CEA Comité Español de Automática

XLVI JORNADAS AUTOMÁTICA CARTAGENA 2025

UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Aprendizaje por refuerzo de tareas de recogida y colocación

Hernández-Hernández, Jose María[a], Castaño-Amorós, Julio[b], Gil, Pablo[a,*]

[a]*Depto. Física, Ingeniería de Sistemas y Teoría de la Señal, Universidad de Alicante, Carretera de San Vicente del Raspeig, Alicante, 03690, Spain.*
[b]*Instituto Universitario de Investigación Informática, Universidad de Alicante, Carretera de San Vicente del Raspeig, 03690, Alicante, Spain.*

To cite this article: Hernández-Hernández, J.M., Castaño-Amorós, J., Gil, P. 2025. Reinforcement learning of pick and place tasks. Jornadas de Automática, 46. https://doi.org/10.17979/ja-cea.2025.46.12226

Resumen

La recogida y colocación de objetos es una de las tareas más comunes y de mayor implementación en entornos robotizados. Cualquier tarea de manipulación robótica compleja lleva intrínseca la necesidad de agarrar un objeto desde una ubicación para llevar a cabo alguna acción específica con él y, una vez terminada, volverlo a dejar en la misma o en otra ubicación. En este trabajo, entrenamos un agente robótico con aprendizaje profundo por refuerzo para llevar a cabo tareas de recogida y colocación en las que nuestro agente aprende a agarrar objetos y a depositarlos en ubicaciones de distinta dificultad, tales como, el interior de una cesta, la inserción en un hueco o ranura y el apilamiento sobre otro objeto de pequeñas dimensiones. Hemos definido y ajustado políticas y las hemos evaluado en 50 experimentos con poses arbitrarias de agarre. Los resultados obtenidos muestran que nuestras políticas entrenadas realizan con éxito la tarea en el 98 %, 78 % y 80 %, respectivamente según el tipo de ubicación.

Palabras clave: Aprendizaje por refuerzo, Robótica inteligente, Robots manipuladores, Tecnología Robótica.

Reinforcement learning of pick and place tasks

Abstract

Pick and place is one of the most common and widely implemented tasks in robotic environments. Any complex robotic manipulation task inherently involves the need to grasp an object from one location in order to perform a specific action with it and, once completed, return it to the same or another location. In this work, we trained a robotic agent with deep reinforcement learning to perform pick and place tasks in which our agent learns to grasp objects and place them in locations of varying difficulty, such as inside a basket, insertion into a hole or slot, and stacking on top of another small object. We have defined and adjusted policies and evaluated them in 50 experiments with arbitrary gripping poses. The results obtained show that our trained policies successfully perform the task in 98 %, 78 % y 80 % of cases, respectively, depending on the type of location.

Keywords: Reinforcement learning, Intelligent Robotics, Robots manipulators, Robotics technology.

1. Introducción

Las tareas de manipulación robótica han sido llevadas a cabo con éxito en entornos industriales durante varias décadas. En este tipo de entornos, es frecuente que los escenarios de trabajo de los robots sean estructurados o semiestructurados, es decir se disponga de un conocimiento total o parcial de las características de la escena y los objetos presentes en ella. Los avances de los últimos años, en sensores exteroceptivos (cámaras, fuerza-tacto, etc.), visión por computador y técnicas de control han favorecido que brazos robots sean capaces de desarrollar tareas concretas con un altísimo grado de autonomía, incluso cuando es requerida cierta flexibilidad de acciones que involucra adaptación de los movimientos o trayectorias de éstos.

No obstante, la incorporación de la robótica a otros entornos no-estructurados, tales como el entorno doméstico, el sanitario u otros espacios físicos como los presentes en los

*Autor para correspondencia: pablo.gil@ua.es

servicios logísticos, la industria del reciclado y reutilización, etc. ha generado nuevos desafíos en el campo de la manipulación como muestra (Billard and Kragic, 2019). En estos nuevos entornos, a priori, es habitual tener cierto desconocimiento del escenario de interacción del robot, debido a que se trata de entornos muy dinámicos, donde los elementos presentes en la escena cambian con el tiempo. Llevar a cabo tareas de manipulación en este contexto puede involucrar una o múltiples acciones físicas complejas combinadas, tales como: agarrar objetos, mover o transportar estos, moverlos con destreza teniéndolos sujetos o incluso llegar a cambiar su forma geométrica y/o su pose mientras se interactúa con ellos. En este contexto de tareas de manipulación, técnicas basadas en Aprendizaje Robótico e Inteligencia Artificial (Kroemer et al., 2021), como el Aprendizaje por Imitiación (Fang et al., 2019) y, especialmente, el Aprendizaje por Refuerzo (Han et al., 2023) (RL) favorecen la posibilidad de adaptación del robot a esos cambios que se pueden producir en el entorno con el que iteractúa (Cui and Trinkle, 2021).

En este trabajo, exploramos una tarea tradicional de manipulación robótica que es habitual en escenarios dinámicos, en concreto tareas del tipo *Pick and Place* (Lobbezoo et al., 2021) (Gomes et al., 2021). En nuestra propuesta, suponemos que un objeto desconocido se presenta apoyado sobre una mesa o superficie plana con una pose desconocida. Nuestro objetivo es entrenar políticas de RL para que un brazo con una pinza de dos dedos pueda aprender a acercarse, agarrar y transportar el objeto hasta situarlo en otra localización distinta. A diferencia de otros trabajos no hacemos uso de sistemas de teleoperación para enseñar al robot como en (Si et al., 2021) y tampoco hacemos uso de técnicas de aprendizaje por imitación como en (Zhu et al., 2023). Además, el método presentado permite entrenar la tarea con limitados recursos computacionales.

2. Aprendizaje por Refuerzo

2.1. Fundamentos

El *Reinforcement Learning* (RL) (Sutton and Barto, 2018) es una rama del aprendizaje automático en la que un agente, en este caso un brazo robótico, aprende a llevar a cabo una tarea a partir de su interacción con el entorno mediante la aplicación del paradigma de "prueba y error".

Esto es, en cada iteración t durante la ejecución de la tarea, el agente robótico recibe el estado actual s_t determinado por una descripción parcial de éste conocida como "observación". Y después, en función de ésta realiza una acción a_t, que puede involucrar uno o varios movimientos. A continuación, el agente recibe una recompensa r_t que le indica cómo de buena ha sido la acción a_t para transicionar del estado s_t al nuevo estado s_{t+1}. Este bucle se repite en cada iteración durante el proceso de aprendizaje. La función del agente robótico es aprender una política óptima $\pi(a_t|s_t)$ que le permita maximizar una recompensa acumulada definida como en (1)

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t \cdot r_{\mathrm{t}} \tag{1}$$

donde τ representa la secuencia de estados y acciones del agente y donde $\gamma^t \in [0, 1]$ es un factor de descuento que prioriza las recompensas a lo largo del tiempo.

La tarea de RL, en general, se modela como un *Markov Decision Process* (MDP) (van Otterlo and Wiering, 2012) donde s_{t+1} causado por a_t solo depende de s_t y no del historial completo de estados previos. Y, el ajuste de su política π se realiza mediante un proceso de entrenamiento, que consiste en la interacción del agente con su entorno para ver que valores de política optimizan el resultado de ejecución de la tarea. Cuando la política se ajusta mediante redes neuronales parametrizables en vez de con funciones matemáticas, nace lo que se conoce como *Deep Reinforcement Learning* (DRL) (Han et al., 2023) . EL DRL es especialmente de interés en tareas complejas cuyo espacio de acciones y observaciones es grande.

2.2. Método de entrenamiento

En este trabajo, se ha empleado el método conocido como *Soft Actor Critic* (SAC) (Haarnoja et al., 2018) para ajustar la política capaz de hacer que el agente robot realice tareas de recogida y colocación de objetos. El objetivo de SAC es obtener una política óptima π^* que maximice la recompensa esperada r_t , a la misma vez que también maximiza el valor de entropía de la política $H(\pi(\cdot|s_t))$ que mide la incertidumbre o diversidad de la política, como se muestra en (2).

$$\pi^* = \arg\max_{\pi} \mathrm{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t \left[r_t + \alpha \cdot H(\pi(\cdot|s_t))\right]\right] \tag{2}$$

donde el parámetro $\alpha > 0$ es un parámetro de ponderación que permite equilibrar la exploración del entorno y el funcionamiento del modelo.

La elección de SAC en ese trabajo es debida a que tiene un enfoque de entrenamiento *off-policy* que favorece la reutilización de experiencias generadas con otras políticas, aumentando la eficiencia del entrenamiento cuando hay limitaciones de recursos computacionales, como en nuestro caso ya que entrenamos con un Intel Core i7 8700K de 32GB de RAM que monta una GPU NVIDIA GeForce GTX-1080Ti de 11GB.

Además, permite mejorar la convergencia del entrenamiento y evitar caer en mínimos locales, gracias a la maximización del factor de entropía durante la optimización.

3. Sistema de recogida y colocación de objetos

3.1. Descripción del escenario propuesto

Para la implementación de la tarea se ha empleado el entorno de simulación Pybullet (Coumans and Bai, 2016-2021), así como las librería Gymnasium (Towers et al., 2024) que proporciona una interfaz para la creación y entrenamiento de agentes mediante RL, y la Stable-Baseline3 (Raffin et al., 2021) que facilita la implementación de algoritmos RL y DRL sobre Pytorch.

La creación del escenario en Pybullet consiste en un robot manipulador Kinova Gen 3 (Kinova, 2024) de 6 DoF con una pinza Robotiq 2F-140 (Robotiq, 2024) montada en su extremo. Además, se ha creado una mesa sobre la que situar objetos de distintas geometrías y tamaños sobre los que llevar a cabo la recogida, así como un recipiente o cesta que hace las labores de almacén para la colocación (Figura 1).

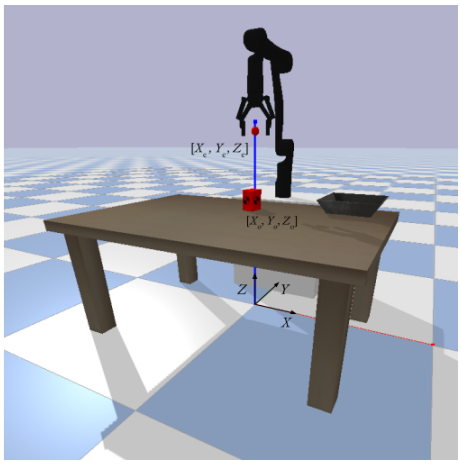


Figura 1: Esquema del entorno de RL usado en este trabajo.

Como se observa en la Figura 1, el espacio de observación para llevar a cabo DRL sobre nuestro escenario es el siguiente:
- Posición cartesiana del efector final (x_e, y_e, z_e)
- Posiciones articulares del robot $(q_1, q_2, q_3, q_4, q_5, q_6)$
- Velocidades articulares del robot $(\dot{q_1}, \dot{q_2}, \dot{q_3}, \dot{q_4}, \dot{q_5}, \dot{q_6})$
- Posición de apertura de la pinza q_P y sus velocidad $\dot{q_P}$
- Posición del objeto (x_O, y_O, z_O) y su orientación mediante cuaterniones (q_x, q_y, q_z, q_w)

3.2. *Modelado de la tarea*

Para maximizar el éxito del aprendizaje de la tarea de recogida y colocación de objetos se ha modelado la tarea como un conjunto de subtareas más simples. La aplicación del paradigma "divide y venceras" permite optimizar la búsqueda de soluciones para cada subtarea, definiendo funciones de recompensa para cada una de ellas. En concreto, el aprendizaje de la tarea completa de *Pick and Place* se aborda como un aprendizaje curricular. Es decir, primero se aborda el aprendizaje de una subtarea, después a ésta se le añade complejidad añadiéndole una nueva subtarea y así, sucesivamente, hasta que el acumulado de subtareas constituyen la tarea de manipulación completa. Así, el desarrollo del aprendizaje propuesto se aborda en cuatro fases o etapas (Figura 2) y por lo tanto, el espacio de acciones y las funciones de recompensa de la política de DRL varían según la fase de aplicación. En la fase 1, el robot manipulador aprende a orientarse y agarrar un objeto que se presupone situado justo debajo del efector en posición conocida. En la fase 2, se le añade la acción de tener que aprender a levantar el objeto en movimiento vertical. En una fase 3, se añade complejidad adicional, suponiendo que se desconoce la posición del objeto, y por lo tanto, antes de agarrar y levantar el objeto, se requiere aprender a aproximarse a él además de orientarse. Finalmente, en la fase 4, suma la subtarea de aprender a transportar y dejar el objeto agarrado a otra localización. Por lo tanto, en cada fase se entrena el sistema de la Figura 1, usando una o varias funciones de recompensa, adaptándolas a cada una de las nuevas primitivas de movimiento que se van acumulando para llevar a cabo la tarea de completa. Cada recompensa se normaliza en el rango [0, 1], y se ponderan mediante un peso para priorizar unas recompensas frente a otras.

3.2.1. *Fase 1. Agarre del objeto*

En la fase 1, se pretende aprender a agarrar el objeto. Para lograrlo, la recompensa r_{f1} se ha definido en función de un factor de alineamiento r_{alin} que depende de la diferencia angular entre la orientación de la pinza θ_{P} y la orientación del objeto θ_{O} que se desea agarrar y en función de un factor de la calidad de agarre r_{grasp}. En este caso, la orientación θ_{P} viene determinada por la posición articular q_6 del robot. El factor de calidad de agarre a su vez viene determinado, por un lado, por la distancia Euclidea entre el valor de apertura de la pinza P y el centro de masas del objeto O y por otro lado, por un factor de penalización de cierre prematuro de pinza, como se indica en (3).

$$r_{\text{f1}} = r_{\text{alin}} + r_{\text{grasp}} = (1 - \left|\cos(d_{\theta_{\text{P}},\theta_{\text{O}}})\right|) + (\mu_1 \cdot (1 - d_{\text{P,O}}) + \mu_2 \cdot p_{\text{p}}) \quad (3)$$

donde $\mu_1 = 2$ y $\mu_2 = 3$ son parámetros de escalado y, donde el factor de penalización de cierre prematuro p_{p}, modela aquellos casos en los que durante el aprendizaje la pinza se cierra antes de alcanzar la orientación del objeto. Este factor de penalización se calcula como en (4).

$$p_{\text{p}} = \begin{cases} -1, & \text{si } (\left|d_{\theta_{\text{P}},\theta_{\text{O}}} - 90°\right| > \theta_{max}) \ \wedge \ (\alpha_{\text{P}} > \alpha_{close}) \\ 0, & \text{en otros casos} \end{cases} \quad (4)$$

siendo θ_{max} la variación angular máxima tolerable entre la orientación de la pinza y el objeto, y siendo $\alpha_{close} = 0{,}25$ rad el valor angular de apertura de la pinza a partir del cual se considera que está cerrada.

3.2.2. *Fase 2. Elevación del objeto*

En la fase 2, se pretende aprender a levantar el objeto, una vez que éste se aprendió a agarrar correctamente en la fase 1. Por lo tanto, la recompensa r_{f2} está definida como la recompensa ya definida en (3) más un nuevo valor que define la recompensa de levantar correctamente el objecto r_{lift} y que modifica la recompensa total en esta nueva fase, como se muestra en (5).

$$r_{\text{f2}} = r_{\text{f1}} + r_{\text{lift}} = r_{\text{f1}} + (\sigma_1 \cdot h_{\text{O}}/H_{max} + \sigma_2 \cdot \text{-d}) \quad (5)$$

donde $\sigma_1 = 6$ y $\sigma_2 = 10$ son parámetros de escalado y, donde el factor h_{O}/H_{max} pondera la recompensa en función de la altura del objeto h_{O} apoyado sobre la superficie, siendo $H_{max} = 0{,}33$ m la elevación máxima que puede alcanzar el objeto durante el levantamiento o elevación deseada. Además, al igual que en la fase 1, se añade una penalización para reducir la distancia entre el centro de masas del objeto y el centro de la pinza, y así evitar que el objeto se separe de la pinza durante el levantamiento.

3.2.3. *Fase 3. Aproximación al objeto*

Una vez el sistema aprende a agarrar un objeto en cualquier orientación y a levantarlo es necesario dotar al sistema de más generalidad. Así, se hace necesario incorporar aprendizaje para que el sistema sea capaz de aproximarse al objeto en cualquier posición espacial, es decir variando además de su

orientación su posición inicial. Para ello, se incorpora un nuevo factor de recompensa r_{appr} que modifica el valor de la recompensa de las fases anteriores, como se indica en (6). Éste depende de una recompensa de distancia r_{dist} y de un factor de penalización por exceso de velocidad durante la aproximación p_{vel}. Además, se ha añadido una factor de recompensa adicional r_{goal} para premiar el aprendizaje cuando el objeto es alcanzado, es decir, se sitúa sobre éste.

$$r_{\text{f3}} = r_{\text{appr}} + r_{\text{f2}} = (r_{\text{dist}} + \lambda_1 \cdot p_{\text{vel}} + \lambda_2 \cdot r_{\text{goal}}) + (r_{\text{ali}} + r_{\text{grasp}} + \lambda_3 \cdot r_{\text{lift}}) \tag{6}$$

donde $\lambda_1 = 5$, $\lambda_2 = 8 \cdot 10^3$ y $\lambda_3 = 60$ son parámetros de escalado.

La recompensa de distancia r_{dist} se calcula como $r_{\text{dist}} = (1 - d_{\text{P,O}}/D_{max})$ siendo $d_{\text{P,O}}$ la distancia entre pinza y objeto, y $D_{max} = 0{,}488$ m la distancia máxima permitida entre estos elementos. El factor de penalización $p_{\text{vel}} = -v_{\text{e}}^2 \cdot (1 - d_{\text{P,O}}/D_{max})$ es dependiente de la velocidad linea del efector v_{e} y de la distancia entre pinza y objeto. Este factor aplica una penalización mayor cuando el robot está lejos de la pinza y, ésta decrece rápidamente cuando está cerca. Por otro lado, la recompensa por alcanzar el objeto r_{goal} viene determinada por (7).

$$r_{\text{goal}} = \begin{cases} 1, & \text{si } (h_{\text{O}} \geq H_{min}) \wedge (g == 1) \\ 0, & \text{en otros casos} \end{cases} \tag{7}$$

donde el factor g es un parámetro binario que expresa si el objeto está agarrado por la pinza o no lo está, adoptando los valores 1 o 0 respectivamente.

3.2.4. Fase 4. Transporte y colocación

Finalmente, es necesario incorporar la subtarea de transportar el objeto hasta la ubicación deseada y una vez allí, depositarlo y/o apoyarlo de nuevo sobre una superficie. Así, en esta fase 4, se combinan todas las fases anteriores para poder definir el aprendizaje de la tarea completa.

En esta última fase, se ha definido una nueva función de recompensa r_{place} que se añade a las ya existentes en fases anteriores. En particular, ésta depende de dos factores. Estos son un factor de aproximación al lugar donde depositar el objeto r_{appr2} y un parámetro que determina el éxito de alcanzar dicha localización r_{goal2}, como se muestra en la ecuación para r_{f4} mostrada en (8).

$$r_{\text{f4}} = r_{\text{f3}} + r_{\text{place}} = (r_{\text{appr}} + r_{\text{ali}} + r_{\text{grasp}} + r_{\text{lift}}) + (\eta_1 \cdot r_{\text{appr2}} + \eta_2 \cdot r_{\text{goal2}}) \tag{8}$$

donde $\eta_1 = 20$ y $\eta_2 = 10^4$ son parámetros de escalado y donde r_{appr2} se define como:

$$r_{\text{appr2}} = e^{-3 \cdot d_{\text{P,C}}} \tag{9}$$

y donde el r_{goal2} se modela como:

$$r_{\text{goa2}} = \begin{cases} 1, & \text{si } (d_{\text{P,C}} < d_{min}) \wedge (\alpha_{\text{P}} < \alpha_{\text{open}}) \\ 0, & \text{en otros casos} \end{cases} \tag{10}$$

siendo $d_{\text{P,C}}$, en ambas ecuaciones, la distancia entre la pinza y la ubicación de destino en la cual se depositará el objeto agarrado, donde d_{min} es un parámetro que determina la distancia mínima necesaria para considerar que la ubicación es adecuada, y $\alpha_{\text{open}} = 0{,}03$ rad es el valor angular mínimo necesario que debe alcanzar la apertura de la pinza α_{P} para considerarse que está lo suficientemente abierta para que el objeto pueda ser soltado.

4. Resultados experimentales

4.1. Entrenamiento del sistema por fases

Para ajustar las funciones de recompensa que se han propuesto en la sección anterior, se ha entrenado cada una de las fases anteriormente mencionadas, del siguiente modo. Se entrena el sistema en fase i para observar su rendimiento, después se modifica éste para convertirlo en un sistema en fase $i + 1$ y se vuelve a entrenar, y así sucesivamente hasta alcanzar la última fase. En la Figura 2 se muestran imágenes de los estados iniciales y finales del sistema en las fases 2 y 4.

La Tabla 1 muestra el rendimiento alcanzado por el sistema previamente entrenado en cada fase. En total se desarrollaron en cada fase un total de 50 episodios, siempre con el mismo objeto, un prisma rectangular. En todos los casos la tasa de fallo es pequeña, oscilando entre 1 y 3 fallos según la fase. Notar que la tarea no es la misma en cada fase y que las recompensas se van modificando ligeramente para hacer el sistema más robusto, aunque la tarea se vaya volviendo más compleja conforme se incrementa la fase. Notar que en la fase 4, se contempla el desarrollo completo de la tarea de recogida y colocación[1] , desde su aproximación y agarre hasta su colocación en un contenedor, y que se ha alcanzado un éxito del 98 % con el objeto prisma rectangular colocado en cualquier localización de la mesa.

Tabla 1: Rendimiento del agente robot para llevar a cabo la tarea de manipulación de un prisma rectangular de dimensiones 60x80x100 mm descrita en cada fase y pasos de tiempo en los que se maximiza la recompensa.

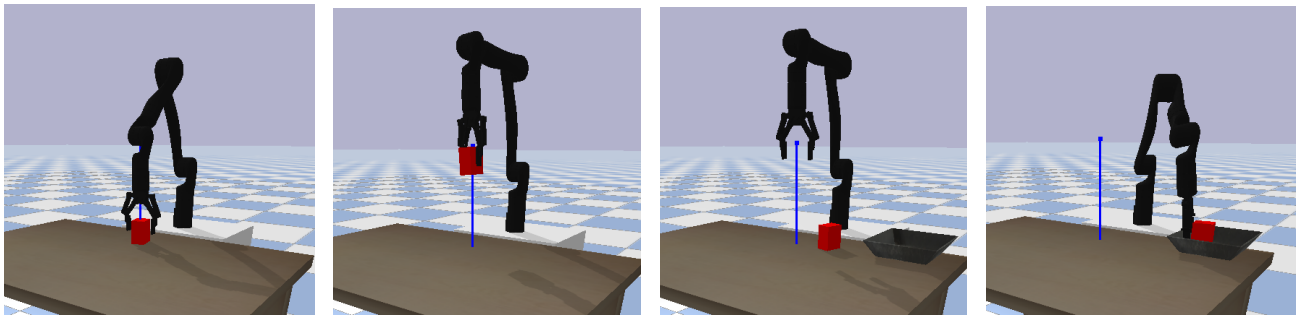
	Fase 1	Fase 2	Fase 3	Fase 4
Tasa de éxito	98 %	94 %	96 %	98 %
Recompensa r_{fi}	48	115	$8{,}5 \cdot 10^3$	10^4
Pasos de tiempo	10^5	$2 \cdot 10^4$	$1{,}5 \cdot 10^6$	$2{,}5 \cdot 10^6$

4.2. Generalización con otros objetos

Una vez demostrado que la tarea completa de recogida y colocación puede ser entrenada con éxito haciendo uso de políticas de DRL, se quiere determinar la capacidad del sistema entrenado con el prisma para replicar la tarea con otros tipos de objetos con los que no ha sido entrenado.

Para evaluar la capacidad de generalización, se ha empleado un conjunto de 10 objetos diferentes al prisma de la Tabla 1, tanto en geometría como en tamaño, así como distintos entre sí. El modelo tridimensional en formato URDF de cada uno de estos objetos, no vistos por el sistema durante el proceso de entrenamiento, ha sido extraído de la conocida base de datos *YCB Object and Model Set* (Calli et al., 2017).

[1] Ver https://youtu.be/Qa7HnIW0J34, consultado 16 de junio de 2025



(a) Fase 2. Estado inicial (b) Fase 2. Estado final (c) Fase 4. Estado inicial (d) Fase 4. Estado final

Figura 2: Ejemplos de ejecución del sistema DRL entrenado para la tarea de recogida y colocación, en fase 2 (elevación del objeto) y en fase 4 (transporte y colocación del objeto).

En concreto, en la Figura 3 se muestra el subconjunto de 10 objetos escogidos para probar el comportamiento de generalización del sistema, y en la Tabla 2 se describen sus características. El rendimiento medido como tasa de éxito para llevar a cabo la tarea completa de recogida y colocación descrita en la Fase 4 de la sección 3 para cada uno de los objetos se muestra en la Tabla 3.

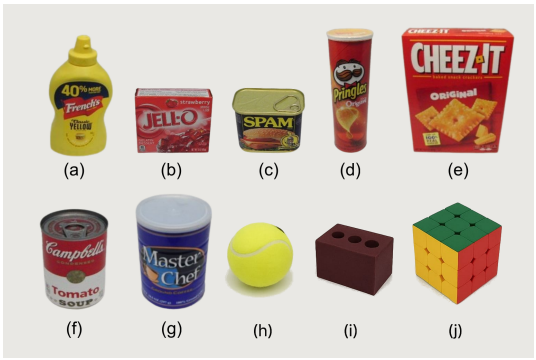


Figura 3: Subconjunto de *YCB Object and Model Set* empleado.

Tabla 2: Características del subconjunto de objetos de la Figura 3 empleados para medir la capacidad de generalizar de nuestro sistema DRL.

Objeto	Descripción	Dimensiones en mm
a	Bote de mostaza	50x85x175
b	Caja de gelatina en polvo	28x85x73
c	Lata de carne	50x97x82
d	Bote de patatas chips	75x250
e	Caja de galletas	60x158x210
f	Lata de tomate frito	66x101
g	Lata de café molido	75x250
h	Pelota de tenis	65
i	Ladrillo de juguete	50x75x50
j	Cubo de Rubik	57x57x57

La Tabla 3 muestra que la tasa de éxito es mejor en aquellos objetos cuya geometría, tanto en forma como tamaño, se asemeja al objeto prisma empleado en el entrenamiento del agente con DRL. Así, objetos como la lata de carne (c) y el cubo de Rubik (j) ofrecen rendimientos del 80 % y 90 %. Mientras, que la tarea de recogida y colocación tiene resultados muy variables con otro tipo de objetos. Por ejemplo, se consiguen tasas de éxito medio con objetos alargados y dimensiones similares como el bote de mostaza (a), la lata de sopa de tomate (f) y la lata de café molido (g), no importa que en algunos casos la forma sea cilíndrica en vez de prismática.

Tabla 3: Tasa de éxito (%) del agente entrenado con el prisma rectangular para llevar a cabo la tarea completa (Fase 4) con cada uno de los objetos de la Tabla 2.

a	b	c	d	e	f	g	h	i	j
60	30	80	18	12	70	64	44	48	90

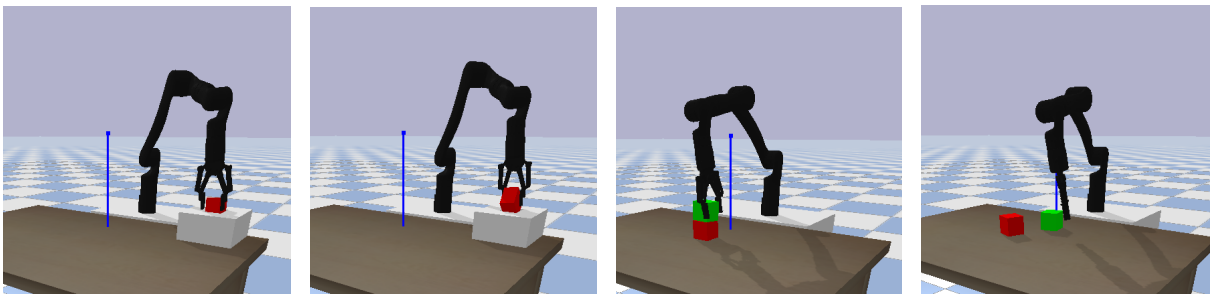
No obstante, el rendimiento decae bastante cuando se manipula el bote de patatas chips (d) y la caja de galletas (e), probablemente debido a su aumento de dimensiones que ha dificultado que el agente robot agarre el objeto lo más cerca posible de su centro de masa.

4.3. *Mejorando la precisión de la tarea*

Después de evaluar la capacidad de generalización del sistema, se ha decidido aumentar la complejidad de la tarea de recogida y colocación para evaluar su precisión ante nuevos desafíos, tales como tareas de manipulación de inserción o apilamiento. Para ello, se ha procedido a aplicar un enfoque basado en el concepto de *fine-tuning* o ajuste fino. La idea es emplear el mismo sistema con el agente robot entrenado con las políticas comentadas en la sección 3 y siguiendo el procedimiento descrito en sección 4.1. A partir de éste, se realiza un reentrenamiento incorporando nuevas acciones que modifiquen la tarea de recogida y colocación en la etapa del *colocación*, para que el agente robótico sea capaz de llevar a cabo los movimientos adecuados que permita realizar tareas de inserción[2] y de apilamiento[3]. En una tarea de inserción, el agente debe introducir un objeto en una caja con una ranura de tamaño similar al objeto. En una tarea de apilamiento, se debe conseguir colocar un objeto sobre otro de área reducida sin que se caiga.

En ambos casos, se ha requerido reentrenar el agente y adaptar los pesos de las recompensas, consiguiendo una tasa

[2] Ver https://youtu.be/0r1q-7bLou, consultado 16 de junio de 2025
[3] Ver https://youtu.be/tEaU08A-Vac, consultado 16 de junio de 2025



(a) Éxito (b) Fracaso (c) Éxito (d) Fracaso

Figura 4: Ejemplos de inserción de un objeto en una ranura o hueco de otro objeto y de apilamiento de un objeto sobre la superficie de otro.

de éxito del 78 % en inserciones y del 80 % en el apilamiento. En el caso de la inserción, se considera que la tarea es un éxito cuando el objeto queda insertado dentro de la ranura (Figura 4(a)) y fracaso cuando queda fuera porque tropieza (Figura 4(b)). En el apilamiento, la tarea ha tenido éxito cuando el objeto se coloca sobre el otro sin que se caiga (Figura 4(c)) o sin que el objeto que queda debajo se desplace más de 5 cm de la posición inicial (Figura 4(d)).

5. Conclusiones y trabajos futuros

En este trabajo, hemos evaluado en simulación políticas de RL por fases o etapas aplicando una estrategia de "divide y venceras", que garantice la realización de una tarea completa de recogida y colocación de objetos. Hemos entrenado un agente robótico mediante *aprendizaje curricular*, ajustando sus políticas con el método SAC, y usando un objeto de geometría prisma rectangular situándolo en una mesa en una variedad de poses aleatorias.

Los resultados experimentales han mostrado que nuestro sistema aprende con éxito variando la tarea de recogida y colocación en sus acciones finales, mostrando capacidades de inserción en ranuras o apilamiento de objetos. Además, las políticas entrenadas con el prisma son satisfactorias con algunos objetos de distinta geometría y tamaño, aunque con otros tiene un rendimiento pobre. Por lo tanto, el trabajo futuro se centrará en estudiar mejoras en la capacidad de generalización de la política para otros objetos.

Agradecimientos

Este trabajo ha sido realizado gracias al proyecto REMAIN (S1/1.1/E0111) financiado con fondos del programa Interreg-VI Sudoe y FEDER, así como a la Universidad de Alicante a través de la beca predoctoral UAFPU21-26.

Referencias

Billard, A., Kragic, D., 2019. Trends and challenges in robot manipulation. Science 364 (6446).
DOI: 10.1126/science.aat8414

Calli, B., Singh, A., Bruce, J., Walsman, A., Konolige, K., Srinivasa, S., Abbeel, P., Dollar, A., 2017. Yale-cmu-berkeley dataset for robotic manipulation research. The Int. J. of Robotics Research 36 (3), 261–268.
DOI: 10.1177/0278364917700714

Coumans, E., Bai, Y., 2016-2021. Pybullet, a python module for physics simulation for games, robotics and machine learning.
URL: http://pybullet.org

Cui, J., Trinkle, J., 2021. Toward next-generation learned robot manipulation. Science Robotics 6 (54).
DOI: 10.1126/scirobotics.abd9461

Fang, B., Jia, S., Guo, D., et al., 2019. Survey of imitation learning for robotic manipulation. Int. J. of Intelligent Robotics Application 3, 362–369.
DOI: 10.1007/s41315-019-00103-5

Gomes, N., Martins, N., Lima, J., Wörtche, H., 2021. Deep reinforcement learning applied to a robotic pick and place application. In: Pereira, A., Fernandes, F., Coelho, J., Pacheco, M., Alves, P., Lopes, R. (Eds.), Optimization, Learning Algorithms and Applications. Springer International Publishing, pp. 251–265.
DOI: 10.1007/978-3-030-91885-9$_1$8

Haarnoja, T., Zhou, A., Abbeel, P., Levine, S., 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Dy, J. G., Krause, A. (Eds.), Proc. of the 35th Int. Conf. on Machine Learning (ICML). Vol. 80. PMLR, pp. 1856–1865.

Han, D., Mulyana, B., Stankovic, V., Cheng, S., 2023. A survey on deep reinforcement learning algorithms for robotic manipulation. Sensors 23 (7).
DOI: 10.3390/s23073762

Kinova, 2024. Kinova.
URL: https://www.kinovarobotics.com/product/gen3-robots

Kroemer, O., Niekum, S., Konidaris, G., 2021. A review of robot learning for manipulation: Challenges, representations, and algorithms. J. of Machine Learning Research 22 (30), 1–82.
URL: http://jmlr.org/papers/v22/19-804.html

Lobbezoo, A., Qian, Y., Yanjun, K., Kwon, H.-J., 2021. Reinforcement learning for pick and place operations in robotics: A survey. Robotics 10 (3).
DOI: 10.3390/robotics10030105

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N., 2021. Stable-baselines3: Reliable reinforcement learning implementations. J. of Machine Learning Research 22 (268), 1–8.
URL: http://jmlr.org/papers/v22/20-1364.html

Robotiq, 2024. Robotiq.
URL: https://robotiq.com/

Si, W., Wang, N., Yang, C., 2021. A review on manipulation skill acquisition through teleoperation-based learning from demonstration. Cognitive Computation and Systems 3 (1), 1–16.
DOI: https://doi.org/10.1049/ccs2.12005

Sutton, R. S., Barto, A., 2018. Reinforcement learning: An introduction. MIT Press, Cambridge, Massachusetts.

Towers, M., Kwiatkowski, A., Terry, J. K., Balis, J. U., Cola, G. D., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J. J., Tan, H., Younis, O. G., 2024. Gymnasium: A standard interface for reinforcement learning environments. CoRR abs/2407.17032.
DOI: 10.48550/ARXIV.2407.17032

van Otterlo, M., Wiering, M., 2012. Reinforcement learning and markov decision processes. In: Wiering, M., van Otterlo, M. (Eds.), Reinforcement Learning: State of the Art. Springer Berlin Heidelberg, pp. 3–42.
DOI: 10.1007/978-3-642-27645-3$_1$

Zhu, Y., Joshi, A., Stone, P., Zhu, Y., 2023. Viola: Imitation learning for vision-based manipulation with object proposal priors. In: Liu, K., Kulic, D., Ichnowski, J. (Eds.), Proc. of The 6th Conference on Robot Learning (CoRL). Vol. 205. PMLR, pp. 1199–1210.