Aproximación de la cinemática inversa de los brazos de un robot Pepper mediante aprendizaje por refuerzo

Proyecto de Grado

Nicolás Rincón Sánchez

1 de julio del 2025

Introducción

Pepper es un robot social desarrollado por SoftBank Robotics, diseñado para la interacción humana y ampliamente utilizado en entornos comerciales y educativos.

Principales capacidades para tareas de:

- Navegación
- Percepción
- Comunicación
- Manipulación: Movimiento y operación de los brazos.

Problemática

En el campo de la robótica, la automatización es fundamental para asegurar que un robot sea capaz de completar satisfactoriamente las tareas para las cuales fue construído.

La automatización de las tareas de manipulación o, en general, de acomodación de posiciones forma parte del **problema de la cinemática inversa**, que se suele tratar con métodos numéricos.

- Limitaciones en escenarios con restricciones complejas o geometría redundante.
- Complejos para manipuladores con más de 2 grados de libertad.

Objetivos

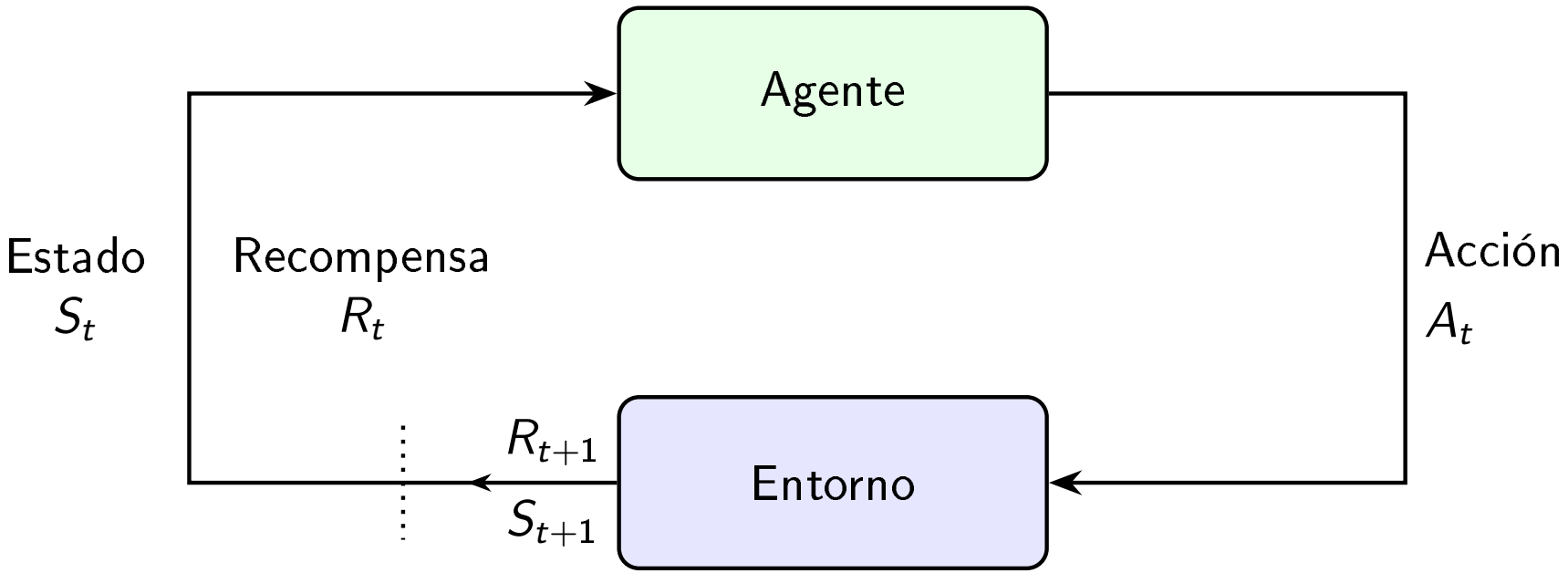
- **General:** Desarrollar una aproximación basada en Aprendizaje por Refuerzo que permita al robot Pepper aprender a resolver la cinemática inversa de sus brazos para alcanzar posiciones específicas en el espacio de trabajo.
- **Específicos:**
  - Modelar un entorno de simulación personalizado utilizando Gymnasium.
  - Diseñar una función de recompensa que guíe el aprendizaje del agente.
  - Implementar y entrenar algoritmos de Aprendizaje por Refuerzo para optimizar las políticas del agente dentro del entorno simulado.
  - Evaluar el desempeño y la capacidad de generalización de las políticas aprendidas.

Aprendizaje por Refuerzo (RL)

Agente que interactúa con un entorno y le cambia su estado.

El agente aprende las mejores acciones maximizando la recompensa que recibe del entorno.

Los escenarios se modelan como **Procesos de Decisión de Markov**.

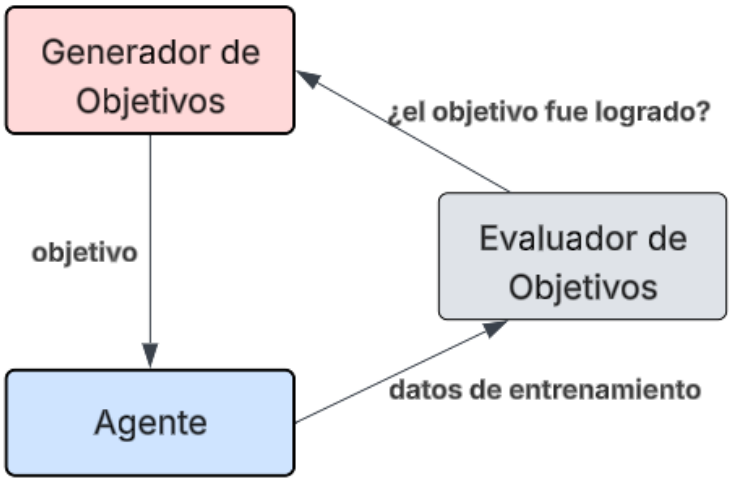


Aprendizaje Curricular

Se basa en la idea de realizar un ordenamiento de los datos o proceso de entrenamiento para que este suceda análogo a como aprenden los seres humanos: **de lo más fácil a lo más difícil**.

Para RL, consiste en un entrenamiento en niveles de dificultad crecientes.

Técnica de Generación de objetivos automática



Modelos de interés

- **Tarea de Manipulación / Posicionamiento**
  - Se representa como un Proceso de Decisión de Markov finito
  - Representable como una 6-tupla

$$M_i = (S_i, A, R_i, T_i, \gamma, \tau_i)$$

  - Objetivo principal: aprendizaje de una política de habilidad parametrizable y aplicable en un controlador intermedio.
- **Problema de Cinemática Inversa**
  - Dada una meta x^* para el efector final, se busca el vector de ángulos articulares q tal que $f(q) = x^*$, donde f es la función cinemática directa.
  - Equivalentemente, encontrar una solución del vector $(q_1, q_2, ..., q_n)$ que satisfaga la ecuación:

$$T_n^0(q_1, q_2, ..., q_n) = A_1(q_1)A_2(q_2)...A_n(q_n) = \begin{bmatrix} R_n^0 & o_n^0 \\ 0 & 1 \end{bmatrix}$$

Planteamiento del problema

- Cada brazo representado como un agente con su rango de variables en espacio de observación y propia representación como MDP.
- **Espacio de Observación**:

Variable	Descripción
θ_1	Valor del ángulo en radianes para la articulación *ShoulderPitch*
θ_2	Valor del ángulo en radianes para la articulación *ShoulderRoll*
θ_3	Valor del ángulo en radianes para la articulación *ElbowYaw*
θ_4	Valor del ángulo en radianes para la articulación *ElbowRoll*
θ_5	Valor del ángulo en radianes para la articulación *WristYaw*
e_x	Distancia entre posiciones x_{actual} y x_{goal}
e_y	Distancia entre posiciones y_{actual} y y_{goal}
e_z	Distancia entre posiciones z_{actual} y z_{goal}

- **Espacio de Acción**: Movimiento de cada ángulo dentro del rango $[-0{,}50, 0{,}50]$ rad

Diseño de la Función de Recompensa

La **Función de Recompensa** se definió como: $r_n(s) = \sum_k R_n^k$.

- Mejoramiento del error:

$$R_n^{\text{improvement}} = 30 \cdot (d_{n-1} - d_n) \tag{1}$$

- Penalización constante del error:

$$R_n^{\text{proximity}} = -2{,}0 \cdot d_n = -2{,}0\sqrt{e_x^2 + e_y^2 + e_z^2} \tag{2}$$

- Suavidad de los movimientos del brazo:

$$R_n^{\text{smoothness}} = -0{,}15 \cdot |\Delta\theta|^2 \tag{3}$$

Diseño de la Función de Recompensa

La **Función de Recompensa** se definió como: $r_n(s) = \sum_k R_n^k$.

- Penalización por alcanzar los límites de alguno de los ángulos:

$$R_n^{\text{limit}} = \begin{cases} -0{,}75 & \text{si } \theta_i = \theta_{i,\text{mín}} \vee \theta_i = \theta_{i,\text{máx}} \quad \forall i \in \{1,2,3,4,5\} \\ 0 & \text{de lo contrario} \end{cases} \tag{4}$$

- Recompensa especial por alcanzar una posición final con un error menor a 2cm:

$$R_n^{\text{success}} = \begin{cases} 25{,}0 & \text{si } d_n \leq 0{,}02 \\ 0 & \text{de lo contrario} \end{cases} \tag{5}$$

Manejo del Currículo

El **Radio de Currículo** define una zona esférica alrededor del objetivo dentro de la cual se inicializa la posición del efector final al comienzo de cada episodio de entrenamiento.

El valor del radio del currículo c por cada nivel k viene dado por:

$$c_k = \text{mín}(c_{k-1} + \Delta c, c_{\text{máx}})$$

donde $\Delta c = 0{,}1$ y $c_{\text{máx}} = 0{,}51$, expresados en metros.

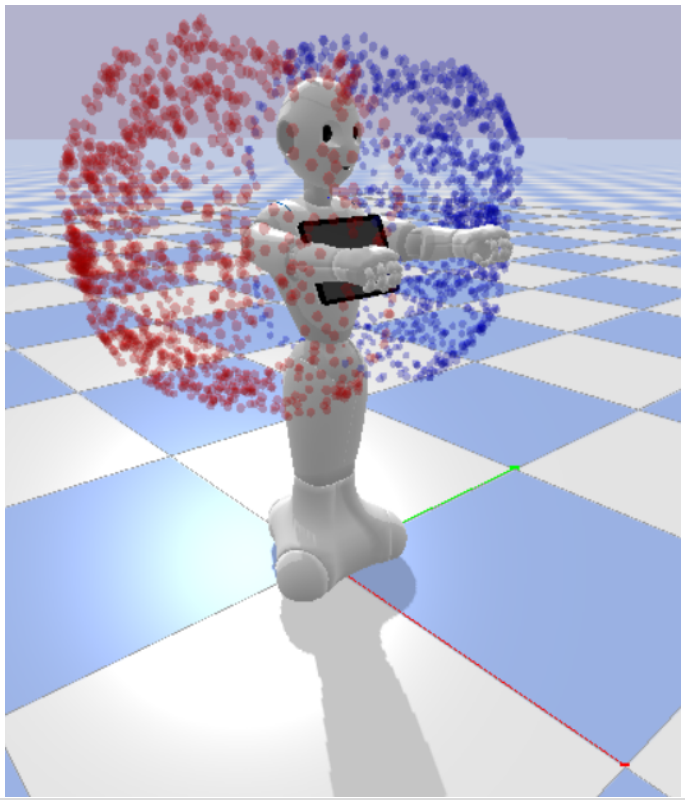
Simulación con qiBullet

Simulador que replica la física del entorno y el control de movimiento del robot Pepper.

Además, contempla perturbaciones aleatorias en la simulación del movimiento de las articulaciones.

Utilizado para:

- Configuración del entorno simulado
- Visualización del espacio de trabajo
- Visualización de pruebas



Sintonización de Hiperparámetros

Se realizaron estudios de optimización de hiperparámetros con Optuna para determinar los mejores parámetros para los dos algoritmos seleccionados: PPO y SAC.

Mejores estudios de Optuna para el brazo izquierdo

`PPO-analytical-3`

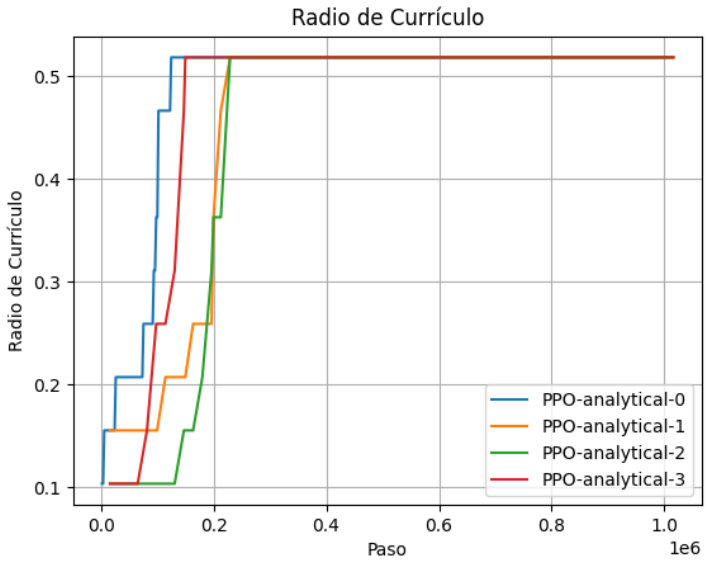
Hiperparámetro	Valor
batch_size	128
clip_range	0.2560204063533433
ent_coef	1.5204688692198897e-06
gae_lambda	0.9258792340272566
gamma	0.9258521201618417
learning_rate	7.591104805282687e-05
n_steps	2048
vf_coef	0.5920392514089517

`SAC-analytical-1`

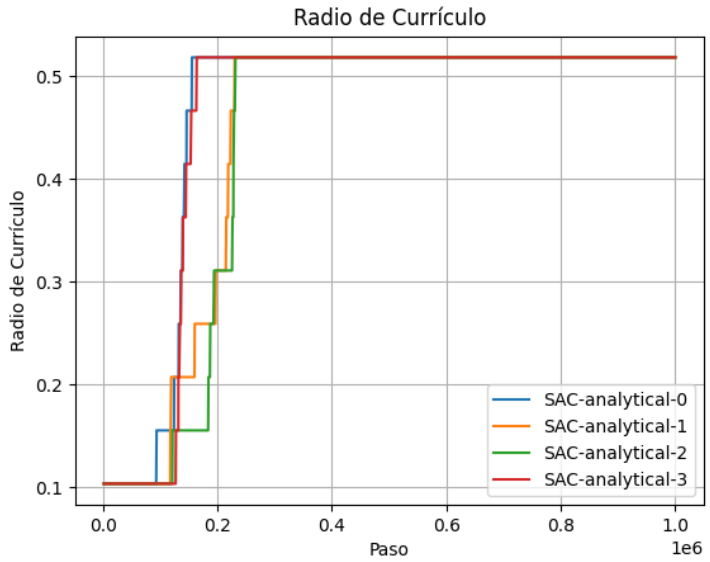
Hiperparámetro	Valor
batch_size	512
buffer_size	300 000
gamma	0.9199474108376201
gradient_steps	8
learning_rate	7.309539835912905e-05
tau	0.04033291826268561
train_freq	16

Resultados de Entrenamiento

Las figuras mostradas comparan el desempeño de los algoritmos en entrenamiento para el brazo izquierdo



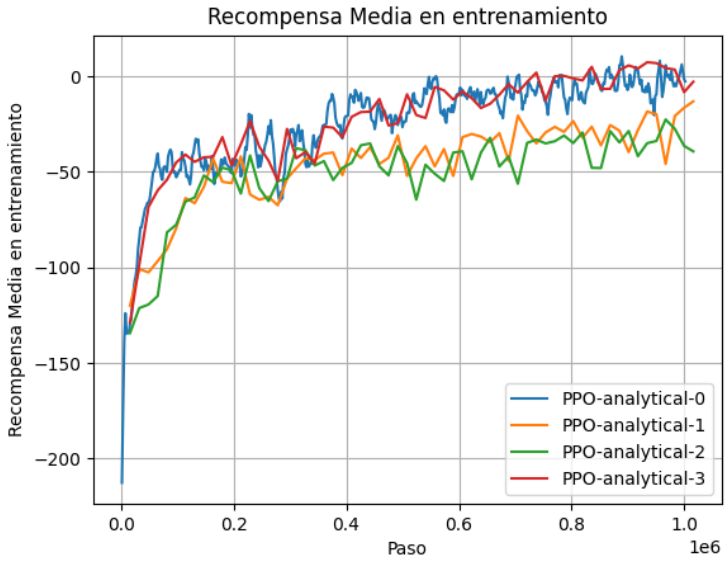
(a) Entrenamiento con Algoritmo PPO



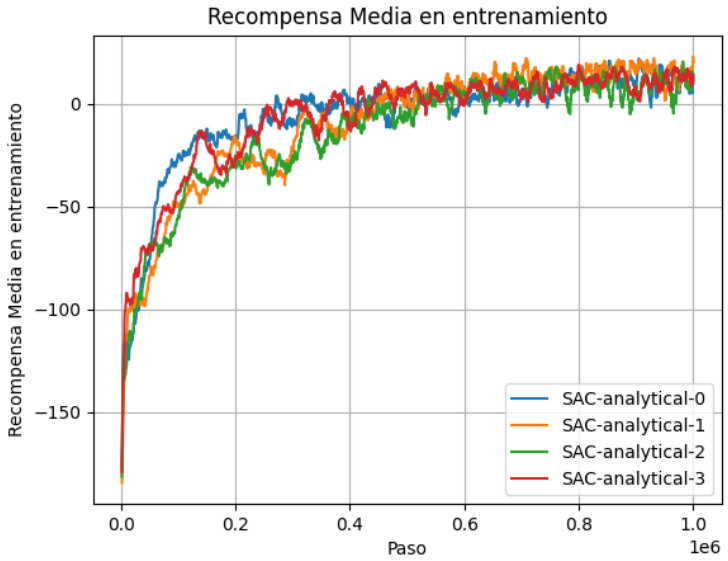
(b) Entrenamiento con Algoritmo SAC

Resultados de Entrenamiento

Las figuras mostradas comparan el desempeño de los algoritmos en entrenamiento para el brazo izquierdo



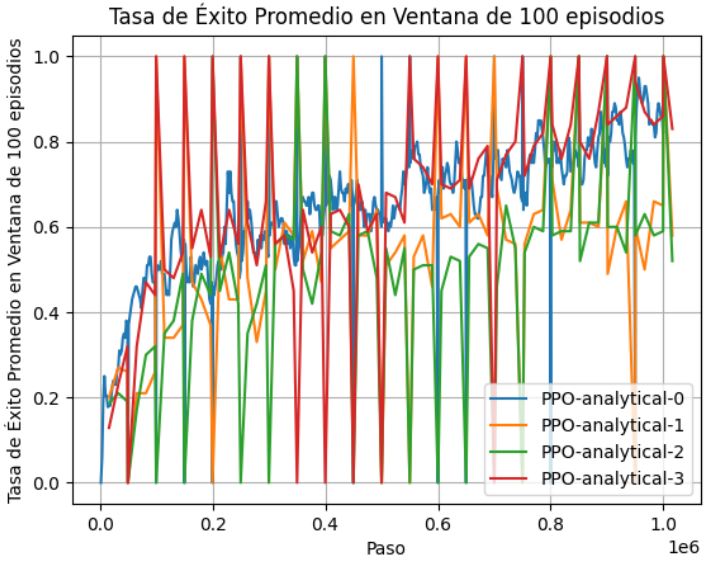
(a) Entrenamiento con Algoritmo PPO



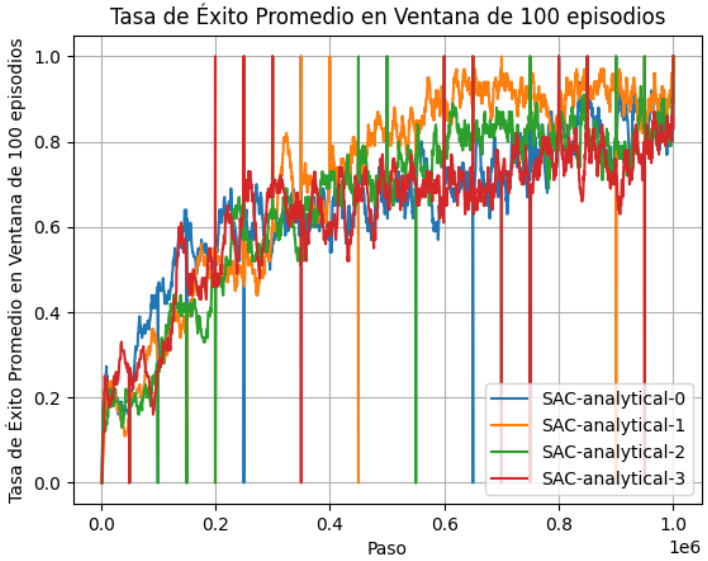
(b) Entrenamiento con Algoritmo SAC

Resultados de Entrenamiento

Las figuras mostradas comparan el desempeño de los algoritmos en entrenamiento para el brazo izquierdo



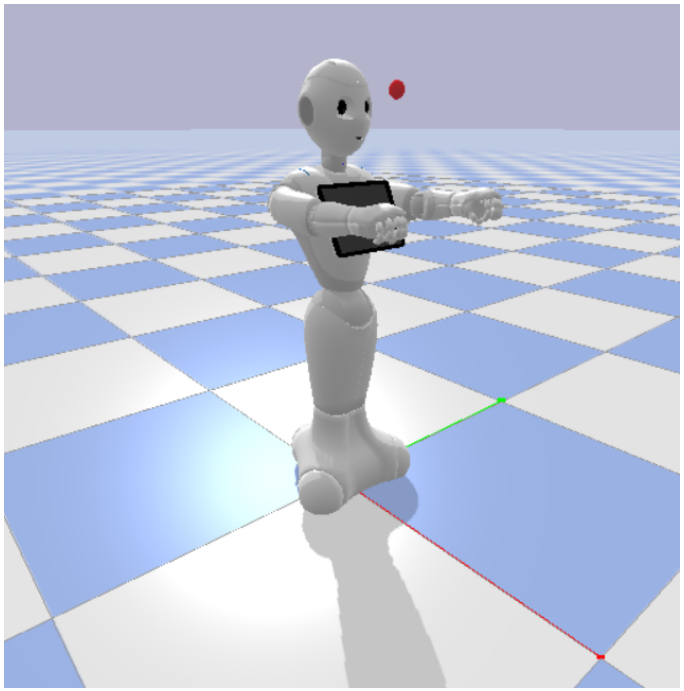
(a) Entrenamiento con Algoritmo PPO



(b) Entrenamiento con Algoritmo SAC

Pruebas y Validación

- Las pruebas de validación se realizaron sobre la interfaz del simulador seleccionando puntos del espacio de trabajo muestreado al azar.
- Por cada brazo, se realizaron 1000 pruebas en total para 3 umbrales de éxito para la cercanía a la posición final.



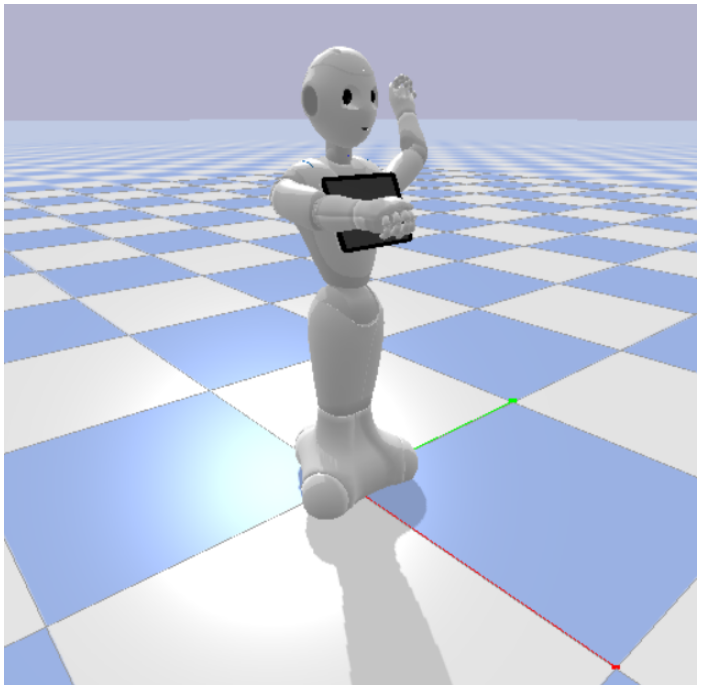
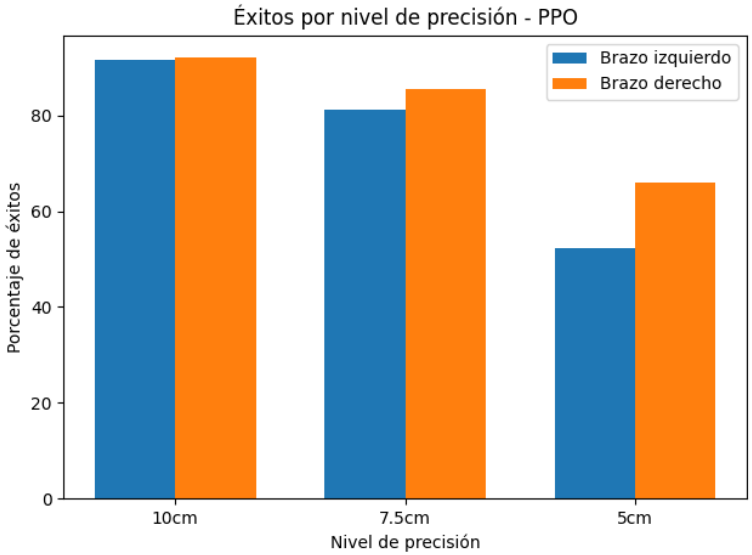
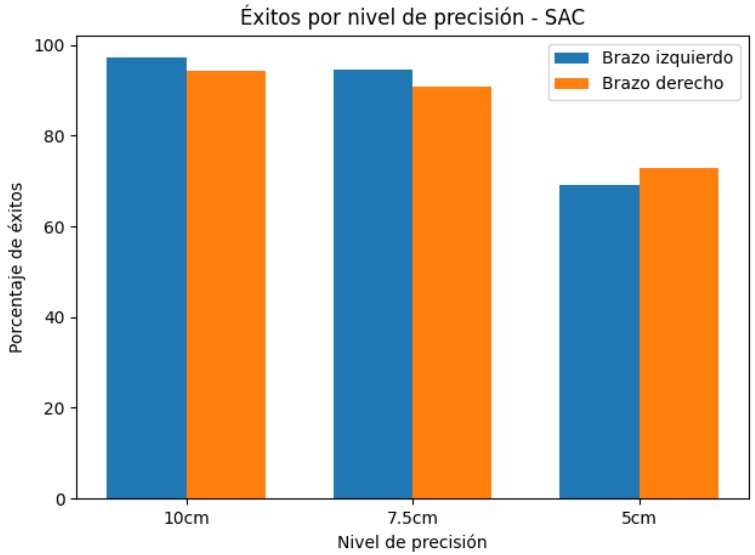


Figura: El punto rojo muestra la posición deseada representada en el sistema de coordenadas global: en este caso, el punto $(x, y, z) = (-0{,}22, 0{,}41, 1{,}16)$ m.

Pruebas y Validación

Algoritmo	10 cm	7.5 cm	5 cm
PPO (izq)	91.6 %	81.1 %	52.3 %
PPO (der)	92.1 %	85.6 %	66.0 %
SAC (izq)	97.2 %	94.5 %	69.2 %
SAC (der)	94.3 %	90.8 %	72.8 %





Conclusiones

- RL con currículo produjo políticas de control capaces de alcanzar con precisión y robustez una posición objetivo en un espacio de trabajo de cinco grados de libertad.
- SAC superó a PPO en métricas fundamentales de entrenamiento como la tasa de éxito y la recompensa final promedio.
- Las combinaciones de hiperparámetros seleccionadas conducen a comportamientos estables y reproducibles, satisfaciendo el requisito de un agente capaz de adaptarse a variaciones en la posición inicial y a posibles perturbaciones del entorno.

Referencias I

[1] O. Kroemer, S. Niekum y G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *Journal of Machine Learning Research*, vol. 22, n.º 1, págs. 1-82, 2021, https://www.jmlr.org/papers/volume22/19-804/19-804.pdf.

[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford y O. Klimov, *Proximal Policy Optimization Algorithms*, 2017. dirección: https://arxiv.org/pdf/1707.06347.

[3] T. Haarnoja, A. Zhou, P. Abbeel y S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," *arXiv preprint arXiv:1801.01290*, 2018. DOI: 10.48550/arXiv.1801.01290. dirección: https://arxiv.org/abs/1801.01290.

Referencias II

[4] SoftBank Robotics, *Pepper: The humanoid robot (Product Documentation)*, 2023. dirección: https://aldebaran.com/en/pepper/.

[5] R. S. Sutton y A. G. Barto, *Reinforcement Learning: An Introduction*, 2.ª ed. MIT Press, 2018.

[6] N. Bäuerle y U. Rieder, "Markov Decision Processes," *Jahresbericht der Deutschen Mathematiker-Vereinigung*, vol. 112, n.º 4, págs. 217-243, 2010, ISSN: 1869-7135. DOI: 10.1365/s13291-010-0007-2. dirección: https://doi.org/10.1365/s13291-010-0007-2.

[7] N. Nekamiche, "Curriculum Learning," Publicado en Medium, AIGuys (Medium), visitado 23 de jun. de 2025. dirección: https://medium.com/aiguys/curriculum-learning-83b1b2221f33.

Referencias III

[8] R. Ozalp, A. Ucar y C. Guzelis, "Advancements in Deep Reinforcement Learning and Inverse Reinforcement Learning for Robotic Manipulation: Toward Trustworthy, Interpretable, and Explainable Artificial Intelligence," *IEEE Access*, vol. 12, págs. 51 840-51 858, 2024. DOI: 10.1109/ACCESS.2024.3385426.

[9] L. Weng, "Curriculum for Reinforcement Learning," *lilianweng.github.io*, ene. de 2020. dirección: https://lilianweng.github.io/posts/2020-01-29-curriculum-rl/.

[10] A. Malik, Y. Lischuk, T. Henderson y R. Prazenica, "A Deep Reinforcement-Learning Approach for Inverse Kinematics Solution of a High Degree of Freedom Robotic Manipulator," *Robotics*, vol. 11, n.º 2, 2022, ISSN: 2218-6581. DOI: 10.3390/robotics11020044. dirección: https://www.mdpi.com/2218-6581/11/2/44.

Referencias IV

[11] R. Liu, F. Nageotte, P. Zanne, M. de Mathelin y B. Dresp-Langley, "Deep Reinforcement Learning for the Control of Robotic Manipulation: A Focussed Mini-Review," *Robotics*, vol. 10, n.º 1, pág. 22, 2021. DOI: 10.3390/robotics10010022. dirección: https://doi.org/10.3390/robotics10010022.

[12] H. El-Hussieny, "Lecture 7: Inverse Kinematics," Apuntes de curso, Faculty of Engineering, Shoubra, Benha University. dirección: https://bu.edu.eg/portal/uploads/Engineering,%20Shoubra/Electrical%20Engineering/823/crs-14135/Files/lecture7_InverseKinematics.pdf.

[13] C. Zhao, Y. Wei, J. Xiao y et al., "Inverse kinematics solution and control method of 6-degree-of-freedom manipulator based on deep reinforcement learning," *Scientific Reports*, vol. 14, pág. 12467, 2024. DOI: 10.1038/s41598-024-62948-6. dirección: https://doi.org/10.1038/s41598-024-62948-6.

Referencias V

[14] P. Adjei, N. Tasfi, S. Gomez-Rosero y M. A. M. Capretz, "Safe Reinforcement Learning for Arm Manipulation with Constrained Markov Decision Process," *Robotics*, vol. 13, n.º 4, pág. 63, 2024. DOI: 10.3390/robotics13040063. dirección: https://doi.org/10.3390/robotics13040063.

[15] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus y N. Dormann, "Stable-Baselines3: Reliable Reinforcement Learning Implementations," *Journal of Machine Learning Research*, vol. 22, n.º 268, págs. 1-8, 2021.

[16] T. Akiba, S. Sano, T. Yanase, T. Ohta y M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," en *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[17] M. Busy y M. Caniot, "qiBullet, a Bullet-based simulator for the Pepper and NAO robots," *arXiv preprint arXiv:1909.00779*, 2019.

Referencias VI

[18] M. Towers et al., *Gymnasium: A Standard Interface for Reinforcement Learning Environments*, 2024. arXiv: 2407.17032 [cs.LG]. dirección: https://arxiv.org/abs/2407.17032.