Control de brazo robótico Franka Emika 3 usando aprendizaje por refuerzo en MuJoCo

Proyecto de curso IIC3675 - Aprendizaje Reforzado. Pontificia Universidad Católica de Chile Departamento de Ciencias de la Computación

1st Diego Muñoz Rojas
Departamento de Ingeniería Eléctrica
Pontificia Universidad Católica de Chile
Santiago, Chile
dammr@uc.cl

2nd Anaís Montanares Valdés
Departamento de Ciencias de la Computación
Pontificia Universidad Católica de Chile
Santiago, Chile
aamontanares@uc.cl

***Abstract*—The implementation of advanced robotic manipulators in agricultural tasks represents a significant step toward the sustainable automation of harvesting processes. This work presents a reinforcement learning (RL)-based control system for the Franka Emika Research 3 (Fr3) robotic arm, aimed at replicating object collection tasks such as picking, placing, and moving in a simulation environment developed with MuJoCo. For this purpose, a simulation model derived from the Franka Research 3 and Franka Emika Panda robotic arms, provided by Google DeepMind in the MuJoCo Menagerie collection, was utilized. The system implements a Soft Actor-Critic (SAC) agent combined with Hindsight Experience Replay (HER) to optimize system performance, enhancing the precision and efficiency of the tasks performed.**

The simulation environment enables the extraction of key variables such as position, velocity, and actuator torque, which are essential for training the agent. The results highlight the proposed model's ability to adapt to complex dynamics and validate its potential in simulated environments, laying the groundwork for its transfer and evaluation in real hardware, thus paving the way for future automated agricultural applications.

***Palabras clave*—State space, uncertain load, constraints, robust predictive control, control algorithm, Kalman filter, optimization.**

Resumen—La implementación de manipuladores robóticos avanzados en tareas agrícolas representa un avance significativo hacia la automatización sostenible de procesos de recolección. Este trabajo presenta un sistema de control basado en aprendizaje por refuerzo (RL) para el brazo robótico Franka Emika Research 3 (Fr3), con el cual se busca replicar tareas de recolección de objetos, como tomar, dejar y mover, en un entorno de simulación desarrollado con MuJoCo. Para ello, se utilizó un modelo de simulación derivado de los brazos robóticos Franka Research 3 y Franka Emika Panda, proporcionados por Google DeepMind en la colección MuJoCo Menagerie. El sistema implementa un agente Soft Actor-Critic (SAC) combinado con Hindsight Experience Replay (HER) para optimizar el rendimiento del sistema, mejorando la precisión y la eficiencia de las tareas realizadas.

El entorno de simulación permite extraer variables clave como posición, velocidad y torque de los actuadores, necesarias para el entrenamiento del agente. Los resultados obtenidos destacan la capacidad del modelo propuesto para adaptarse a dinámicas complejas y validar su potencial en entornos simulados, lo que sienta las bases para su transferencia y evaluación en hardware real, abriendo la puerta a futuras aplicaciones agrícolas automatizadas.

Keywords—Aprendizaje reforzado, Soft Actor-Critic, Hindsight Experience Replay, manipuladores robóticos, control de brazos robóticos, simulación en MuJoCo, Franka Emika Research 3, automatización en manipulación de objetos.

I. Introducción

La robótica avanzada está transformando diversas industrias, incluyendo la agricultura, donde las tareas repetitivas y de alta precisión, como la recolección de frutos, enfrentan desafíos importantes. La integración de manipuladores robóticos en estas tareas no solo tiene el potencial de aumentar la eficiencia y reducir costos, sino también de abordar problemas asociados a la sostenibilidad, como el consumo energético y la optimización de recursos. Sin embargo, desarrollar sistemas de control adaptativos para estos manipuladores en entornos complejos sigue siendo un reto significativo debido a la dinámica compleja y estocasticidad del ambiente.

Este trabajo propone un sistema de control basado en aprendizaje por refuerzo para el brazo robótico Franka Emika Research 3, diseñado para realizar tareas como tomar, mover y dejar objetos en un entorno de simulación desarrollado con MuJoCo. La metodología combina un modelo de simulación realista derivado de los brazos robóticos Franka Research 3 y Franka Emika Panda, con un agente Soft Actor-Critic (SAC) que utiliza Hindsight Experience Replay (HER) para mejorar el aprendizaje en escenarios dinámicos. Este enfoque permite superar las limitaciones de los métodos tradicionales de control óptimo, ofreciendo un sistema más adaptable y eficiente.

La metodología combina los algoritmos Soft Actor-Critic (SAC), que mejora la estabilidad y la exploración en el aprendizaje, y Hindsight Experience Replay (HER), que optimiza el aprendizaje en entornos con recompensas escasas.

El entorno de simulación proporciona variables clave, como la posición, la velocidad y el torque de los actuadores, mientras que el agente se entrena en un bucle de episodios diseñado para alcanzar objetivos específicos antes de terminarlo. Para implementar el entrenamiento, se exploraron dos enfoques: un desarrollo de algoritmo propio y el uso de funciones prediseñadas proporcionadas por la librería stable_baselines, lo que permitió evaluar el rendimiento del agente bajo diferentes configuraciones y comparar la efectividad de ambas metodologías.

Los resultados demuestran el potencial del modelo para adaptarse a dinámicas complejas, validando su implementación en simulación y abriendo posibilidades para aplicaciones agrícolas automatizadas y sostenibles.

II. Estado del arte

La robótica avanzada ha permitido el desarrollo de manipuladores robóticos aplicados a diversas tareas, incluyendo la agricultura, donde

destacan actividades como la poda, la recolección de frutas y el manejo de objetos delicados. Estas aplicaciones requieren sistemas precisos y adaptativos que combinen grados de libertad elevados con estrategias de control robustas. Particularmente, el uso de manipuladores con uno o dos brazos ha sido un enfoque común, con aplicaciones que van desde la fumigación de cultivos hasta la manipulación de objetos deformables y la ejecución de tareas colaborativas.

En el ámbito del control de manipuladores robóticos, se han utilizado métodos tradicionales como la formulación de Newton-Euler y los multiplicadores de Lagrange para abordar problemas de cinemática y dinámica. Aunque estos métodos son fundamentales para derivar ecuaciones de movimiento y entender las dinámicas de sistemas robóticos, presentan limitaciones importantes cuando se enfrentan a restricciones no holonómicas o dinámicas complejas. Además, su implementación en aplicaciones en tiempo real suele ser impráctica debido al alto costo computacional asociado a la resolución de sistemas de ecuaciones no lineales y a la complejidad de integrar las restricciones algebraicas.

Alternativamente, metodologías como la formulación recursiva de Gibbs-Appel y las ecuaciones de Kane han demostrado ser especialmente eficaces para abordar sistemas con restricciones complejas y estructuras dinámicas avanzadas. La formulación recursiva de Gibbs-Appel, por ejemplo, es un método numérico que emplea un enfoque iterativo para calcular tanto la cinemática inversa como las ecuaciones de movimiento, optimizando así la eficiencia computacional en manipuladores con múltiples grados de libertad. Esta característica la hace adecuada para manipuladores con configuraciones redundantes o aplicaciones que exigen alta velocidad de procesamiento. Sin embargo, su implementación puede enfrentar desafíos, como una mayor susceptibilidad a singularidades y problemas de convergencia, lo que limita su eficacia en comparación con métodos analíticos más robustos en escenarios donde la precisión es crítica. [5]

Por otro lado, las ecuaciones de Kane, derivadas del principio de trabajo y energía virtual, han demostrado ser particularmente útiles para sistemas con restricciones no holonómicas. Estas ecuaciones permiten una representación más compacta y manejable de las dinámicas del sistema, reduciendo el número de cálculos necesarios en comparación con métodos como Lagrange. Además, su capacidad para trabajar con variables generalizadas y no generalizadas evita explícitamente el cálculo de fuerzas de reacción de restricción, lo que las hace computacionalmente más eficientes. Este enfoque permite incorporar las restricciones de manera más directa y sistemática, sin aumentar innecesariamente el número de ecuaciones y reduciendo la dimensionalidad del problema al centrarse en las variables independientes. [5]

Los manipuladores robóticos deben adaptarse a entornos altamente dinámicos y ejecutar tareas con un alto grado de precisión para consolidarse como herramientas efectivas en los procesos de producción. Los enfoques modernos han incorporado algoritmos de aprendizaje por refuerzo como Soft Actor-Critic, reconocidos por su capacidad para manejar espacios de acción continuos y optimizar políticas de control en escenarios complejos. Asimismo, técnicas complementarias como Hindsight Experience Replay (HER) han demostrado ser particularmente efectivas en entornos con recompensas escasas donde los objetivos son difíciles de alcanzar o altamente costosos, aspectos cruciales para aplicaciones agrícolas y otros sectores exigentes.

Paralelamente, los avances en simulación, como los proporcionados por el motor de física MuJoCo, han sido clave para el desarrollo de entornos virtuales realistas donde se puede probar y optimizar el rendimiento de los manipuladores. Estos simuladores, en combinación con herramientas como stable_baselines3, han facilitado la implementación de agentes de RL que integran controladores de espacio operativo (OSC) con algoritmos de aprendizaje. Este enfoque ha permitido abordar de manera efectiva problemas como la eficiencia energética, planificación de rutas, evitación de colisiones y la reducción de movimientos redundantes, mejorando la precisión y adaptabilidad de los manipuladores en escenarios agrícolas. [1]

En el ámbito de la robótica, se están integrando técnicas que originalmente se empleaban en aplicaciones industriales con altas tasas de muestreo, anteriormente limitadas por las capacidades computacionales. Sin embargo, los avances tecnológicos han permitido superar estas barreras, abriendo nuevas posibilidades. El Control Predictivo Basado en Modelos (MPC) ha emergido como una metodología efectiva para el control de manipuladores robóticos. Este enfoque permite anticipar y planificar movimientos al predecir el comportamiento futuro del sistema, optimizando las trayectorias y asegurando el cumplimiento de restricciones físicas y dinámicas. Estas capacidades hacen que el MPC sea particularmente valioso en aplicaciones donde la estabilidad y la adaptabilidad son esenciales. Además, su flexibilidad permite combinarlo con otras técnicas ampliando su aplicabilidad en tareas robóticas avanzadas.

A pesar de estos avances, los manipuladores robóticos aún enfrentan desafíos significativos en la transferencia de soluciones de simulación a sistemas reales, debido a diferencias en parámetros dinámicos y restricciones físicas. Sin embargo, investigaciones recientes han demostrado que la integración de algoritmos de aprendizaje profundo y técnicas de control adaptativo puede superar estas limitaciones, proporcionando sistemas más robustos y flexibles. Estas estrategias no solo han mejorado la eficiencia en aplicaciones agrícolas, sino que también han extendido su aplicabilidad a sectores como la industria alimentaria, la medicina, manufactura y la exploración espacial.

Este trabajo busca aportar al estado del arte mediante la implementación de un agente de RL basado en los algoritmos SAC y HER, con el objetivo de optimizar el control del manipulador robótico Franka Emika Research 3. La propuesta se valida en un entorno simulado desarrollado con el motor de física MuJoCo, sentando las bases para su futura transferencia a hardware real y su aplicación en entornos agrícolas automatizados.

III. Herramientas

El desarrollo del código se llevó a cabo utilizando Python 3.12.6, junto con las siguientes librerías y paquetes: MuJoCo 3.2.6, PyOpenGL 3.1.7, pygame 2.6.1, gymnasium 1.0.0, glfw 2.8.0, stable_baselines3 2.4.0, así como torch 2.5.1, torchvision 2.5.1 y torchaudio 0.20.1 con soporte para CUDA 11.8 para aprovechar las capacidades de procesamiento de la GPU.

El sistema de hardware empleado incluye Windows 10 Pro, un procesador AMD Ryzen 7 5800X3D de 8 núcleos a 3.4 GHz (arquitectura x64), 32 GB de memoria RAM, y una tarjeta gráfica NVIDIA RTX 4060 TI O8G.

También se utilizó GitHub como herramienta de desarrollo donde se proporciona el código en https://github.com/dammr54/ProyectoRL.

IV. Entorno de simulación

El entorno de simulación utilizado se basa en un modelo de simulación de MuJoCo, derivado de los modelos del brazo robótico Franka Research 3 y Franka Emika Panda, disponibles en la colección MuJoCo Menagerie proporcionada por Google DeepMind [2][3][4].

Del entorno se extraen las variables de estado de posición y velocidad, así como las variables de control relacionadas con el torque de los actuadores, las cuales son fundamentales para la implementación del algoritmo de aprendizaje por refuerzo. Una visualización del entorno se presenta en la Figura 1.

V. Algoritmos

El Soft Actor-Critic es un algoritmo de aprendizaje por refuerzo especialmente diseñado para abordar espacios de acción continuos, destacándose por promover políticas exploratorias de manera eficiente y estable. Este enfoque se basa en una arquitectura Actor-Critic, donde el actor genera acciones que maximizan simultáneamente el valor esperado de la función Q y la entropía, incentivando políticas con un grado óptimo de aleatoriedad. Este equilibrio permite a SAC

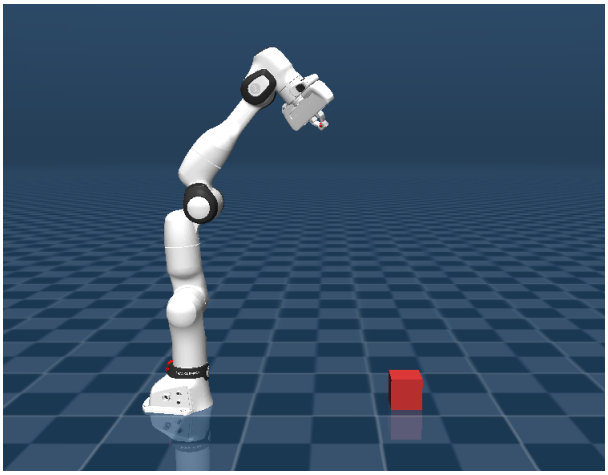

Fig. 1: Entorno de simulación de MuJoCo. Fuente propia

evitar la convergencia prematura hacia soluciones subóptimas, asegurando un balance efectivo entre exploración y explotación durante el proceso de aprendizaje.

Por su parte, los críticos actualizan sus estimaciones de la función Q utilizando una pérdida de error cuadrático medio, que mide la diferencia entre las predicciones actuales y las recompensas futuras descontadas. Para garantizar la estabilidad durante el entrenamiento, SAC emplea redes críticas objetivo que se actualizan de manera gradual. Este mecanismo suaviza los cambios en los parámetros, mitigando oscilaciones en las estimaciones y facilitando una convergencia más estable.

El algoritmo HER mejora significativamente la eficiencia del entrenamiento al redefinir las metas de episodios fallidos, transformándolos en experiencias útiles para el aprendizaje del agente. HER utiliza un enfoque retrospectivo, donde las metas originales son reemplazadas por estados efectivamente alcanzados durante la interacción del agente con el entorno. Al recalcular las recompensas en función de la proximidad del estado final a estas metas alternativas, el agente puede aprender estrategias para aproximarse a objetivos alcanzables. Este enfoque no solo aprovecha de manera más efectiva las experiencias previas, sino que también mejora la capacidad del agente para generalizar a diferentes metas y situaciones futuras.

VI. Implementación

En esta ocasión, empleamos dos enfoques diferentes para la implementación de SAC y HER. Primero, desarrollamos una implementación personalizada, diseñada específicamente para adaptarse al entorno de simulación MuJoCo. Segundo, utilizamos la implementación provista por la librería `stable-baselines3` [7].

La recompensa en cada paso (*step*) se calcula comparando la distancia actual del brazo al objetivo con la distancia registrada en el paso anterior considerando la distancia relativa del extremo efector a la meta. La recompensa se define como la diferencia entre la distancia previa y la actual: si el brazo se acerca al objetivo, la recompensa es positiva; si se aleja, la recompensa es negativa. A continuación, se presenta un pseudocódigo que describe esta función de recompensa:

Pseudocódigo para calcular la recompensa

Entradas: noitemsep, left=1em

- data: Datos del simulador
- target: Posición objetivo ($\mathbb{R}^3$)
- all_d: Lista con distancias previas

1) Obtener la posición del efector final:

$$position \leftarrow data.xpos[6]$$

2) Calcular la distancia actual al objetivo:

$$distance \leftarrow \|position - target\|$$

3) Verificar si all_d contiene valores:
   - **Si está vacío:**

$$last\_distance \leftarrow distance$$

   - **Si no está vacío:**

$$last\_distance \leftarrow all\_d[-1]$$

4) Agregar la distancia actual a la lista:

$$all\_d.append(distance)$$

5) Calcular el cambio en la distancia:

$$distance\_change \leftarrow last\_distance - distance$$

6) Asignar el valor de la recompensa:

$$reward \leftarrow distance\_change$$

7) Retornar:

$$reward$$

VII. Resultados

Comparamos el rendimiento de ambas implementaciones ejecutando nuestro experimento durante 50,000 rondas. Los resultados se presentan en la figura 3. Aunque los resultados obtenidos con nuestra implementación muestran mayor inestabilidad, en algunas ocasiones el brazo logra acercarse más al objetivo, superando el desempeño de la implementación de `stable-baselines3`.

Es importante destacar que, en la figura 3, la línea correspondiente a `stable-baselines3` parece estancarse. Sin embargo, al hacer un acercamiento a partir de los 6,000 pasos aproximadamente (fig. **??**), podemos observar que la distancia al objetivo en realidad disminuye, aunque de manera muy lenta.

En una segunda iteración de nuestro modelo, se realizaron modificaciones en la generación del replay buffer. En lugar de utilizar directamente el estado final, se implementó una versión acotada del estado alcanzado por el extremo efector. Además, la nueva recompensa se calculó en función de la distancia entre la meta alternativa generada y el objetivo final que se debía alcanzar. En este caso se pudieron observar mejoras significativas en la estabilidad de aprendizaje pareciendo más prometedor que la primera implementación si consideráramos más tiempo de entrenamiento (fig. 4). Se pudo dar cuenta de la importancia de definir correctamente el Replay Buffer para la convergencia de la política óptima.

Por otro lado, es importante destacar que se implementaron objetivos variables y aleatorios entre episodios para fomentar la adaptabilidad del agente y simular escenarios más realistas. Este enfoque permite que el agente desarrolle políticas generalizables, capaces de responder eficazmente a diferentes configuraciones del entorno y objetivos específicos. Además, al incorporar esta variabilidad, se busca reducir el riesgo de sobreajuste a un conjunto limitado

de metas, mejorando su desempeño en entornos dinámicos y con múltiples restricciones.

VIII. Conclusiones

En este trabajo se exploró la implementación del algoritmo SAC combinado con HER para el control de un manipulador robótico en un entorno de simulación desarrollado con MuJoCo.

A partir de los resultados obtenidos con la implementación de `stable-baselines3`, podemos concluir que, si el experimento se hubiera ejecutado durante un mayor número de pasos, es probable que el modelo hubiera convergido a una distancia más cercana al objetivo. Sin embargo, es importante destacar que la tasa de disminución de la distancia al objetivo se reduce progresivamente con el tiempo. Esto sugiere que, aunque el entrenamiento se hubiera prolongado, es poco probable que el modelo hubiera logrado alcanzar el objetivo de manera efectiva.

En contraste, nuestra primera implementación personalizada mostró un mejor rendimiento en ciertos casos, aunque presentó inestabilidades significativas. Estas podrían atribuirse a la falta de robustez en el diseño o a discrepancias en los parámetros utilizados, lo que limitó su capacidad para mantenerse cerca del objetivo de manera consistente. Estos hallazgos resaltan la importancia de ajustar cuidadosamente los hiperparámetros, así como de diseñar una función de recompensa bien calibrada y un correcto replay buffer, para mejorar el desempeño del modelo en tareas de control continuo.

En nuestra segunda implementación se observan diferencias favorables en cuanto a la estabilidad de aprendizaje, ya que la distancia disminuye progresivamente con una mejor tasa de aprendizaje. Aunque al igual que el resto de modelos implementados se puede observar que se estanca en un valor con un alto indice de error con respecto al punto objetivo.

Se pudo dar cuenta de que el uso de un entorno de simulación como MuJoCo fue fundamental para evaluar y refinar el sistema de manera eficiente antes de una posible implementación en hardware real. Esto pone de manifiesto el valor de las simulaciones avanzadas como herramientas esenciales para reducir costos y riesgos en el desarrollo de manipuladores robóticos para diversos entornos.

En resumen, los resultados obtenidos destacan los desafíos inherentes al uso del aprendizaje por refuerzo en problemas de manipulación robótica, donde resulta fundamental lograr un equilibrio entre una convergencia rápida y una estabilidad sostenida. Este trabajo proporciona un área de investigación orientadas a mejorar la integración y combinación de diversos modelos de aprendizaje en entornos con espacios de estados continuos, así como a facilitar la transferencia efectiva de estos modelos a aplicaciones prácticas en escenarios del mundo real.

References

[1] Garcés H., Torres M., Auat F. & Acuña D. (08/01/2024). Improving Operational Space Control for Robotic Manipulators by RL-based Redundancy Resolution. Revisado en https://repositorio.uc.cl/server/api/core/bitstreams/f1960cbc-c1a7-413b-8941-c5ae481f49ca/content

[2] Google DeepMind. (s.f.). Modelo Franka Emika Panda. MuJoCo Menagerie. GitHub. Revisado en https://github.com/google-deepmind/mujoco_menagerie

[3] Google DeepMind. (s.f.). Modelo Franka Research 3. MuJoCo Menagerie. GitHub. Revisado en https://github.com/google-deepmind/mujoco_menagerie

[4] Moreno J. Modelo Franka Research 3 dual. Laboratorio de Robótica y Automatización. Departamento de Eléctrica de la Pontificia Universidad Católica de Chile.

[5] Korayem, M. H., Shafei, A. M., & Seidi, E. (2014). Symbolic derivation of governing equations for dual-arm mobile manipulators used in fruit-picking and the pruning of tall trees. Computers and Electronics in Agriculture 105, 95 - 102. Revisado en http://dx.doi.org/10.1016/j.compag.2014.04.013

[6] Abbas, M., Narayan, J., & Dwivedy, S. K. (2023). A systematic review on cooperative dual-arm manipulators: modeling, planning, control, and vision strategies. International Journal of Intelligent Robotics and Applications. Revisado en https://doi.org/10.1007/s41315-023-00292-0

[7] Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-Baselines3: Reliable Reinforcement Learning Implementations. Journal of Machine Learning Research, 22(268), 1-8. Revisado en http://jmlr.org/papers/v22/20-1364.html

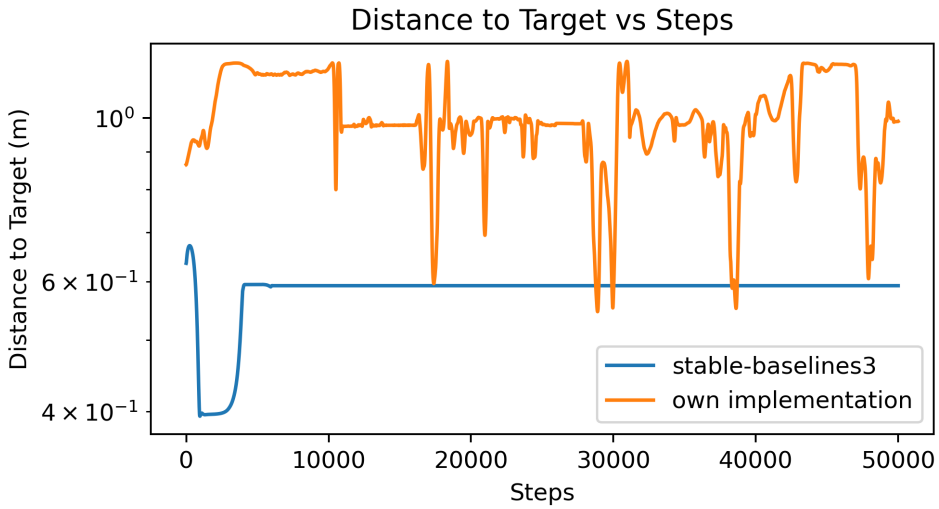

Fig. 2: Distancia del brazo al objetivo vs *steps* para dos implementaciones.

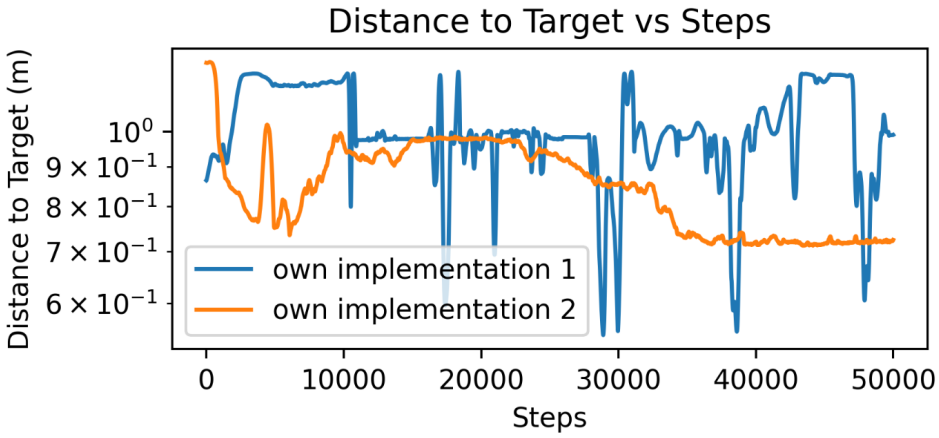

Fig. 3: Distancia del brazo al objetivo vs *steps* para dos implementaciones.

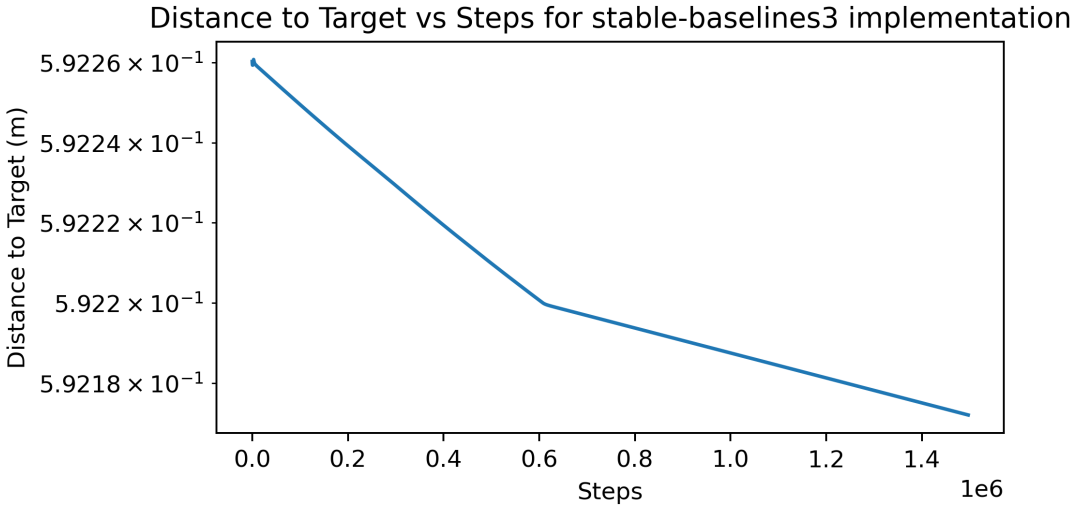

Fig. 4: Distancia del brazo al objetivo vs *steps* para implementación con `stable-baselines3`.