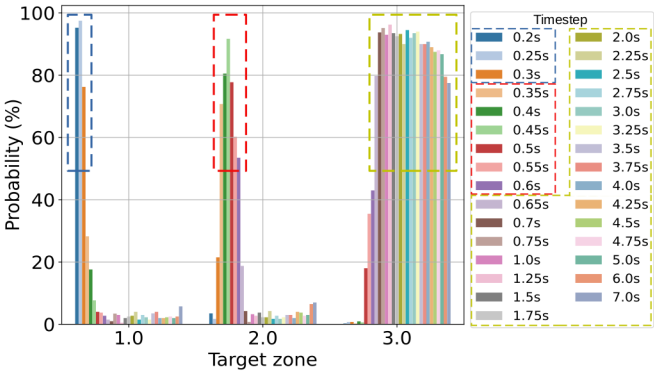

Figura 3: Histograma de accesibilidad a los anillos del objetivo por tiempo de paso. Emergen tres grupos de límites difusos, marcados en azul, rojo y amarillo, que coinciden con las tres tareas definidas (probabilidad > 50 %).

Más allá de establecer los límites (difusos) de las tareas en nuestro estudio, el LAT resulta crucial para identificar las latencias del sistema y evaluar la viabilidad de transferir políticas entrenadas en simulación a robots reales. En este contexto, se concluye lo siguiente:

- En sistemas que no son de tiempo real estricto (*hard real time*), las discrepancias entre los tiempos de paso nominales y reales son inevitables y estocásticas, lo que afecta al entrenamiento (convergencia, etc.).
- Dado que estas discrepancias suelen diferir entre el entrenamiento fuera de línea y el despliegue en un robot real, también afectan a la brecha entre simulación y realidad (*sim-to-real gap*) de manera impredecible.

3.3. *Análisis de los efectos de la duración del tiempo de paso nominal*

Finalizados los entrenamientos, se han explotado los modelos obtenidos, en condiciones idénticas, a lo largo de 400 episodios. Como se resume en la Tabla 1, para la tarea consistente en alcanzar el anillo más interno —solo alcanzable físicamente para tiempos de paso superiores a 0.6 s—los mejores

resultados se obtienen para duraciones en el rango [0.7, 1.5] s, con un pico en la recompensa media de 0.92 a los 0.75 s y tasas de colisión inferiores al 1 %, lo que refleja un control preciso y una navegación eficiente. Por el contrario, el rendimiento disminuye significativamente para tiempos de paso más largos: el modelo de 6 s produce una recompensa de 0.63 con 11.25 % de colisiones; para las demás tareas, las mejores recompensas tienden a obtenerse con la duración de tiempo de paso más larga. En términos generales, los tiempos de paso más cortos permiten completar la tarea con mayor rapidez debido a una retroalimentación más frecuente y una modulación de velocidad más segura, mientras que los tiempos de paso más largos aumentan generalmente el riesgo de colisión, a pesar de que el agente ralentiza la velocidad en un intento de compensación. Nótese también que existe al menos una duración óptima por cada tarea.

Tabla 1: Resultados experimentales de navegación con tiempos de paso representativos, agrupados en las tres distintas tareas.

Tiempo de paso (s)	Recompensa media	Tiempo medio (s)	Colisiones (%)
0.20	0.233	**7.023**	1.026
0.25	0.240	8.070	**0.000**
0.30	**0.269**	8.406	1.031
0.35	0.394	9.247	0.777
0.40	0.421	8.435	1.047
0.45	0.459	8.493	**0.000**
0.50	0.549	**8.203**	**0.000**
0.55	0.628	9.173	0.521
0.60	**0.660**	10.270	0.779
0.65	0.853	**9.025**	**0.000**
0.70	0.903	9.297	0.765
0.75	**0.922**	9.517	0.513
1.00	0.902	10.345	0.779
2.00	0.873	12.245	1.575
⋮	⋮	⋮	⋮
6.00	0.627	16.325	11.250
7.00	0.645	16.121	9.750

Desde una perspectiva visual, la Fig.4 ilustra la evolución de la recompensa media durante el entrenamiento. Los modelos entrenados con tiempos de paso de 0.2–0.3s se comportan de manera similar, con ligeras mejoras en el progreso del aprendizaje a medida que aumenta el tiempo de paso. Las duraciones de 0.35–0.6s muestran una tendencia ascendente más marcada, así como algunos saltos tardíos cuando el agente finalmente explora combinaciones de velocidad y posición que, con un LAT suficientemente alto (como se analiza en la Sec. 3.2), permiten el acceso a los anillos interiores. Para el rango de 0.65–7s, el agente alcanza el anillo más interno de forma fiable, incrementándose la recompensa de manera constante hasta alcanzar su máximo con un tiempo de paso situado entre 0.75–1.25s, el óptimo aparente para esta tarea. Más allá de este rango, el rendimiento se deteriora, debido principalmente a: (a) un aumento de las colisiones motivado por prolongados periodos de navegación "a ciegas" y (b) un movimiento más lento para compensar la reducción en la frecuencia de observación.

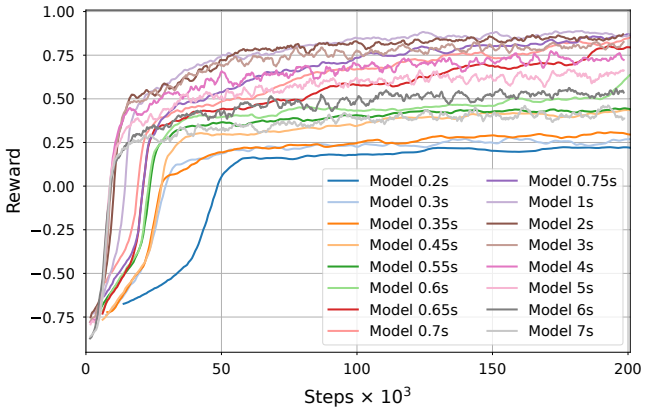

Figura 4: Comparativa de recompensas del entrenamiento para las duraciones de tiempo de paso representativas en todo el rango estudiado.

En la gráfica de la Fig.5 cada punto representa el número de pasos de entrenamiento requeridos para converger para un tiempo de paso determinado. La tendencia general revela una disminución en el número de pasos a medida que aumenta el tiempo de paso, observándose el menor esfuerzo de convergencia en las duraciones más largas, lo que permite al robot alcanzar el objetivo con un menor esfuerzo en la toma de decisiones. Nótese que los peores casos (0.5 s y 0.75 s) se encuentran en la región de transición entre la segunda y la tercera tarea, lo que explica su inestable comportamiento de convergencia. Esto resalta asimismo la importancia de la discrepancia en la duración del LAT y el valor nominal.

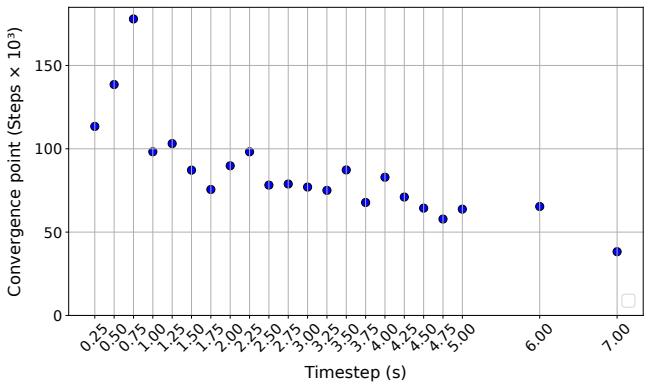

Figura 5: Número de pasos de entrenamiento requeridos para converger para un subconjunto representativo de duraciones del tiempo de paso.

Más allá de la convergencia, también se han examinado las velocidades angulares del agente durante la explotación. Los histogramas mostrados en la Fig. 6 reflejan la simetría de la aleatorización de las tareas (ver Sec. 2.2) y, por consiguiente, su ergodicidad: el robot encuentra un número aproximadamente igual de situaciones en las que debe girar a la derecha como a la izquierda en todos los tiempos de paso.

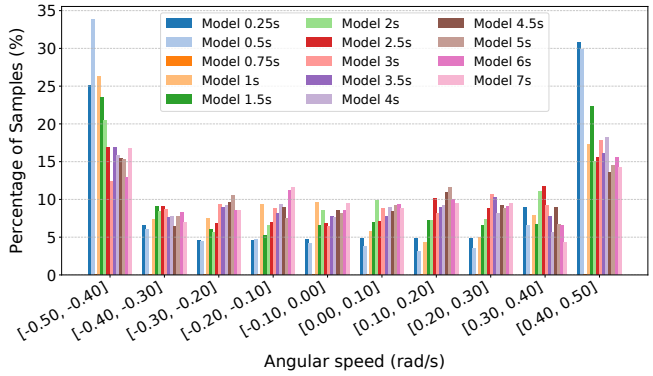

Figura 6: Histogramas de velocidad angular del agente bajo duraciones de tiempo de paso seleccionadas.

4. Conclusiones y trabajo futuro

Este trabajo ha explorado experimentalmente el impacto de la duración del tiempo de paso en el aprendizaje por refuerzo profundo (DRL) aplicado a tareas robóticas de navegación autónoma. Usando una configuración desacoplada e independiente del algoritmo que permite la implementación asíncrona del agente y del algoritmo de aprendizaje, hemos propuesto un marco de evaluación realista y detallado.

Con esta configuración se han realizado más de 2000 horas de entrenamiento en simulación para recopilar diversas métricas de rendimiento. Los análisis demuestran la existencia de valores óptimos de tiempo de paso para la tarea que maximizan tanto la eficiencia del aprendizaje como el rendimiento del control. Sin embargo, encontrar esos óptimos antes de diseñar la tarea resulta difícil, dado que también dependen de las características de tiempo real de la plataforma de computación.

Hasta donde alcanza nuestro conocimiento, éste es el primer estudio sistemático que cuantifica el impacto de la duración del tiempo de paso en las dinámicas de entrenamiento y el éxito de la tarea en DRL. Además, aporta nuevas perspectivas sobre la discrepancia existente entre las duraciones nominal y real del tiempo de paso, demostrando su impacto estocástico en el aprendizaje, aspecto que, en gran medida, se había pasado por alto en trabajos previos.

Basándose en estas contribuciones, surgen varias direcciones de trabajo futuro. Un desafío clave consiste en lograr que los agentes aprendan el tiempo de paso óptimo como parte del propio proceso de entrenamiento. Otra línea prometedora es el desarrollo de modelos predictivos para anticipar las latencias del sistema, ajustando la ejecución para mejorar la robustez y la precisión temporal, y aprovechando los retrasos no utilizados y la discrepancia entre la duración nominal y real del tiempo de paso para un mejor aprendizaje. Finalmente, la metodología agnóstica presentada aquí es suficientemente general para extenderse a otros dominios robóticos y algoritmos, permitiendo una generalización más amplia y una comprensión más profunda de las dinámicas temporales en DRL.

Agradecimientos

Este trabajo se ha desarrollado en el contexto del proyecto de investigación TYRELL (PID2023-147392NB-I00) financiado por MICIUAEI10.13039501100011033 y por los Fondos Europeos de Desarrollo Regional de la UE (FEDER). También agradecemos el apoyo técnico de IMECH.UMA a través de PPRO-IUI-2023-02.

Referencias

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
URL: `https://www.tensorflow.org/`

Chen, X., Wang, J., 2022. Inhomogeneous deep q-network for time sensitive applications. Artificial Intelligence 312, 103757.

de Jesus, J. C., Kich, V. A., Kolling, A. H., Grando, R. B., de Souza Leite Cuadros, M. A., Gamarra, D. F. T., 2021. Soft actor-critic for navigation of mobile robots. Journal of Intelligent & Robotic Systems 102.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., Levine, S., 2019. Soft actor-critic algorithms and applications.

Ibarz, J., Tan, J., Finn, C., Kalakrishnan, M., Pastor, P., Levine, S., 2021. How to train your robot with deep reinforcement learning: lessons we have learned. The International Journal of Robotics Research 40, 698 – 721.

IMECH.UMA, 2025. University Research Institute in Mechatronics Engineering and Cyber-Physical Systems, University of Málaga, Spain. `https://www.imech-uma.es/`, accessed on July 14, 2025.

Kuo, P.-H., Huang, C.-T., Chang, C.-W., Feng, P.-H., Lin, Y.-S., 2025. Design and implementation of a soft actor–critic controller for a robotic arm. Engineering Applications of Artificial Intelligence 151, 110589.

Le, H., Saeedvand, S., Hsu, C.-C., 11 2024. A comprehensive review of mobile robot navigation using deep reinforcement learning algorithms in crowded environments. Journal of Intelligent & Robotic Systems 110 (4), 158.

Lee, J., 2013. Turtlebot 2: The new standard hardware reference platform. In: ROSCon 2013. Open Robotics.
URL: `http://dx.doi.org/10.36288/roscon2013-899053`

Mahmood, A. R., Korenkevych, D., Komer, B., Bergstra, J., 2018. Setting up a reinforcement learning task with a real-world robot. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 4635–4640.

Puterman, M. L., 2014. Markov Decision Processes: Discrete Stochastic Dynamic Programming, 1st Edition. John Wiley & Sons.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N., 2021. Stable-baselines3: Reliable reinforcement learning implementations. Journal of Machine Learning Research 22 (268), 1–8.

Robotics, C., 2024. Coppeliasim. `https://www.coppeliarobotics.com/`, accessed: 2025-05-08.

Salvato, E., Fenu, G., Medvet, E., Pellegrino, F. A., 2021. Crossing the reality gap: a survey on sim-to-real transferability of robot controllers in reinforcement learning. IEEE Access PP, 1–1.

Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., et al., 2024. Gymnasium: A standard interface for reinforcement learning environments. arXiv preprint arXiv:2407.17032.

UNCORE-Team, 2025. rl_spin_decoupler: A library for decoupling agent and environment control loops in RL. `https://github.com/uncore-team/rl_spin_decoupler/tree/main`, accessed: 2025-05-13.

Wang, D., Beltrame, G., 2024. Variable time step reinforcement learning for robotic applications.

Yuan, Y., Mahmood, A. R., 2022. Asynchronous reinforcement learning for real-time control of physical robots.

Zhou, H., Huang, A., Azizzadenesheli, K., Childers, D., Lipton, Z., 02–04 May 2024. Timing as an action: Learning when to observe and act. In: Dasgupta, S., Mandt, S., Li, Y. (Eds.), Proceedings of The 27th International Conference on Artificial Intelligence and Statistics. Vol. 238 of Proceedings of Machine Learning Research. PMLR, pp. 3979–3987.