CEA Comité Español de Automática

XLV JORNADAS AUTOMÁTICA MÁLAGA 2024

UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Aprendizaje por refuerzo en vivo para robots móviles: mejoras en autonomía y autoadaptación

Asensio-Huonder, S.[a,*], Arévalo-Espejo, V.[a], Cruz-Martín, A.[a], Fernández-Madrigal, J.A.[a]

[a] *Dpto. de Ingeniería de Sistemas y Automática, Universidad de Málaga, Arquitecto Francisco Peñalosa, nº 6, 29071, Málaga, España.*

To cite this article: Asensio-Huonder, S., Arévalo-Espejo, V., Cruz-Martín, A., Fernández-Madrigal, J.A. 2024. Live reinforcement learning for mobile robots: improvements in autonomy and self-adaptation. Jornadas de Automática, 45. https://doi.org/10.17979/ja-cea.2024.45.10898

Resumen

En este trabajo se propone una arquitectura de reflejos ante estímulos sensoriales para que el aprendizaje por refuerzo en vivo para robots móviles mejore su adaptación a cambios en la tarea, aumente su autonomía para regresar a estados seguros tras errores y reduzca, en general, la supervisión por parte del humano. El trabajo se ha enfocado en la navegación de un robot móvil con evitación de obstáculos y hemos utilizado una versión modificada de los algoritmos de aprendizaje por refuerzo *Q-learning* y *True On-Line SARSA*(λ). Se ha estudiado adicionalmente un aprendizaje que traslada lo aprendido en simulación al aprendizaje en vivo, llamado aprendizaje híbrido. Los resultados muestran que nuestra arquitectura mejora la seguridad del robot y su adaptabilidad a cambios en la tarea, minimiza la intervención humana y extiende el tiempo de entrenamiento sin supervisión.

Palabras clave: Aprendizaje por refuerzo, Robótica y Mecatrónica, Robots móviles, Simulación.

Live reinforcement learning for mobile robots: improvements in autonomy and self-adaptation

Abstract

In this study, we propose an architecture based on automatic reflexes in response to sensory stimuli for live reinforcement learning to achieve better task adaptation, increased autonomy for returning to safe states after errors, and reduced learning supervision. A mobile robot has been used in a navigation task while avoiding obstacles, using modified versions of the reinforcement learning algorithms *Q-learning* and *True On-Line SARSA*(λ). Additionally, we explore a hybrid learning approach, which involves transferring insights gained from simulations to live environments. The results show that our arquitecture with reflexes improves the robot safety and adaptability to task changes, minimizing human intervention and extending unsupervised training time.

Keywords: Reinforcement Learning, Robotics and Mechatronics, Mobile robots, Simulation.

1. Introducción

El aprendizaje por refuerzo (*Reinforcement Learning* o RL (Sutton and Barto, 2005)) es una de las áreas del *Machine Learning* más en auge actualmente. Este paradigma permite que un agente computacional aprenda en base a su propia experiencia las mejores políticas de acción para operar eficientemente en un entorno determinado. Con una configuración paramétrica reducida, el RL permite encontrar soluciones a tareas secuenciales complejas en situaciones de incertidumbre.

Desafortunadamente, cuando el RL se implanta en un robot, aparecen problemas concretos que no se dan en la aplicación del RL a otras disciplinas (Kober et al., 2013), como el coste del aprendizaje si se realiza en vivo y no en simulación. Aunque en los últimos años se han desarrollado numerosas aplicaciones de RL en Robótica, la mayoría se han enfocado en el aprendizaje en sí, es decir, en conseguir que el robot realice adecuadamente operaciones muy específicas

*Autor para correspondencia: asensioh.santiago@uma.es

y delimitadas como la manipulación (Hu et al., 2020; Parisi et al., 2015; Hwang et al., 2014; Zhang et al., 2022) o la navegación (Wen et al., 2021; Lin et al., 2018; Okal and Arras, 2016; Kawano, 2013); no se ha estudiado con la misma intensidad el aprendizaje en vivo o la adaptación del robot, tras haber aprendido, a variaciones en la tarea, en su entorno o en su propia estructura.

Este trabajo propone una aproximación novedosa para el RL en vivo en robots que consiste en la inclusión de un conjunto de reflejos, concebidos en semejanza a los reflejos innatos a los seres vivos. Los reflejos son una secuencia de acciones desencadenadas ante la recepción de ciertos estímulos sensoriales con las que se pretende guiar al robot en situaciones específicas, como la cercanía a obstáculos, manteniendo el aprendizaje del robot durante este proceso. De esta forma, se pretende disminuir la supervisión humana necesaria para completar aprendizajes por refuerzo en vivo, que son siempre muy costosos, sobre todo en tiempo, y han de ser supervisados de continuo.

El trabajo aquí expuesto se organiza como sigue. La sección 2 presenta los algoritmos de aprendizaje por refuerzo empleados en este trabajo y tres estrategias de selección de acción. En la sección 3 se detalla la tarea a aprender: se describen las acciones, el estado del agente robótico y su discretización, y la función de recompensa establecida. En la sección 4 se propone un método para la adaptación del aprendizaje por refuerzo para la ejecución en vivo mediante dos algoritmos RL modificados con la incorporación de reflejos. La sección 5 recoge los resultados de las experimentaciones en simulación y en vivo. Por último, en la sección 6 se incluyen las conclusiones extraídas y se proponen futuras líneas de trabajo.

2. Visión general de algoritmos RL

En esta sección se introducen los algoritmos de aprendizaje por refuerzo *Q-learning*, *SARSA* y una versión más avanzada del algoritmo *SARSA*: *True On-Line SARSA* (λ), en adelante *TOLS*. Además, se presentan tres estrategias de selección de acciones, como las ampliamente conocidas ϵ-*greedy* y *softmax*, junto con otra más sofisticada: la estrategia de selección de acción *Gumbel-Boltzmann* (Cesa-Bianchi et al., 2017). *TOLS* es el método que mejores resultados ha dado en el aprendizaje de diversas tareas robóticas (Martínez-Tenor et al., 2018); la estrategia *Gumbel* se implementa en este trabajo de forma novedosa para el aprendizaje de un robot móvil.

Un entrenamiento de RL consiste en el lanzamiento de múltiples episodios de entrenamiento, donde cada episodio agrupa un conjunto de pasos o iteraciones del algoritmo RL. En cada paso, el agente se halla en un estado s_k, escoge una acción a_k y la ejecuta, recibe una recompensa r_k, actualiza el valor del par estado-acción visitado y registra la información asociada al paso de ejecución actual (ver la figura 1). Durante el transcurso del aprendizaje, el agente aprende una política de toma de decisiones.

Un concepto fundamental en el contexto del RL es el modelo del entorno, que comprende todo aquello que puede ser empleado por el agente para predecir cómo responde dicho entorno a sus acciones (Sutton and Barto, 2005). Hay métodos de aprendizaje por refuerzo que pueden ser empleados sin la necesidad de modelar el entorno, conocidos como *model free*, en los que se ha enfocado este trabajo.

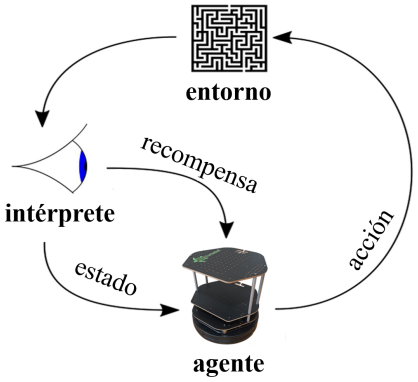

Figura 1: Diagrama de interacciones entre el agente RL y su entorno.

2.1. Algoritmos RL tabulares

Existen diversas implementaciones de algoritmos de RL; las descritas en este proyecto se han llevado a cabo tomando un enfoque tabular, esto es, los valores asociados a cada par estado-acción (s_k, a_k) son almacenados en una matriz de valores $Q_{S\times A}$, con S estados diferentes (filas) y A acciones disponibles (columnas) para el agente en cada estado. En cada paso del algoritmo se actualizan uno o varios elementos de la matriz Q en base al estado del agente y a la acción ejecutada durante la iteración. A continuación se describen brevemente tres algoritmos RL.

Q-learning es un algoritmo de aprendizaje por refuerzo *model-free* que permite a un agente aprender de forma iterativa su comportamiento mediante la estimación de la acción óptima en función del estado actual. La ecuación 1 muestra la actualización del valor asociado al par estado-acción (s_k, a_k) que realiza el algoritmo de *Q-learning* en el paso de ejecución k de un episodio.

$$Q_{s_k,a_k} = Q_{s_k,a_k} + \alpha \cdot \left[r_k + \gamma \cdot \max_a\{Q_{s'_k,a_k}\} - Q_{s_k,a_k} \right] \quad (1)$$

donde α es el ritmo de aprendizaje, Q_{s_k,a_k} es el valor asociado al par estado-acción (s_k, a_k), r_k es la recompensa recibida en la transición desde el estado s_k al estado s'_k y γ es el factor de descuento, entre 0 y 1. El término $\max_a\{Q_{s'_k,a_k}\}$ indica la estimación del valor asociado a la elección de la acción a_k en el estado s'_k que maximiza el valor de Q en el futuro. En el presente trabajo se ha empleado este algoritmo, entre otros, debido a su sencillez de implementación.

Otro algoritmo RL muy conocido es *SARSA*, el cual lleva a cabo la actualización del valor asociado al par estado-acción visitado por el agente en el paso de ejecución k haciendo uso de (2), a partir del par estado-acción actual y del siguiente par estado-acción elegido. Esto implica que es necesario conocer la siguiente acción que debe tomar el agente. En cada iteración se hace uso de los siguientes elementos: s_k, a_k, r_k, s'_k, a'_k, lo que da nombre al algoritmo.

$$Q_{s_k,a_k} = Q_{s_k,a_k} + \alpha \cdot \left[r_k + \gamma \cdot \max_a\{Q_{s'_k,a'_k}\} - Q_{s_k,a_k} \right] \quad (2)$$

El algoritmo *TOLS* aumenta el rendimiento del algoritmo *SARSA* mediante el empleo de trazas de elegibilidad de

tamaño limitado (Martínez-Tenor et al., 2018). Una traza de elegibilidad es una matriz encargada de almacenar, durante un corto periodo de tiempo, el valor asociado a los últimos pares de estado-acción visitados por el agente (Sutton and Barto, 2005). Estas trazas sirven para propagar lo que se observa en el estado actual hacia atrás, de modo que en un único paso se actualice una mayor porción de la matriz Q. Los valores almacenados en las trazas de elegibilidad decaen con el tiempo en función del parámetro de decaimiento de trazas, λ.

2.2. Estrategias de selección de acción

En este apartado se describen tres estrategias para la selección de las acciones del robot durante el aprendizaje RL.

Dado un estado concreto y un conjunto de acciones disponibles, la estrategia de selección de acción ϵ-*greedy* se basa en la elección de aquella acción que posea el mayor valor asociado en la mayoría de las ocasiones o, con una pequeña probabilidad ϵ, cualquier acción elegida al azar. En este trabajo se ha escogido esta estrategia para la selección de acciones en el algoritmo *Q-learning*.

La estrategia de selección de acción *softmax* garantiza que todas las acciones disponibles en cada estado posean una probabilidad no nula, de forma que existe una probabilidad de explorar el espacio de estados al mismo tiempo que se favorece la selección de acciones óptimas. El método de selección de acción *Gumbel-Boltzmann* modifica la estrategia *softmax* en el aprendizaje por refuerzo añadiendo incertidumbre a la selección de acciones para mejorar la adaptabilidad y la optimalidad de la política aprendida. En el presente trabajo se ha aplicado esta estrategia en el algoritmo *TOLS*.

3. Tarea de aprendizaje: navegación en entorno con obstáculos

La tarea de aprendizaje propuesta consiste en la navegación autónoma de un robot móvil en un entorno de interiores hasta una posición objetivo o *target*, mediante la ejecución secuencial del menor número posible de acciones y evitando las colisiones con obstáculos fijos. El robot real utilizado en los experimentos ha sido un *Turtlebot2* dotado de un escáner láser *Hokuyo URG-04LX-UG01* con un FOV de 240º.

3.1. Estado discreto del agente robótico

El estado de un robot recoge la información a partir de la cual se caracteriza cada posible interacción que puede darse entre el robot y su entorno. En el presente trabajo se ha realizado la discretización del espacio de estados y de acciones (tablas 1 y 2) para adaptarnos a la naturaleza tabular de los algoritmos utilizados.

Tabla 1: Discretización del espacio de estados.

Espacio	Min.	Máx.	Divisiones
Distancia *target*-robot (m)	0	10	6
Orientación *target*-robot (º)	0	360	7
Distancia a obstáculos (m)	0.3	5	5

Tabla 2: Discretización de las acciones.

Acción	Tiempo de acción (s)	Velocidad
Avance frontal	1	0.25 m/s
Giro antihorario	1	15 º/s
Giro horario	1	15 º/s

3.2. Recompensas

En cada paso del aprendizaje se otorga al robot una recompensa negativa (penalización) de -1. Con esto se pretende que el agente valore en mayor medida la elección de secuencias de acciones que le permitan alcanzar el *target* en un número de pasos lo más reducido posible y sin colisionar con obstáculos. Cuando el robot se halle a una distancia inferior a 0.2 m de cualquier obstáculo, se otorga una recompensa negativa de -100 para penalizar muy severamente las colisiones. En caso de alcanzar la posición del *target*, se otorga una recompensa positiva de valor +100.

4. Adaptación del aprendizaje a la ejecución en vivo y a cambios en la tarea

En este trabajo se propone un enfoque novedoso basado en reflejos que se desencadenan automáticamente ante la recepción de estímulos sensoriales, lo que agiliza el aprendizaje en vivo y permite la adaptación del agente a cambios en la tarea del aprendizaje. Habitualmente, el RL en Robótica se hace en simulación y posteriormente se traslada a la ejecución en vivo.

4.1. Integración de reflejos en el aprendizaje por refuerzo

Se han implementado tres reflejos específicos dentro del proceso de aprendizaje por refuerzo, que se activan bajo condiciones críticas:

- Reflejo de proximidad a obstáculo. Activado cuando un obstáculo se detecta a menos de 0.4 metros del robot, desencadena un algoritmo de guiado que emite acciones de giro hasta que el robot deja de estar orientado hacia el obstáculo; todas las acciones tomadas, aunque dictadas por el algoritmo de guiado, continúan alimentando el proceso de aprendizaje mediante la actualización de la matriz Q. Una vez que el robot regresa a una situación segura, el control sobre la selección de acciones se devuelve al algoritmo de RL.

- Reflejo de colisión con obstáculo. Se activa ante la detección de un obstáculo a menos de 0.2 metros, que se considera inevitable, y emite un aviso al supervisor humano.

- Reflejo de nivel bajo de batería. Se dispara cuando la batería cae por debajo del 10 %; envía un mensaje de aviso al supervisor, solicita la recarga y apaga el ordenador y el robot.

La figura 2 muestra el flujo de ejecución del algoritmo de RL con reflejos.

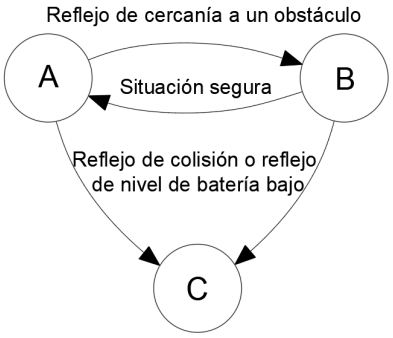

Figura 2: (A) Ejecución de política RL basada en la estrategia de selección de acción, (B) ejecución del algoritmo de guiado de acciones tras el disparo de un reflejo y (C) fin de episodio.

4.2. *Aprendizaje híbrido y adaptación a cambios en la tarea*

El aprendizaje híbrido (Ibarz et al., 2021) busca aunar las ventajas asociadas al entrenamiento en simulación con el entrenamiento en vivo. Consiste en el lanzamiento de una primera fase de entrenamiento en simulación, durante la cual el agente simulado lleva a cabo el aprendizaje inicial de la tarea en una réplica virtual del entorno real; la segunda fase consiste en el lanzamiento de episodios de aprendizaje del agente físico, el cual parte de la experiencia obtenida por el agente en simulación para proseguir con el entrenamiento en vivo.

Es habitual observar un comportamiento en el robot real que difiere del encontrado en el aprendizaje efectuado en simulación, conocido como *reality gap*; en el presente trabajo está causado, entre otros factores, por la presencia de una deriva direccional que afecta al desplazamiento en línea recta del robot empleado en los experimentos en vivo.

5. Resultados

En primer lugar, se analizan los resultados obtenidos en los aprendizajes en simulación y en vivo haciendo uso de los algoritmos con reflejos *Q-learning* y *TOLS*. Finalmente, se muestran los resultados del lanzamiento de un entrenamiento híbrido haciendo uso del algoritmo *TOLS* con reflejos.

En la figura 3 se muestran los entornos, el real y el simulado en *Gazebo*, utilizados en los experimentos, un pasillo de dimensiones 10.6×3.7 m.

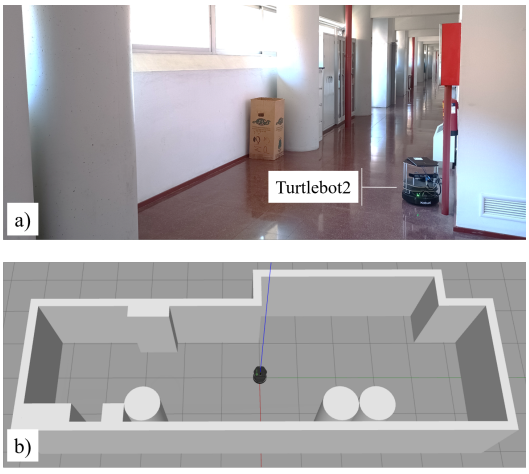

Figura 3: a) Entorno real de experimentación. b) Modelado en *Gazebo* de un entorno en simulación de dimensiones semejantes al entorno real.

Se han completado 20 entrenamientos en simulación de 2800 episodios cada uno haciendo uso del algoritmo *Q-learning* con reflejos y 8 entrenamientos de 3000 episodios empleando el algoritmo *TOLS* con reflejos, tomando un ritmo de aprendizaje de $\alpha = 0.1$ en ambos algoritmos. Las figuras 4 y 5 incluyen las gráficas de resultados de *Q-learning* y *TOLS* respectivamente.

Se ha representado la curva media de recompensa acumulada por episodio (R), la frecuencia media de llegada a *target* y de colisión con obstáculos tomando ventanas de 10 episodios, para ambos algoritmos; el porcentaje restante es la frecuencia de finalización de episodios debido a la superación del máximo número de pasos. En base a los resultados en simulación, se observa que la frecuencia media de colisiones con obstáculos al final del aprendizaje es menor en el caso del *TOLS*, lo que constituye una ventaja a la hora de elegir un algoritmo para llevar a cabo entrenamientos en vivo. *TOLS* proporciona una menor frecuencia media de llegada a *target* frente al *Q-learning*.

Se han completado asimismo entrenamientos adicionales de la misma longitud empleando ambos algoritmos RL sin reflejos, con el fin de demostrar la utilidad de los mismos; la tabla 3 resume los resultados finales medios de los entrenamientos en simulación con y sin reflejos. Tanto la recompensa acumulada como la frecuencia de llegada al objetivo son más elevadas en el caso de la incorporación de los reflejos en el aprendizaje y las colisiones son más infrecuentes.

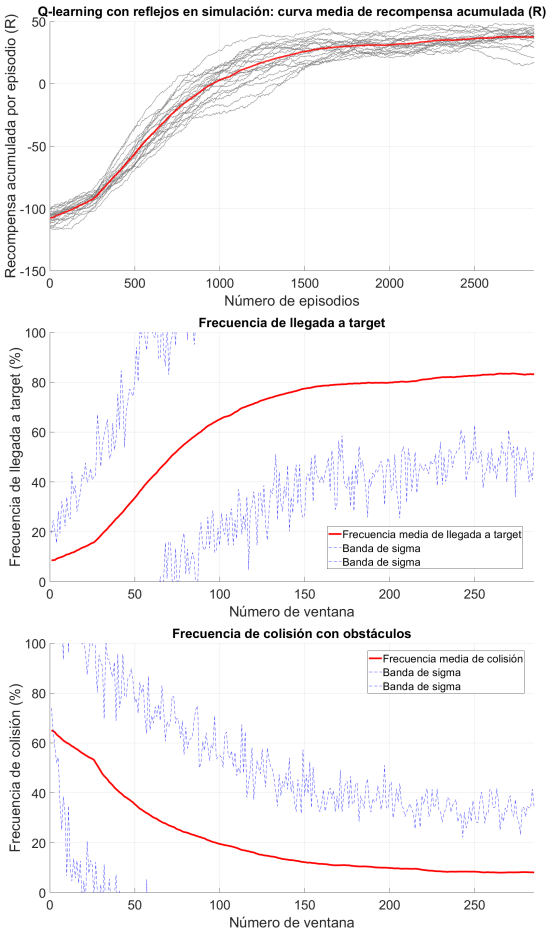

Figura 4: Resultados de los entrenamientos en simulación mediante el algoritmo *Q-learning* con reflejos.

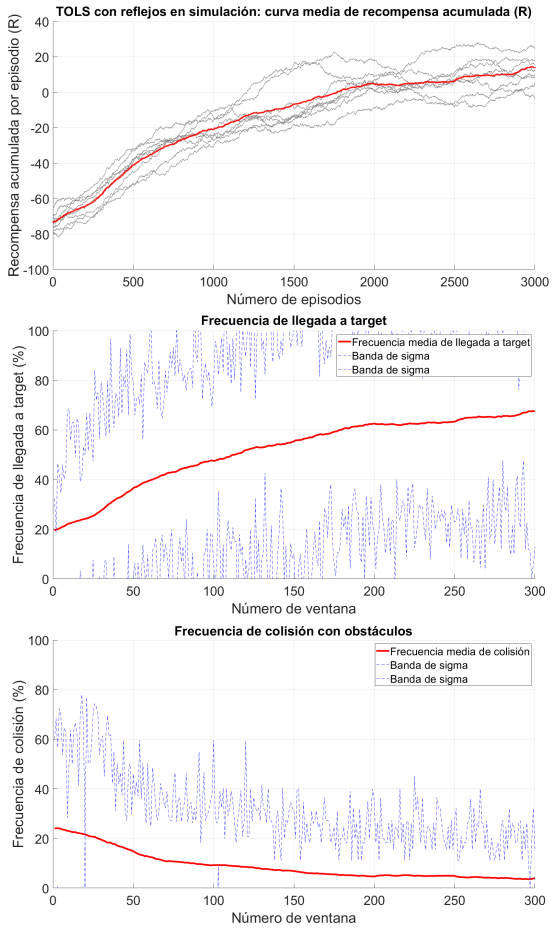

Figura 5: Resultados de los entrenamientos en simulación mediante el algoritmo *TOLS* con reflejos.

Tabla 3: Resultados de experimentación en simulación con y sin reflejos.

Algoritmo	R	Llegadas (%)	Colisiones (%)
QL con reflejos	32	80	8
TOLS con reflejos	10	66	4
QL sin reflejos	7	66	26
TOLS sin reflejos	-17	49	12

Para los experimentos reales, se ha efectuado un entrenamiento en vivo de 1700 episodios con el algoritmo *Q-learning* con reflejos y un entrenamiento en vivo de 1700 episodios empleando el algoritmo *TOLS* con reflejos bajo la suposición de la ergodicidad de la tarea de aprendizaje, es decir, asumiendo que se visitan todos los pares estado-acción relevantes de forma suficientemente frecuente con independencia del estado de partida.

En el lanzamiento del entrenamiento híbrido se ha optado de nuevo por confiar en la ergodicidad del problema, por lo que se ha efectuado un único entrenamiento haciendo uso del algoritmo *TOLS* con reflejos, debido a que los resultados en vivo muestran una menor frecuencia de colisión con obstáculos. Se ha completado un entrenamiento en simulación de 3200 episodios y se ha transferido el aprendizaje de la tarea en simulación al entrenamiento en vivo, consistente en inicializar la matriz Q del aprendizaje en vivo usando la Q aprendida en simulación. En la segunda fase del entrenamiento híbrido se han completado 350 episodios en vivo.

Se observa un descenso de la frecuencia de llegada al *target* en los episodios iniciales, debido a la existencia de un *reality gap* entre el entorno y robot simulados frente a la realidad; el robot real presenta una deriva direccional y el entorno simulado no tiene por qué coincidir exactamente con el real. Tras una fase transitoria de adaptación al entorno real, se logra sortear el *reality-gap* y el aprendizaje se estabiliza en torno a una frecuencia de llegada a *target* cercana al 88 %.

La tabla 4 resume los resultados finales de los entrenamientos en vivo y del híbrido; las figuras 6 y 7 incluyen las gráficas de resultados de *Q-learning* y *TOLS* en vivo y del entrenamiento híbrido, respectivamente. Se observan resultados similares a los obtenidos en simulación, tanto en la frecuencia de llegada a *target* como en la frecuencia de colisión con obstáculos, destacando el algoritmo *TOLS* por lograr una menor frecuencia de colisión con obstáculos y el algoritmo *Q-learning* por la elevada frecuencia de llegada a *target*. También se observa que en el aprendizaje híbrido se parte, de entrada, con un comportamiento ya muy definido.

Tabla 4: Resultados de la experimentación en vivo y del aprendizaje híbrido con reflejos.

Algoritmo	R	Llegadas (%)	Colisiones (%)
Q-learning	18	75	17
TOLS	7	65	5
TOLS híbrido	23	88	5

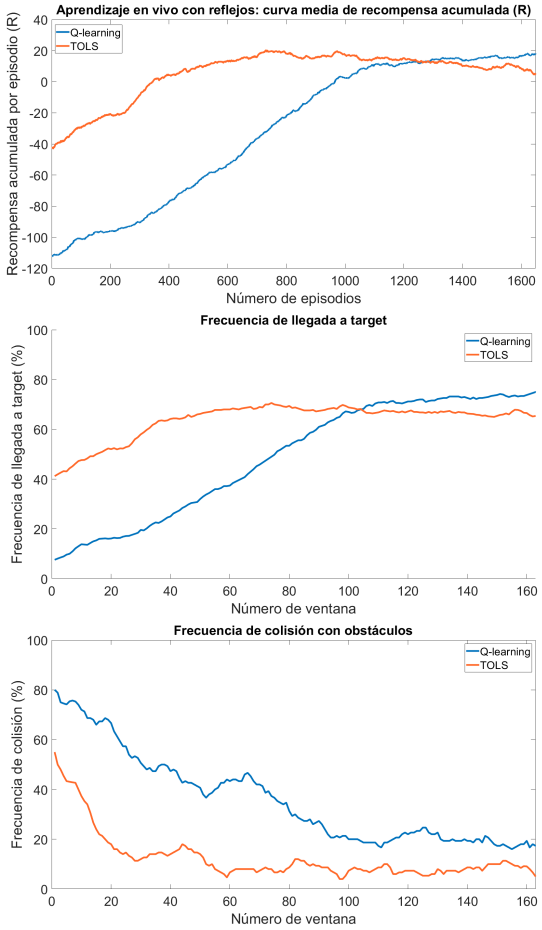

Figura 6: Comparación de los resultados del entrenamiento en vivo mediante los algoritmo *Q-learning* y *TOLS* con reflejos.

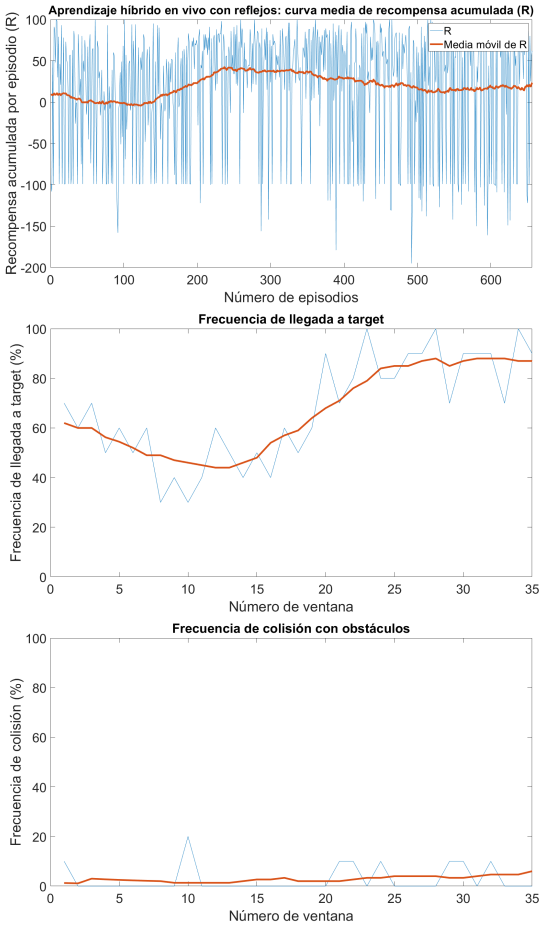

Figura 7: Resultados del entrenamiento híbrido mediante el algoritmo *TOLS* con reflejos.

6. Conclusiones

En el presente trabajo se ha propuesto una arquitectura basada en reflejos integrada en el aprendizaje RL para la exploración de mejoras en el aprendizaje en vivo en robots, destinada a incrementar su autonomía para retornar a estados seguros tras un resultado erróneo en el aprendizaje y su adaptación a cambios en la tarea de aprendizaje, lo que supone un enfoque original y con un gran potencial para reducir la supervisión de los entrenamientos. Para ello se ha planteado una aproximación al entrenamiento en vivo a través del lanzamiento previo de experimentos en simulación, logrando así obtener un conjunto de valores para la parametrización de los aprendizajes en vivo de forma rápida y eficiente y para el análisis del funcionamiento de la arquitectura de reflejos en simulación.

En la sección 5 se concluye que el algoritmo *TOLS* con reflejos logra un aprendizaje más seguro tanto en vivo como en simulación debido a la baja frecuencia de colisiones con obstáculos, aunque con valores de recompensa acumulada inferiores a los obtenidos mediante *Q-learning*.

Por último, se ha completado un entrenamiento híbrido empleando el algoritmo *TOLS*, que ha sido escogido debido a la baja tasa de colisiones con obstáculos. La política previamente aprendida en simulación y transferida al inicio del aprendizaje en vivo proporciona buenos resultados tanto en alcance de *target* como en evitación de obstáculos, a pesar de la existencia de un *reality-gap* que posee como principal causa una deriva direccional del robot real; el robot logra finalmente adaptarse a la existencia de diferencias entre las tareas de aprendizaje simulado y en vivo. Se concluye que este método constituye una opción veloz y segura para la compleción de aprendizajes en vivo mediante agentes robóticos, con un porcentaje muy reducido de colisiones desde el inicio de la segunda fase del entrenamiento híbrido, en la cual se opera ya con el robot físico. Se proponen como futuras líneas de trabajo el lanzamiento de un mayor número de entrenamientos en vivo, el desarrollo de un método de RL adaptativo para el ajuste del factor de aprendizaje (α) y la implementación del *reward shaping* (Wiewiora et al., 2003) para definir una función de recompensas continua basada en los estados y acciones del agente.

Agradecimientos

Este trabajo ha sido realizado parcialmente gracias al apoyo del Vicerrectorado de Investigación y Transferencia de la Universidad de Málaga, por la financiación de este trabajo a través de la Ayuda a la Iniciación a la Investigación (Modalidad A) del Plan Propio de Investigación de la Universidad de Málaga.

Referencias

Cesa-Bianchi, N., Gentile, C., Lugosi, G., Neu, G., 2017. Boltzmann exploration done right. In: 31st Conference on Neural Information Processing Systems (NIPS). Vol. 30. Long Beach, pp. 6284–6293.

Hu, Y., Wang, W., Liu, H., Liu, L., 2020. Reinforcement learning tracking control for robotic manipulator with kernel-based dynamic model. IEEE Transactions on Neural Networks and Learning Systems 31 (9), 3570–3578.

Hwang, K., Ling, J., Wang, W., 2014. Adaptive reinforcement learning in box-pushing robots. In: IEEE International Conference on Automation Science and Engineering (CASE). Taipei, pp. 1182–1187.

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 (4-5), 698–721.

Kawano, H., 2013. Hierarchical sub-task decomposition for reinforcement learning of multi-robot delivery mission. In: IEEE International Conference on Robotics and Automation (ICRA). Karlsruhe, pp. 828–835.

Kober, J., Bagnell, J. A., Peters, J., 2013. Reinforcement learning in robotics: A survey. The International Journal of Robotics Research 32, 1238–1274.

Lin, H., Zhang, S., Li, X., Hwang, K., 2018. An adaptive decision-making method with fuzzy bayesian reinforcement learning for robot soccer. Information Sciences 436-437, 268–281.

Martínez-Tenor, A., Fernández-Madrigal, J. A., Cruz-Martín, A., González-Jiménez, J., jun 2018. Towards a common implementation of reinforcement learning for multiple robotic tasks. Expert Systems with Applications 100, 246–259.

Okal, B., Arras, K., 2016. Learning socially normative robot navigation behaviors with bayesian inverse reinforcement learning. In: IEEE International Conference on Robotics and Automation (ICRA). Stockholm.

Parisi, S., Abdulsamad, H., Paraschos, A., Daniel, C., Peters, J., 2015. Reinforcement learning vs human programming in tetherball robot games. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Hamburg, pp. 6428–6434.

Sutton, R. S., Barto, A. G., 2005. Reinforcement learning: An introduction. IEEE Transactions on Neural Networks 16, 285–286.

Wen, S., Wen, Z., Zhang, D., Zhang, H., Wan, T., 2021. A multi-robot path-planning algorithm for autonomous navigation using meta-reinforcement learning based on transfer learning. Applied Soft Computing 110.

Wiewiora, E., Cottrell, G. W., Elkan, C., 2003. Principled methods for advising reinforcement learning agents. In: 20th International Conference on Machine Learning (ICML). Washington.

Zhang, R., Li, Q. J., Bao, J., Liu, T., Liu, S., 2022. A reinforcement learning method for human-robot collaboration in assembly tasks. Robotics and Computer-Integrated Manufacturing 73 (C).