XXI Simposio CEA de Control Inteligente
24-26 de junio de 2026, Salamanca

Aprendizaje por imitación y refuerzo para la generación de trayectorias de un robot colaborativo

Menéndez-García, Alejandro.[a,*], Sierra-García, Jesús Enrique.[b]

[a] *Digitalización, Universidad de Burgos, Campus Río Vena, Avda. Cantabria, s/n, 09006 Burgos, España.*

To cite this article: Menéndez García, Alejandro, Sierra García, Jesús Enrique, 2026. "Imitation Learning and Reinforcement Learning for the trajectory generation of a Collaborative Robot" XXI Simposio CEA de Control Inteligente.

Resumen

Este artículo propone una arquitectura de control jerárquica en tres fases para generar movimientos articulares que lleven el efector final hasta una pose objetivo. La metodología se valida en simulación con el robot colaborativo ABB GoFa CRB 15000-5/0.95. La estructura del trabajo combina un planificador experto basado en optimización GPU (cuRobo), un módulo de clonación de comportamiento (BC) sobre un dataset de 10 000 demostraciones, una etapa de Dagger (Dataset Aggregation) para corregir el desplazamiento de distribución de la política clonada, y una política residual entrenada mediante PPO (Proximal Policy Optimization) en el simulador Isaac Lab, penalizando las correcciones excesivas respecto a la acción propuesta por la política clonada. La política residual aplica una corrección acotada (±0,5 rad) sobre la acción del BC, lo que permite refinar el comportamiento sin destruir el conocimiento aprendido por imitación. Tras un entrenamiento de 300 iteraciones con 128 entornos paralelos sobre una distribución de metas centrada en los fallos del BC, el sistema alcanza una tasa de éxito del 95,33 % en 300 episodios de evaluación con umbrales de 2 cm de error de posición y 5,73° de error de orientación. El error medio en la pose final es de 1,31 cm y 5,2°, alcanzando en la mejor pose por episodio un error medio de 8,72 mm y 3,1°. Los resultados muestran que la combinación de BC supervisado, DAgger y refuerzo es viable para inicializar políticas de control reactivo en manipuladores colaborativos sin requerir cinemática inversa analítica en línea.

Palabras clave: Aprendizaje por imitación, aprendizaje por refuerzo, robot colaborativo, planificación de trayectorias, simulación.

Imitation Learning and Reinforcement Learning for the trajectory generation of a Collaborative Robot

Abstract

This paper proposes a three-stage hierarchical control architecture to generate joint movements that drive the end-effector to a target pose. The methodology is validated in simulation using the ABB GoFa CRB 15000-5/0.95 collaborative robot. The proposed structure combines a GPU-optimization-based expert planner (cuRobo), a behavior cloning (BC) module trained on a dataset of 10,000 demonstrations, a Dagger (Dataset Aggregation) stage to correct the distribution shift of the cloned policy, and a residual policy trained with PPO (Proximal Policy Optimization) in the Isaac Lab simulator, penalizing excessive corrections with respect to the action proposed by the cloned policy. The residual policy applies a bounded correction (±0,5 rad) on top of the BC action, allowing the behavior to be refined without destroying the knowledge learned through imitation. After 300 training iterations with 128 parallel environments over a goal distribution focused on BC failures, the system achieves a success rate of 95,33% in 300 evaluation episodes, using thresholds of 2 cm for position error and 5,73° for orientation error. The mean final-pose error is 1,31 cm and 5,2°, while the best pose reached in each episode achieves a mean error of 8,72 mm and 3,1°. The results show that the combination of supervised BC, DAgger, and reinforcement learning is a viable approach for initializing reactive control policies in collaborative manipulators without requiring online analytical inverse kinematics.

Keywords: Imitation learning, reinforcement learning, collaborative robot, trajectory planning, simulation.

1. Introducción

Durante los últimos años el paradigma de control en manipuladores ha experimentado un giro significativo: las políticas visuo-motoras aprendidas directamente de demostraciones han desplazado, en muchas tareas de manipulación, a las clásicas pilas de planificación más control servo. Trabajos como Diffusion Policy (Chi et al., 2023), que

*Autor para correspondencia: alejandro.menendez@ubu.es
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

representa la política como un proceso de difusión condicional sobre acciones, o los modelos visión-lenguaje-acción (VLA) RT-2 (Brohan et al., 2023), Octo (Octo Model Team, 2024) y OpenVLA (Kim et al., 2024) han mostrado que una sola red neuronal puede asumir desde la percepción hasta la generación de acciones articulares, alcanzando niveles de generalización antes inalcanzables. Esta línea sustituye la cinemática inversa explícita por un mapeo aprendido observación → acción que se ejecuta a frecuencias de control elevadas y reacciona ante perturbaciones que no estaban en la trayectoria nominal.

El obstáculo principal de estos enfoques es el coste de los datos. RT-2 se entrena con cientos de miles de episodios recogidos por teleoperación humana en flotas de robots, y OpenVLA con el dataset Open X-Embodiment, de un millón de trayectorias procedentes de 22 plataformas distintas (Kim et al., 2024). La generación de ese volumen de datos es difícilmente accesible en la mayoría de las aplicaciones. Por otra parte, en un escenario industrial con un manipulador y un volumen de trabajo conocidos, gran parte de la riqueza de un dataset humano es innecesaria: basta con cubrir el espacio de tareas relevante con trayectorias cinemáticamente correctas. Esta observación es la que motiva una segunda línea de trabajo: utilizar planificadores GPU rápidos como expertos sintéticos. cuRobo (Sundaralingam et al., 2023) genera trayectorias libres de colisiones en decenas de milisegundos resolviendo en paralelo cientos de inicializaciones de optimización; integrado con Isaac Lab (Mittal et al., 2025) permite producir miles de demostraciones expertas por hora sobre un modelo de robot dado, sin intervención humana.

Disponer de un dataset experto sintético no resuelve, sin embargo, el problema central de la clonación de comportamiento: el desplazamiento covariante. Una política supervisada sobre pares (estado, acción experta) se enfrenta en evaluación a la distribución de estados generada por sí misma, que difiere de la del experto, y los pequeños errores se acumulan a lo largo del episodio (Deshpande et al., 2024; Liang et al., 2024). Trabajos recientes en manipulación fina han documentado este efecto con detalle y han propuesto mitigaciones que van desde la aumentación de datos basada en continuidad (Deshpande et al., 2024) hasta el uso de información histórica de acción (Liang et al., 2024). En este trabajo recurrimos a una mitigación más simple y bien establecida: DAgger (Ross et al., 2011), en la que la política aprendida se despliega en el simulador y, sobre cada estado visitado, se consulta al experto cuRobo para obtener la acción correctiva, agregando esos pares al dataset. El experto sintético hace que esta consulta sea barata, lo que vuelve a DAgger especialmente atractivo en este contexto.

Una segunda dirección de mejora consiste en seguir refinando la política clonada mediante aprendizaje por refuerzo. Bajo el nombre genérico de *imitation-bootstrapped reinforcement learning*, varios trabajos recientes han demostrado que arrancar el RL desde una política supervisada acelera la convergencia y reduce drásticamente la exploración insegura (Hu et al., 2024; Dalal et al., 2024). En esta misma familia, el *residual reinforcement learning* (Johannink et al., 2019) propone superponer una corrección residual acotada sobre una política base existente en lugar de reentrenar toda la política. Aplicado a una política BC, el residuo concentra el aprendizaje justamente en los puntos donde el BC falla, mantiene la mayor parte del comportamiento aprendido por imitación y evita explorar regiones del espacio de estados que producirían trayectorias físicamente inviables. En nuestra implementación, el residuo está acotado a ±0,5 rad por articulación y se entrena con PPO sobre un anclaje cuadrático que penaliza desviaciones grandes respecto al BC.

Este artículo se enmarca en la línea de investigación iniciada en (Menéndez-García y Sierra-García, 2025), donde se exploró el uso de DDPG (Deep Deterministic Policy Gradient) para resolver el modelo cinemático inverso de un robot colaborativo como mapeo estático pose–articulaciones. El presente trabajo aborda un problema fundamentalmente distinto: una política reactiva de control que produce, en cada paso del episodio, la acción articular adecuada para seguir una trayectoria completa hasta una pose objetivo teniendo en cuenta la dinámica del manipulador. Este planteamiento es coherente con los casos de uso del proyecto europeo MANiBOT (MANiBOT, 2025), en los que un manipulador móvil bimanual debe ejecutar tareas de alcance en escenarios reales como la reposición de estanterías o el manejo de equipajes, y se alinea con los trabajos del grupo en agentes colaborativos para entornos industriales (Sierra-García y Santos, 2021; Bayona et al., 2024). En entornos con presencia humana, el ajuste de políticas mediante RL plantea además requisitos específicos de seguridad (Thumm y Althoff, 2022) que se contemplan implícitamente al acotar el residuo y anclarlo a la política experta.

En entornos con presencia humana, el ajuste de políticas mediante aprendizaje por refuerzo plantea requisitos específicos de seguridad. En este trabajo, la acotación del residuo y su anclaje a la política clonada se emplean únicamente como mecanismo para limitar desviaciones excesivas respecto a la trayectoria base, pero no constituyen una garantía formal de ausencia de colisiones con humanos.

Frente a enfoques clásicos basados en la resolución explícita de la cinemática inversa y el posterior control dinámico, la metodología propuesta permite aprender directamente una política reactiva observación–acción, evitando resolver el problema cinemático inverso en línea y facilitando la corrección de desviaciones durante la ejecución. Además, el uso de un experto sintético como cuRobo permite generar grandes volúmenes de demostraciones sin teleoperación humana.

Este proyecto se divide en cuatro apartados fundamentales. En primer lugar, se propone una metodología reproducible que combina un planificador experto basado en cuRobo, una política inicial obtenida mediante clonación de comportamiento, una etapa de corrección con DAgger y un refinamiento final mediante PPO residual en Isaac Lab. En segundo lugar, se plantea un muestreo de metas centrado parcialmente en los fallos de la política clonada, de forma que el aprendizaje por refuerzo se concentra en las regiones donde la política inicial presenta más dificultades. En tercer lugar, se introduce una penalización sobre la magnitud de la corrección residual, que evita que PPO se aleje en exceso de la acción propuesta por la política clonada. En cuarto lugar, se analizan los modos de fallo restantes, principalmente asociados a metas

alejadas de la configuración inicial y a configuraciones espejo del efector final.

El resto del artículo se estructura como sigue. La Sección 2 describe el robot, el entorno de simulación y la arquitectura BC. La Sección 3 detalla el proceso de entrenamiento, incluyendo DAgger y el ciclo PPO residual. La Sección 4 presenta los resultados experimentales. La Sección 5 expone las conclusiones y los trabajos futuros.

2. Metodología

La arquitectura propuesta combina tres tipos de aprendizaje complementarios. La Figura 1 muestra el esquema general del planteamiento del proyecto:

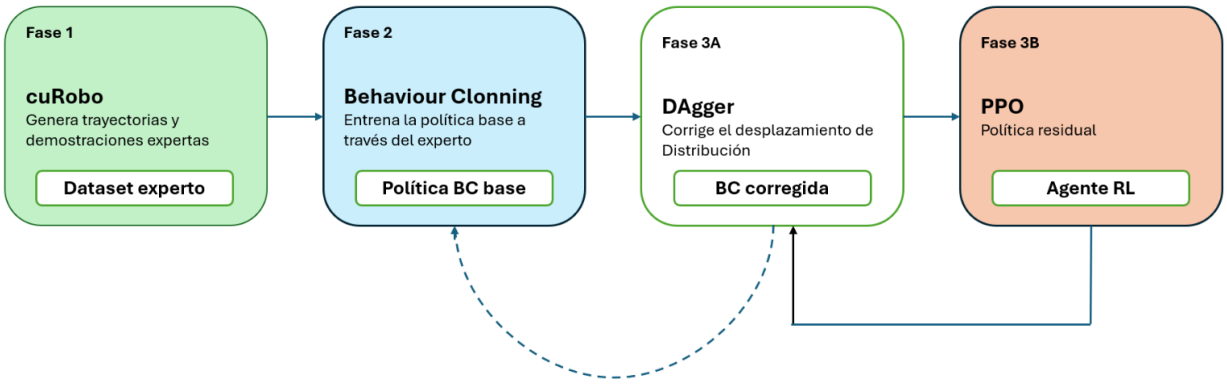


Figura 1: Desarrollo en tres fases. cuRobo; Behavior Cloning; DAgger; PPO añade una política resultada de aprendizaje por refuerzo.

2.1. Robot y entorno de simulación

El robot empleado es el ABB GoFa CRB 15000-5/0,95, un manipulador colaborativo de seis grados de libertad con un alcance de 950 mm y carga útil de 5 kg, equipado con sensores de par en cada articulación para limitación de fuerza (ABB, 2024). El URDF (Unified Robot Description Format) utilizado se construye a partir de los mallados oficiales del fabricante y se importa al simulador Isaac Lab 2.3 (Mittal et al., 2025), construido sobre Isaac Sim 5.1 y PhysX. El control se realiza por posición articular sobre una simulación a 120 Hz, lo que produce un paso de control efectivo de 16,7 ms y una duración de episodio de 8 s (480 pasos). El espacio de trabajo de las metas se acota a un prisma rectangular en el frame base del robot con x ∈ [0,30; 0,75] m, y ∈ [−0,40; 0,40] m y z ∈ [0,25; 1,00] m, lo que cubre la mayor parte del volumen alcanzable manteniendo configuraciones lejos de singularidades.

La observación que recibe la política tiene dimensión 27 y se compone de la posición articular relativa al joint_pos_default (6), la velocidad articular (6), la información de la meta (posición, cuaternión, distancia de posición y de orientación al objetivo: 9), y la última acción enviada (6). La acción tiene dimensión (6) y se interpreta como un offset sobre la posición articular de referencia, escalado a unidad.

2.2. Generación del dataset experto con cuRobo

cuRobo (Sundaralingam et al., 2023) es una librería de generación de movimiento basada en GPU que combina cinemática inversa en paralelo y optimización de trayectoria por L-BFGS con múltiples semillas. Para este trabajo se ha configurado cuRobo con el URDF (Unified Robot Description Format) del GoFa y la pose de referencia q_{home} = [0,0; −0,7; 0,8; 0,0; 0,8; 0,0] rad. Para cada uno de los 10000 episodios del dataset se muestrea uniformemente una pose objetivo (posición y cuaternión) dentro del prisma de trabajo y se invoca el planificador con horizonte 96 pasos, obteniéndose una trayectoria articular q*t* continua y sin colisiones. Cada par (observación, acción experta) se almacena junto con la fase normalizada φ = t/(H−1) y un conjunto de descriptores temporales (sen(πφ), cos(πφ), φ^2) que permiten al modelo razonar sobre el progreso del episodio. El dataset resultante contiene 960 000 transiciones y se almacena en formato NPZ.

2.3. Behavior Cloning

El modelo BC es una red completamente conectada con 4 capas ocultas de 1024 unidades, LayerNorm y activación SiLU, que recibe un vector de entrada de dimensión 17 (posición, cuaternión y descriptores de fase) y produce una posición articular absoluta de dimensión 6. En esta versión no se emplea información visual ni sensores externos del entorno. La política BC se entrena únicamente con información propioceptiva del robot, la pose objetivo y descriptores temporales de la trayectoria. A diferencia de la versión abierta inicial, la variante cerrada incorpora también la posición articular actual q*t*, lo que aporta información de retroalimentación y mejora la robustez ante perturbaciones, en línea con la propuesta de BC con restricciones históricas de (Liang et al., 2024).

El entrenamiento minimiza una pérdida SmoothL1 ponderada que asigna mayor peso al tramo final de cada trayectoria, tal y como muestra la siguiente ecuación:

$$L_{BC} = \left(\frac{1}{N}\right)\sum_t w(t) \times SmoothL1(\pi_{BC}(o_t), q_t^*) \quad (1)$$

donde w(t) = 1 + (γ − 1) · t/(H−1), con γ = 4 como ganancia del peso terminal. El optimizador es *'AdamW'* con *'learning rate'* $1{,}5\times10^{-4}$, '*decay coseno*' hasta $1{,}0\times10^{-5}$, *'batch'* de 256 y 260 épocas. Tras el entrenamiento el mejor checkpoint logra una pérdida de validación de $1{,}56\times10^{-4}$, lo que se traduce en errores medios sobre el dataset de 1,7 mm en posición y 1,1° en orientación.

2.4. DAgger: corrección del desplazamiento de distribución

La política BC entrenada se ejecuta sobre el simulador y, en cada paso visitado por la red, se invoca a cuRobo para obtener la acción experta sobre ese estado, agregándose el par (o, q*) al dataset original (Ross et al., 2011). Se han recogido aproximadamente 200000 transiciones adicionales y se ha reentrenado el BC con la unión de ambos datasets durante 80 épocas adicionales. El modelo resultante, denominado BC-cerrado-DAgger, recupera trayectorias que retornan al experto cuando el robot se desvía. Esta única pasada de DAgger es suficiente porque el experto cuRobo es determinista y rápido, lo que permite consultarle sobre cualquier estado visitado en cuestión de milisegundos.

2.5. PPO residual con penalización respecto a la política clonada

La política final se construye como una superposición aditiva (Johannink et al., 2019):

$$a_t = clip(a_{BC}(o_t) + s \cdot a_{res}(o_t),\ a_{BC} \pm a_{max}) \quad (2)$$

donde a_{BC} es la acción determinista del BC-DAgger, a_{res} es la salida de una política gaussiana entrenada con PPO, s = 0,5 es la escala del residuo y a_{max} = 0,5 rad/paso es el máximo desvío permitido por articulación. La política residual es una red [256, 128, 64] con activación ELU. Para mantener la corrección pequeña durante todo el entrenamiento se añade un coste de anclaje a la función objetivo de PPO:

$$L_{total} = L_{PPO} + \lambda_{anc} \cdot \| a_{res} \|^2 \quad (3)$$

con λ_{anc} = 0,03 y desviación estándar inicial de la política σ = 0,035 rad. El esquema actor-crítico de PPO emplea ratio_clip = 0,2, GAE con λ = 0,95 y descuento γ = 0,99. Esta estrategia es coherente con el *imitation-bootstrapped RL* de (Hu et al., 2024), que muestra que partir de una política aprendida mejora drásticamente la eficiencia muestral del refuerzo posterior.

3. Entrenamiento del sistema

3.1. Función de recompensa

La recompensa por paso se diseña para favorecer una convergencia progresiva hacia la pose objetivo, manteniendo a la vez la suavidad de la acción articular.

Sea $d_p(t) = \| p_{ee}(t) - p_g \|_2$ el error cartesiano de posición entre el efector final y la posición objetivo, y sea $d_o(t)$ el error angular entre la orientación actual del efector final y la orientación objetivo. La recompensa total se expresa como la suma de las distintas recompensas:

$$r_{total} = \Delta t \times \sum_t r_w \quad (4)$$

Donde r_w equivale a la suma de todas estas subfunciones de recompensa:

$$r_{\text{pose}} = -1{,}5\left(\frac{d_p(t)}{0{,}20} + 0{,}25 * \mathrm{d_o(t)}\right) \quad (5)$$

$$r_{prec_{combo=}} - 8 d_p(t) \quad (6)$$

$$r_{\text{prog}} = 40\,\Delta d_p(t) \quad (7)$$

$$r_{ori_{progg}} = 5\,G\left(d_p(t)\right)\,\Delta d_o(t) \quad (8)$$

$$r_{\text{fine}_{\text{track}}} = 4\left(1 - \tanh\left(\frac{d_p(t)}{0{,}08}\right)\right) \quad (9)$$

$$r_{fine_{pos}} = 6\left(1 - \tanh\left(\frac{d_p(t)}{0{,}02}\right)\right) \quad (10)$$

$$r_{fine_{ori}} = 6\left(1 - \tanh\left(\frac{d_o(t)}{0{,}1}\right)\right) \quad (11)$$

$$r_{succ_{reg}} = 30 S_{reg}\left(d_p(t), d_o(t)\right) \quad (12)$$

$$r_{success_{bonus=}}\ 300\left[d_p(t) < 0{,}02 \wedge d_o(t) < 0{,}10\right] \quad (13)$$

$$\mathrm{r_{smooth}} = -10^{-4} \| \mathrm{a_t - a_{t-1}} \|^2 - 10^{-4} \| \dot{\mathrm{q}}_\mathrm{t} \|^2 \quad (14)$$

Donde Δt es el factor temporal del entorno aproximado de 0,0167 s. $\Delta d_p(t)$ y $\Delta d_o(t)$ representan la mejora acotada del error de posición y orientación respecto al paso anterior. La función $G\left(d_p(t)\right) = 1 - tanh(k_G d_p(t))$ actúa como una función de proximidad, reduciendo el peso de los términos de orientación cuando el efector final todavía está lejos del objetivo. El término B_{prec} recompensa la entrada en regiones de precisión final. S_{reg} recompensa que el efector final esté simultáneamente cerca de la posición y la orientación objetivo. Una parte del bonus de éxito se activa únicamente cuando se cumplen los umbrales $d_p(t) < 2$ cm y $d_o(t) < 0{,}10$ rad, equivalente a 5,73°.

La Tabla 1 resume los pesos principales de cada subfunción de recompensa empleada durante el entrenamiento de la política residual PPO. Se dio mayor importancia a los términos de progreso y precisión para favorecer la convergencia hacia la meta, mientras que las penalizaciones de acción y velocidad se mantuvieron bajas para evitar oscilaciones sin limitar la corrección residual.

Tabla 1: Términos de la función de recompensa empleada para entrenar la política residual PPO.

Nombre del término	Peso	Función
pose_error	−1,5	Norma combinada pos+ori
position_progress	40,0	Reducción de distancia
tracking (tanh)	4,0	Penalización suave de pos.
tracking_fine_success_pos	6,0	Recompensa fina <2 cm
orientation_progress	5,0	Reducción de error angular
tracking_ori_fine_success	6,0	Recompensa fina <5,73°
precision_combo	8,0	Pos+Ori. dentro del umbral
success_region	30,0	Región estable de éxito
success_bonus	300,0	Bonus terminal de éxito
action_rate / joint_vel	−0,0001	Suavidad de la acción

3.2. Ciclo de entrenamiento de PPO

La política residual se entrena durante 300 iteraciones con 128 entornos paralelos en Isaac Lab, recogiendo 96 pasos por entorno y por iteración (12 288 transiciones por iteración). Cada iteración realiza 4 épocas de PPO con 4 mini lotes. El *'learning rate'* es 2×10^{-5} con KL-adaptativo (*'umbral KL'* = 0,01), *'grad-norm clip'* = 1,0 y '*value-clip'* = 0,2. Los entornos se reinician con metas muestreadas con probabilidad 0,65

desde un dataset de 300 fallos del BC-DAgger inicial y con probabilidad 0,35 desde la distribución uniforme del espacio de trabajo. Este muestreo dirigido concentra la corrección del modelo justamente en las regiones donde el BC falla y mantiene a la vez la generalización al volumen completo.

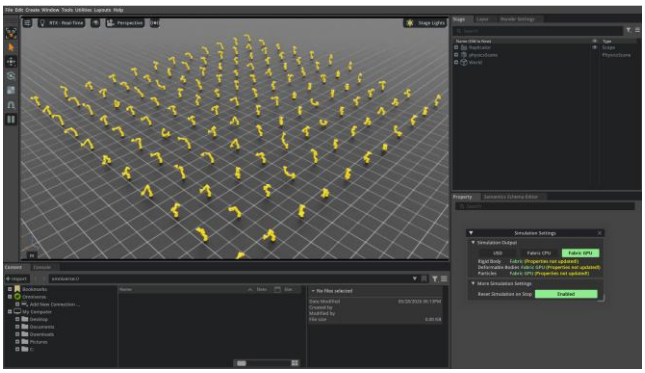


Figura 2: Isaac Lab 128 entornos paralelos

El entrenamiento se ejecuta en una NVIDIA RTX 5080 Laptop y mantiene una tasa de simulación efectiva en torno a 4 000 pasos por segundo. El tiempo total de entrenamiento son aproximadamente 12 minutos.

4. Resultados

4.1. Evolución durante el entrenamiento

La Figura 3 muestra la evolución de la recompensa por episodio durante el entrenamiento PPO residual. Para facilitar la interpretación de la curva, se representa la media móvil de los últimos 20 episodios, hasta un total de 300 iteraciones PPO. Tras una fase inicial de adaptación mediante Behavior Cloning, la recompensa aumenta rápidamente y posteriormente entra en una región más estable, con valores medios aproximados en torno a 125.

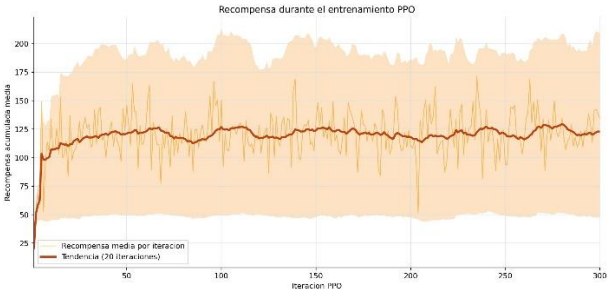


Figura 3: Evolución de la recompensa media.

Debe tenerse en cuenta que esta curva no corresponde a un experimento clásico de RL con un único entorno y episodios homogéneos. En Isaac Lab el entrenamiento se ejecuta con múltiples entornos paralelos, reinicios asíncronos y objetivos de distinta dificultad, por lo que la recompensa por episodio presenta una variabilidad elevada.

Además, la recompensa combina términos densos de seguimiento, términos de precisión y un bonus disperso de éxito. En la Figura 3 no aparecen valores de recompensa de magnitud 300 asociados directamente al bonus, ya que Isaac Lab escala internamente cada término por el paso temporal del entorno. Por tanto, para un paso aproximado de 0,0167 s, el bonus definido con peso 300 contribuye con unos $300 \cdot 0{,}0167 \approx 5$ puntos por paso en el que se cumple la condición de éxito, y no como un salto bruto de +300.

Por este motivo, la Figura 3 debe interpretarse como una indicación de la tendencia general del entrenamiento, y no como una medida directa del rendimiento final de la política. La validación cuantitativa se realiza en el apartado 4.2 mediante una evaluación independiente con métricas explícitas de tasa de éxito, error final de posición y error final de orientación.

4.2. Evaluación final

Se evalúa el checkpoint best_rollout.pt sobre 300 episodios independientes con semillas aleatorias y umbrales de éxito de 2 cm y 5,73° (0,10 rad). La Tabla 2 resume las métricas globales.

Tabla 2: Resultados de la evaluación sobre 300 episodios bajo umbrales de 2 cm y 5,73°.

Métrica	**Valor**
Tasa de éxito (pose final)	95,33 % (286/300)
Error medio de posición en casos exitosos (pose final)	1,31 cm
Error medio de orientación en casos exitosos (pose final)	5,2°
Error medio de posición (mejor por episodio)	8,72 mm
Error medio de orientación (mejor por episodio)	3,1°
Pasos medios por episodio en éxito	≈ 150
Tiempo de inferencia por paso	< 1 ms

La Tabla 2 distingue entre la “mejor pose por episodio”, que mide el menor error alcanzado durante la trayectoria, y la “pose final”, que evalúa el error en el último paso. En la evaluación, la mejor pose presentó errores medios de 8,72 mm y 3,1°, mientras que la pose final alcanzó 1,31 cm y 5,2°. Esta diferencia indica una ligera degradación entre el mejor instante y el final del episodio, probablemente debida a pequeñas correcciones residuales y a la ausencia de una fase terminal específica de mantenimiento de la pose. Aun así, la política permanece dentro de los umbrales de éxito en el 95,33 % de los casos, reduciendo las oscilaciones terminales observadas en versiones anteriores.

Como complemento visual, la Figura 4 muestra un ejemplo de trayectoria del TCP generada por la política residual y comparada con la referencia experta de cuRobo para la misma configuración inicial y pose objetivo. Donde se puede comparar se observa una trayectoria cartesiana próxima a la referencia experta.

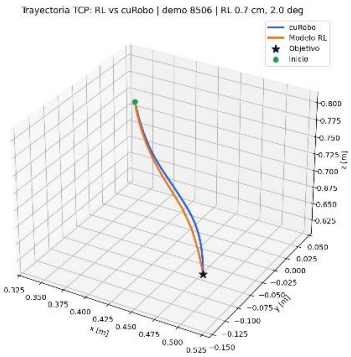


Figura 4: Comparación visual de trayectoria entre cuRobo y la política RL.

4.3. Análisis de modos de fallo

Dado el alto porcentaje relativo de éxito, una inspección episodio a episodio de los 14 casos fallidos (un 4,67 %) revela que el modo de fallo por inestabilidad terminal y micro-oscilaciones ha quedado prácticamente erradicado. El principal modo de fallo residual se concentra ahora exclusivamente en metas muy alejadas del *qhome* (posición inicial > 0,4 m), donde la trayectoria dictada por el BC no llega a converger dentro de los 480 pasos del episodio. En estos casos limítrofes, el error final suele ser de varios centímetros y decenas de grados, y se observan giros completos del cuaternión (errores de orientación de 150–180°) compatibles con configuraciones espejo del *end-effector*. Este modo de fallo es directamente abordable en futuras etapas mediante un BC entrenado con horizontes más largos o estructurando el aprendizaje con trayectorias multi-segmento.

5. Conclusiones y Trabajo Futuro

Se ha presentado un proyecto de tres fases para resolver tareas de alcance de pose en el manipulador colaborativo ABB GoFa dentro del simulador Isaac Lab. La aportación principal es mostrar que la combinación de un planificador GPU como experto, una política clonada con descriptores temporales y una pasada de DAgger, junto con un refuerzo residual acotado y anclado a la acción experta, permite construir rápidamente un controlador reactivo de articulaciones con una sobresaliente tasa de éxito del 95,33 % bajo umbrales estrictos (2 cm, 5,73°). Todo esto se logra sin necesidad de resolver cinemática inversa analítica en línea ni de exponer al robot a la fase costosa e inestable de exploración pura de RL.

Habiendo superado con éxito el desafío inicial de la estabilidad terminal (reduciendo los errores medios a 1,31 cm y 5,2° en la pose final) los resultados sugieren dos líneas inmediatas de mejora y avance. La primera es ampliar el horizonte del BC y reentrenar con trayectorias multi-segmento para cubrir eficazmente las metas residuales en los extremos del espacio de trabajo, donde la política actual agota los pasos disponibles antes de converger. Como complemento, se plantea realizar una evaluación comparativa más completa entre las distintas etapas de la arquitectura propuesta: BC inicial, BC corregido mediante DAgger y BC-DAgger refinado con PPO residual. Esta comparación permitirá cuantificar de forma aislada la contribución de cada fase sobre la tasa de éxito, el error final de posición y orientación, y la estabilidad terminal de la trayectoria. La segunda, sobre la que ya estamos trabajando, es la transferencia a la siguiente etapa de la hoja de ruta, denominada Gofa-AvoidReach-v0, en la que se añaden obstáculos dinámicos al entorno y la política debe combinar el alcance con la evitación reactiva.

Para esta próxima etapa será necesario sensar el obstáculo, ampliar la observación con su posición y reentrenar el BC sobre demostraciones cuRobo en escenas complejas. Por último, la transferencia al robot físico GoFa requerirá completar la cadena con un ajuste fino mediante DAgger en línea con el robot real como entorno y cuRobo como experto, manteniendo la política residual congelada o reduciendo aún más su escala durante la transferencia. Una vez en operación real, la integración de mecanismos de RL seguro será necesaria para garantizar el estricto cumplimiento de las normas ISO de colaboración humano-robot.

Agradecimientos

Este trabajo ha sido realizado parcialmente gracias al apoyo de la Comisión Europea en el marco del proyecto europeo MANiBOT, número de referencia 101120823.

Referencias

ABB Robotics, 2024. GoFa™ CRB 15000 Collaborative Robot – Product Specification. ABB AB. https://new.abb.com/products/robotics/collaborative-robots/crb-15000

Bayona, E., Sierra-García, J. E., Santos, M., Mariolis, I., 2024. In search of the best fitness function for optimum generation of trajectories for Automated Guided Vehicles. *Engineering Applications of Artificial Intelligence* 133, 108440. DOI: 10.1016/j.engappai.2024.108440

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., et al., 2023. RT-2: vision-language-action models transfer web knowledge to robotic control. En *Proceedings of the 7th Conference on Robot Learning (CoRL)*, PMLR 229, pp. 2165–2183.

Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., Song, S., 2023. Diffusion policy: visuomotor policy learning via action diffusion. En *Proceedings of Robotics: Science and Systems (RSS)*, Daegu, Corea del Sur. Versión extendida en *International Journal of Robotics Research* 44(10-11), 2025.

Dalal, M., Chiappa, A., Mandlekar, A., Garrett, C., Handa, A., Salakhutdinov, R., Fox, D., 2024. Local policies enable zero-shot long-horizon manipulation. *arXiv preprint* arXiv:2410.22332.

Deshpande, A., Ke, L., Pfeifer, Q., Gupta, A., Srinivasa, S., 2024. Data efficient behavior cloning for fine manipulation via continuity-based corrective labels. *arXiv preprint* arXiv:2405.19307.

Hu, H., Mirchandani, S., Sadigh, D., 2024. Imitation bootstrapped reinforcement learning. En *Proceedings of Robotics: Science and Systems (RSS)*, Delft, Países Bajos.

Johannink, T., Bahl, S., Nair, A., Luo, J., Kumar, A., Loskyll, M., Ojea, J. A., Solowjow, E., Levine, S., 2019. Residual reinforcement learning for robot control. En *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6023–6029. DOI: 10.1109/ICRA.2019.8794127

Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., et al., 2024. OpenVLA: an open-source vision-language-action model. *arXiv preprint* arXiv:2406.09246.

Liang, H., Cong, Y., Sun, G., 2024. Constrained behavior cloning for robotic learning. *arXiv preprint* arXiv:2408.10568.

MANiBOT, 2025. Advancing the physical intelligence and performance of roBOTs towards human-like bi-manual objects MANipulation. Proyecto Horizon Europe 101120823. https://manibot-project.eu/

Menéndez-García, A., Sierra-García, J. E., 2025. Aprendizaje por refuerzo para el cálculo del modelo cinemático inverso de un robot colaborativo. En *XX Simposio CEA de Control Inteligente*, Huelva, España.

Mittal, M., Roth, P., Tigue, J., Richard, A., et al., 2025. Isaac Lab: a GPU-accelerated simulation framework for multi-modal robot learning. NVIDIA Research. https://github.com/isaac-sim/IsaacLab

Octo Model Team, Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., et al., 2024. Octo: an open-source generalist robot policy. En *Proceedings of*

Robotics: Science and Systems (RSS), Delft, Países Bajos. arXiv:2405.12213.

Ross, S., Gordon, G. J., Bagnell, J. A., 2011. A reduction of imitation learning and structured prediction to no-regret online learning. En *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 627–635.

Sierra-García, J. E., Santos, M., 2021. Control of industrial AGV based on reinforcement learning. En *15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020)*, pp. 647–656. Springer.

Sundaralingam, B., Hari, S. K. S., Fishman, A., Garrett, C., Van Wyk, K., Blukis, V., Millane, A., Oleynikova, H., Handa, A., Ramos, F., Ratliff, N., Fox, D., 2023. cuRobo: parallelized collision-free minimum-jerk robot motion generation. *arXiv preprint* arXiv:2310.17274. https://curobo.org/

Thumm, J., Althoff, M., 2022. Provably safe deep reinforcement learning for robotic manipulation in human environments. En *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6344–6350. arXiv:2205.06311.