Trabajo Fin de Grado
en Ingeniería de las Tecnologías Industriales

Simulación y Control de Manipulador Robótico Aéreo Bi-Brazo mediante Aprendizaje por Refuerzo Transferible a Microgravedad

Autor: Pablo Revuelto de Miguel

Tutor: Alejandro Suárez Fernández-Miranda

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2026

Trabajo Fin de Grado
en Ingeniería de las Tecnologías Industriales

Simulación y Control de Manipulador Robótico Aéreo Bi-Brazo mediante Aprendizaje por Refuerzo Transferible a Microgravedad

Autor:

Pablo Revuelto de Miguel

Tutor:

Alejandro Suárez Fernández-Miranda

Profesor Titular

Dpto. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2026

Trabajo Fin de Grado: Simulación y Control de Manipulador Robótico Aéreo Bi-Brazo mediante Aprendizaje por Refuerzo Transferible a Microgravedad

Autor: Pablo Revuelto de Miguel

Tutor: Alejandro Suárez Fernández-Miranda

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2026

El Secretario del Tribunal

Agradecimientos

A mi familia, amigos y quien siempre estuvo conmigo en este camino.

Este Trabajo Fin de Grado se ha realizado en el contexto de la Red Europea de Robótica e Inteligencia Artificial (euROBIN)

Resumen

Este Trabajo de Fin de Grado aborda el desafío del control autónomo en sistemas de manipulación aérea mediante técnicas de Aprendizaje por Refuerzo Profundo (en inglés Deep Reinforcemente Learning (DRL)). La integración de brazos robóticos en vehículos aéreos altera drásticamente la dinámica del sistema debido a los continuos desplazamientos del centro de masa y la inyección de pares de reacción sobre la base flotante. Para resolver esta complejidad, se ha modelado un cuadrotor equipado con dos brazos robóticos dentro del motor físico MuJoCo. El objetivo central ha consistido en lograr que el agente ejecute una maniobra de aproximación y agarre de precisión sobre una carga útil (una caja), operando tanto en condiciones de gravedad terrestre (1g) como en un entorno de microgravedad simulada (0g).

Frente a las limitaciones del control analítico clásico en escenarios no lineales, el núcleo metodológico se ha basado en el entrenamiento de redes neuronales mediante el algoritmo Proximal Policy Optimization (PPO). El desarrollo ha exigido el diseño de arquitecturas de actuación específicas para cada medio —un control jerárquico en cascada para el vuelo atmosférico y un control omnidireccional de fuerzas para el vacío— junto con un trabajo de ingeniería de la función de recompensa. Gracias a la implementación de penalizaciones dinámicas y restricciones geométricas virtuales, se lograron erradicar problemas conductuales intrínsecos del aprendizaje, tales como colisiones a alta velocidad por falta de frenado aerodinámico o giros descontrolados al extender los brazos manipuladores.

Los resultados empíricos extraídos de la simulación demuestran la convergencia hacia políticas de control robustas y mecánicamente eficientes en ambos regímenes dinámicos. En la Tierra, el agente logró estabilizar la actitud y operar con un esfuerzo de control mínimo, coordinando el vuelo estacionario con la cinemática de los brazos para ejecutar una aproximación segura. En el espacio, el modelo demostró su capacidad para interiorizar la conservación del momento, aprendiendo de forma autónoma a aplicar retropropulsión para detenerse y pares compensatorios anticipados para anclar la plataforma en el vacío. En su conjunto, el proyecto muestra la viabilidad del DRL continuo, al menos en simulación, para la manipulación aérea, y abre una vía directa hacia su futura implementación en robótica de servicio y exploración espacial.

Abstract

The project focuses on the challenge of autonomous control in aerial manipulation systems using Deep Reinforcement Learning (DRL) techniques. The integration of robotic arms into aerial vehicles drastically alters system dynamics due to continuous shifts in the center of mass and the injection of reaction torques into the floating base. To address this complexity, a quadrotor equipped with two robotic arms was modeled within the MuJoCo physics engine. The central objective was to enable the agent to perform a precise approach-and-grasp maneuver on a payload (a box), operating both under Earth gravity (1g) and in a simulated microgravity environment (0g).

Given the limitations of classical analytical control in highly nonlinear scenarios, the methodological core relied on training neural networks using the Proximal Policy Optimization (PPO) algorithm. The development required designing specific actuation architectures for each environment —a cascaded hierarchical controller for atmospheric flight and an omnidirectional force controller for vacuum conditions— along with extensive engineering of the reward function. Through the implementation of dynamic penalties and virtual geometric constraints, the system successfully eliminated behavioral problems inherent to learning, such as highspeed collisions due to the absence of aerodynamic braking or uncontrolled rotations when extending the robotic arms.

Empirical results extracted from simulation demonstrate convergence toward robust and mechanically efficient control policies in both dynamic regimes. On Earth, the agent stabilized its attitude and operated with minimal control effort, coordinating hover flight with arm kinematics to execute a safe approach. In space, the model showed its ability to internalize momentum conservation, autonomously learning to apply retropropulsion to stop and anticipatory compensatory torques to anchor the platform in vacuum. Overall, the project shows the viability of continuous DRL —at least in simulation— for aerial manipulation and opens a direct path toward its future implementation in service robotics and space exploration.

Índice

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Notación

DRL	Deep Reinforcement Learning (aprendizaje profundo por refuerzo)
UAV	Unmanned Aerial Vehicle (vehículo aéreo no tripulado)
UAM	Unmanned Aerial Manipulation System (vehículo aéreo de manipulación no tripulado
RL	Reinforcement Learning (aprendizaje por refuerzo)
DoF	Degrees of Freedom (grados de libertad)
CoM	Center of mass (centro de masa)
MDP	Markov Decision Process (proceso de decisión de Markov)
PPO	Proximal Policy Optimization (optimización proximal de políticas)
PWM	Pulse Width Modulation (modulación por ancho de pulso)

1 INTRODUCCIÓN

La robótica aérea ha experimentado un crecimiento exponencial en la última década, transformando sectores que van desde la inspección de infraestructuras hasta la agricultura de precisión. Sin embargo, la próxima frontera en este campo no se limita a la observación pasiva, sino que busca la interacción física activa con el entorno. En este contexto, el presente Trabajo de Fin de Grado (TFG) aborda el diseño, simulación y control de un sistema de manipulación aérea mediante técnicas de Aprendizaje por Refuerzo Profundo (Deep Reinforcement Learning, DRL).

1.1 Contexto y motivación

Los Vehículos Aéreos No Tripulados, UAV (del inglés Unmanned Aerial Vehicle), comúnmente conocidos como drones, han demostrado ser plataformas versátiles y eficientes para tareas de vigilancia y teledetección. No obstante, existe una demanda creciente en la industria y en operaciones de búsqueda y rescate para dotar a estos robots de capacidades de manipulación. Surge así el campo de la manipulación aérea [1], que consiste en acoplar brazos robóticos u otros efectores finales a una plataforma aérea, permitiendo al sistema agarrar, transportar y ensamblar objetos en lugares de difícil acceso o peligrosos para el ser humano.

La motivación principal para el desarrollo de Sistemas de Manipulación Aérea No Tripulados (UAMS, por sus siglas en inglés) reside en su capacidad para operar en entornos inaccesibles, peligrosos o no estructurados para el ser humano o para la robótica terrestre convencional. Las aplicaciones potenciales son variadas: desde el transporte de cargas en logística de última milla y el ensamblaje de estructuras en altura, hasta la inspección por contacto de infraestructuras críticas (como puentes o líneas de alta tensión) y operaciones de búsqueda y rescate en zonas de desastre.

Uno de los desafíos principales de la manipulación aérea radica en la compleja dinámica del sistema acoplado. A diferencia de un manipulador industrial fijo al suelo, un manipulador aéreo es un sistema de base flotante. El movimiento de los brazos robóticos desplaza el centro de masa del sistema y genera pares de reacción que perturban la estabilidad del vuelo del dron. Tradicionalmente, este problema se ha abordado mediante teoría de control clásica (PID, LQR, control adaptativo), lo cual requiere un modelado matemático complejo (para el caso de esquemas de control centralizados) y una sintonización compleja de ganancias para contrarrestar las perturbaciones no lineales.

En años recientes, el aprendizaje por refuerzo (Reinforcement Learning, RL) ha emergido como una alternativa prometedora. En lugar de programar explícitamente las leyes de control, el RL permite que un agente aprenda una política de control óptima a través de la interacción prueba-error con un entorno simulado. Esta capacidad para manejar dinámicas complejas y no linealidades sin necesidad de un modelo analítico perfecto motiva el uso de esta técnica en el presente proyecto.

1.2 Definición del problema

El problema central que aborda este trabajo es el control autónomo de un sistema de manipulación aérea híbrido para realizar una tarea de agarre de precisión.

El sistema robótico, denominado en este proyecto como la integración de la plataforma aérea "multirotor X4" (un cuadrotor) y el sistema de manipulación "LiCAS A1" (compuesto por dos brazos robóticos antropomórficos), presenta un alto número de grados de libertad (DoF, por sus siglas en inglés). El control debe gestionar simultáneamente la estabilidad del vuelo del cuadrotor (posición y orientación) y la cinemática de los brazos para posicionar los efectores finales.

La tarea específica consiste en aproximar el dron a una caja objetivo situada sobre una estructura elevada (una caja base de mayor tamaño), insertar dos ganchos en forma de L (efectores finales) debajo del asa de la caja objetivo y realizar una maniobra de elevación estable. Este escenario plantea múltiples dificultades técnicas:

1. **Acoplamiento dinámico:** el movimiento de los brazos LiCAS A1 afecta directamente a la actitud del multi-rotor, requiriendo una compensación activa del empuje y los torques de los motores.

2. **Precisión geométrica:** los ganchos en forma de L requieren una alineación precisa en posición y orientación (Yaw) respecto al asa de la caja. Un error milimétrico puede provocar colisiones que desestabilicen el sistema o el fracaso del agarre.

3. **Seguridad y restricciones:** el agente debe evitar colisiones con la caja base sobre la que reposa el objetivo, aprendiendo a aproximarse con la altitud y ángulo adecuados.

4. **Simulación de alta fidelidad:** para que el aprendizaje sea efectivo, es necesario un entorno físico realista. Se utiliza el motor de física **MuJoCo** en un entorno Linux (Ubuntu 24.04 bajo WSL) para simular las interacciones de contacto, fricción y dinámica de cuerpos rígidos.

1.3 Objetivos del trabajo

El objetivo principal de este TFG es desarrollar, entrenar y validar un modelo de control basado en aprendizaje por refuerzo profundo (Deep Reinforcement Learning (DLR)) que permita a un manipulador aéreo (multi-rotor + LiCAS A1) realizar el agarre autónomo de una carga en un entorno simulado bajo gravedad terrestre.

Para alcanzar este propósito general, se definen los siguientes objetivos específicos:

- **Modelado y simulación del sistema:** integrar los modelos físicos del multi-rotor y los brazos LiCAS A1 en un entorno de simulación MuJoCo, definiendo correctamente las propiedades inerciales, los límites de los actuadores y las geometrías de colisión.

- **Diseño del entorno de aprendizaje (Gymnasium):** desarrollar un entorno compatible con la interfaz Gymnasium que gestione las observaciones del agente (estado del dron y brazos), las acciones y la lógica de simulación. En posteriores capítulos se hablará sobre él.

- **Ingeniería de la función de recompensa (Reward Shaping):** esta parte es fundamental. Diseñar una función de recompensa robusta que guíe al agente a través de las distintas fases de la tarea: navegación y aproximación al objetivo, alineación de ganchos, agarre y elevación, penalizando comportamientos indeseados como inestabilidad, movimientos bruscos o colisiones.

- **Implementación de estrategias de entrenamiento avanzadas:** aplicar técnicas como *Curriculum Learning* (incremento progresivo de la dificultad de aparición) y penalizaciones no lineales para optimizar la convergencia del algoritmo PPO (Proximal Policy Optimization).

- **Validación y análisis de resultados:** evaluar el desempeño del modelo entrenado mediante métricas de éxito y análisis visual de los episodios, utilizando un flujo de trabajo híbrido entre simulación local y renderizado en la nube (Google Colab).

- **Exploración de escenarios de microgravedad:** Como objetivo secundario, evaluar la capacidad de transferencia del modelo o su reentrenamiento para operar en entornos de microgravedad (0g), simulando condiciones espaciales.

1.4 Estructura del documento

La presente memoria se organiza en seis capítulos que detallan el desarrollo del proyecto:

- **Capítulo 1: Introducción.** Presenta el contexto, la motivación, la definición del problema y los objetivos del proyecto.

- **Capítulo 2: Formulación del problema y estado del arte.** Revisa la literatura existente sobre manipulación aérea y algoritmos de DRL, además de detallar la formulación matemática de la dinámica del sistema acoplado.

- **Capítulo 3: Modelado y simulación del sistema.** Describe la implementación técnica del robot en MuJoCo, detallando las características físicas del multi-rotor, los brazos LiCAS A1 y la arquitectura de control de bajo nivel.

- **Capítulo 4: Diseño del entorno de aprendizaje (RL) en 1g.** Explica la arquitectura del entorno en Gymnasium, profundizando en el diseño de la función de recompensa, el espacio de observaciones y la configuración del entrenamiento (hiperparámetros y algoritmo), todo en gravedad terrestre.

- **Capítulo 5: Adaptación y diseño del entorno en microgravedad (0g).** Trata la transferencia de conocimiento desde el entorno terrestre al espacial, mostrando las restructuraciones necesarias del modelo físico y la simulación, los problemas y dificultades encontradas, la reingeniería de control y la configuración del entrenamiento.

- **Capítulo 6: Resultados y discusión.** Presenta los resultados experimentales obtenidos, analizando las curvas de aprendizaje, el comportamiento del agente en la fase de gravedad terrestre y el análisis de la transferencia a microgravedad.

- **Capítulo 7: Conclusiones y trabajos futuros.** Resume los logros alcanzados, identifica las limitaciones actuales del sistema y propone líneas de investigación para dar continuidad al proyecto.

2 FORMULACIÓN DEL PROBLEMA Y ESTADO DEL ARTE

Este capítulo establece el marco teórico necesario para abordar el problema de la manipulación aérea autónoma. Se realiza una revisión preliminar de la literatura actual sobre sistemas de manipulación con multirrotores, identificando los desafíos dinámicos inherentes al acoplamiento mecánico. Posteriormente, se introduce el DRL como la solución metodológica adoptada para superar las limitaciones de los enfoques de control clásicos en entornos complejos y dinámicos. Finalmente, se formaliza matemáticamente la dinámica del sistema acoplado compuesto por el cuadrotor y los brazos robóticos LiCAS A1.

2.1 Manipulación aérea con multirrotores

La manipulación aérea representa una evolución natural de la robótica de servicio, extendiendo las capacidades de los UAVs desde la observación pasiva hacia la interacción física activa con el entorno.

2.1.1. Arquitectura y desafíos dinámicos

Un UAM se compone típicamente de una plataforma de vuelo, generalmente un multirrotor debido a su capacidad de vuelo estacionario, y uno o más manipuladores robóticos acoplados a su estructura base. A diferencia de los manipuladores industriales anclados al suelo, donde la base es cinemáticamente fija, un manipulador aéreo opera sobre una base flotante.

Desde la perspectiva del estado del arte, autores como **Ruggiero et al.** [1] clasifican estos sistemas según su configuración mecánica y método de agarre. El sistema propuesto en este proyecto (multi-rotor + LiCAS A1) entra en la categoría de manipuladores aéreos de brazos múltiples, una configuración que, aunque aumenta la carga útil y la destreza operativa, introduce una complejidad dinámica significativa.

Un importante desafío en esta área es el acoplamiento dinámico. El movimiento de las articulaciones de los brazos robóticos (en este caso, los servos del LiCAS A1) provoca dos efectos inmediatos sobre la plataforma aérea:

1. **Desplazamiento del Centro de Masa (CoM):** al extender los brazos para alcanzar la caja objetivo, el CoM global del sistema se desplaza. Si el controlador de vuelo no compensa este cambio, se genera un torque gravitacional que desestabiliza el dron, provocando derivas que pueden impedir la precisión necesaria para insertar los ganchos.

2. **Torques de reacción y momentos de inercia variable:** según la tercera ley de Newton, los torques aplicados por los motores de los brazos generan torques de reacción opuestos sobre el cuerpo del dron. Como señalan **Orsag et al.** [2]**,** estas perturbaciones son altamente no lineales (términos de Coriolis) y dependen de la velocidad y aceleración de las articulaciones de los manipuladores.

3. **Fuerzas externas de contacto:** fuerzas de contacto externas que el entorno y los objetos ejercen sobre los brazos robóticos, perturbando la base flotante del muti-rotor.

2.1.2. Estrategias de Control en la Literatura

Históricamente, el control de manipuladores aéreos se ha abordado mediante técnicas de control desacoplado o control adaptativo.

- **Enfoques desacoplados:** tratan al brazo robótico como una perturbación externa que el controlador de vuelo (generalmente un PID en cascada) debe rechazar. Este enfoque es funcional para movimientos lentos y cargas ligeras, pero falla en maniobras agresivas o cuando la masa del manipulador es significativa respecto a la del dron, como es el caso del sistema considerado aquí.

- **Enfoques basados en modelo:** técnicas como el *Computed Torque Control* o la Linealización por Realimentación intentan modelar matemáticamente toda la dinámica del sistema para predecir y cancelar las perturbaciones. Sin embargo, como discuten **Ollero y Heredia** [3]**,** estos métodos dependen de una identificación paramétrica extremadamente precisa (masas, tensores de inercia, longitudes exactas). En escenarios reales, donde el dron debe levantar una caja (como en este caso), la masa de la carga puede ser desconocida *a priori*, lo que degrada el rendimiento de los controladores basados puramente en modelo.

2.1.3. Manipulación aérea de doble brazo

El uso de dos brazos, como en la configuración LiCAS A1 empleada en este trabajo, añade una capa adicional de complejidad conocida como la coordinación bimanual. En la literatura, sistemas como el desarrollado en el proyecto AEROARMS han demostrado que el uso de dos brazos permite manipular objetos más grandes y reducir los torques en las articulaciones individuales.

Sin embargo, para la tarea específica de este proyecto —agarrar una caja por el asa utilizando ganchos en L— se requiere una precisión milimétrica simultánea en ambos efectores finales. Un error en la sincronización de los brazos o una desviación en el Yaw (guiñada) del dron puede provocar que un gancho colisione con la caja base o falle el agarre del asa, desestabilizando fatalmente al dron. La dificultad de programar explícitamente estas trayectorias coordinadas en un entorno con restricciones (suelo, techo, obstáculo inferior) motiva la exploración de métodos de aprendizaje automático, donde el agente descubre la política de coordinación óptima.

2.2 Aprendizaje por refuerzo en robótica (DRL)

El DRL ha surgido como una metodología disruptiva para el control de sistemas robóticos complejos. A diferencia del control clásico, que requiere un conocimiento explícito y preciso de la dinámica del sistema, el DRL permite que un agente aprenda políticas de control óptimas directamente a partir de la experiencia interactiva con el entorno. Esta capacidad es especialmente importante en tareas de manipulación aérea donde las interacciones de contacto (fricción, colisiones) y las dinámicas aerodinámicas son difíciles de modelar analíticamente con precisión.

2.2.1 Fundamentos de RL: Procesos de Decisión de Markov (MDP)

El problema de aprendizaje por refuerzo se modela matemáticamente como un Proceso de Decisión de Markov, MDP (Markov Decision Process, de sus siglas en inglés). Un MDP proporciona un marco formal para la toma de decisiones en situaciones donde los resultados son en parte aleatorios y en parte bajo el control del decisor.

Formalmente, un MDP se define como una tupla $\langle S, A, P, R, \gamma \rangle$, donde:

- ***S* (Espacio de Estados):** es el conjunto de todas las posibles configuraciones del sistema. En el contexto de este proyecto, el estado $s_t \in S$ incluye la información cinemática del cuadrotor (posición, velocidad lineal y angular) y de los brazos robóticos LiCAS A1 (ángulos y velocidades articulares), así como la posición relativa de la caja objetivo.

- ***A* (Espacio de Acciones):** es el conjunto de decisiones disponibles para el agente. Dado que se controlan motores y servos, el espacio de acciones $a_t \in A$ es continuo. En cada instante *t*, el agente emite un vector de acción que se traduce en comandos de empuje/torques para el dron y posiciones para los servos.

- ***P* (Función de Transición):** denotada como $P(s_{t+1} \mid s_t, a_t)$ define la probabilidad de transicionar al estado s_{t+1} dado el estado actual s_t y la acción ejecutada a_t. En este trabajo, esta función no se conoce explícitamente, sino que es implícita y está gobernada por el motor de física MuJoCo, que calcula la dinámica de cuerpos rígidos y colisiones.

- ***R* (Función de Recompensa):** $R(s_t, a_t, s_{t+1})$ es una señal escalar que el agente recibe tras ejecutar una acción. Es el mecanismo principal de retroalimentación; el objetivo del agente es maximizar la suma acumulada de recompensas a lo largo del tiempo. El diseño de esta función (*Reward Shaping*) es uno de los componentes principales de la ingeniería del sistema.

- ***γ* (Factor de Descuento):** un escalar $\gamma \in [0,1)$ que determina la importancia de las recompensas futuras frente a las inmediatas.

El objetivo del agente es aprender una política $\pi_\theta(a_t \mid s_t)$, parametrizada por una red neuronal con pesos θ, que maximice el retorno esperado $J(\pi)$:

$$J(\pi) = E_{\tau \sim \pi}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$$

Donde τ representa una trayectoria (episodio) de estados y acciones generada por la política.

2.2.2 Algoritmos On-Policy: Proximal Policy Optimization (PPO)

Dentro del espectro de algoritmos de DRL, la manipulación robótica continua suele abordarse mediante métodos de gradiente de política (*Policy Gradient*). En particular, este trabajo utiliza el algoritmo Proximal Policy Optimization (PPO), introducido por Schulman et al. en 2017, [4].

PPO es un algoritmo *On-Policy*, lo que significa que el agente aprende y actualiza su red neuronal utilizando los datos generados por su política actual. Se elige PPO frente a otras alternativas (como Deep Deterministic Policy Gradient (DDPG) o Soft Actor-Critic (SAC)) por su excelente equilibrio entre simplicidad de implementación, eficiencia de muestreo y, sobre todo, estabilidad numérica y convergencia.

2.2.2.1 La función objetivo "Clipped"

El problema fundamental de los métodos de gradiente de política estándar es que una actualización demasiado grande en los pesos de la red neuronal puede "romper" la política, llevando al agente a una zona del espacio de estados de la que no puede recuperarse (colapso de rendimiento).

PPO mitiga esto imponiendo una restricción en cuánto puede cambiar la nueva política π_θ respecto a la política anterior $\pi_{\theta_{old}}$. En lugar de usar una restricción dura (como TRPO), PPO utiliza una función objetivo "recortada":

$$L^{CLIP}(\theta) = \widehat{E}_t\left[\min\left(r_t(\theta)\widehat{A_t}, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\widehat{A_t}\right)\right]$$

Donde:

- $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ es el cociente de probabilidad entre la nueva y la vieja política.
- $\widehat{A_t}$ es la función de ventaja estimada, que cuantifica cuánto mejor es una acción comparada con el promedio de acciones en ese estado.
- ϵ es un hiperparámetro (típicamente 0.2) que define el intervalo de recorte.

2.2.2.2 Arquitectura Actor-Critic

PPO se implementa bajo una arquitectura Actor-Critic:

1. **El Actor (política π_θ)**: decide qué acción tomar dado un estado.
2. **El Critic (función de valor V_ϕ):** estima el retorno esperado de un estado, sirviendo como base para calcular la ventaja $\widehat{A_t}$ y reducir la varianza del gradiente.

Esta estructura permite al manipulador aéreo aprender maniobras complejas de forma robusta, evitando oscilaciones drásticas en el comportamiento del dron durante el entrenamiento, un factor vital cuando se simulan sistemas con dinámicas de vuelo sensibles.

2.3 Dinámica del sistema acoplado

El sistema robótico considerado en este trabajo es un UAM compuesto por una base flotante activa (el cuadrotor) y un sistema de manipulación de dos brazos (*LiCAS A1*). El modelado de este sistema se aborda mediante el formalismo de Newton–Euler para cuerpos rígidos, considerando el sistema como un conjunto de eslabones interconectados dinámicamente.

2.3.1 Ecuaciones de movimiento del cuadrotor

El movimiento del cuadrotor se define respecto a dos marcos de referencia: el marco inercial fijo en el mundo $\mathcal{F}_W = O_W, x_W, y_W, z_W$ y el marco del cuerpo $\mathcal{F}_B = O_B, x_B, y_B, z_B$ fijado al centro de masa (CoM) del dron.

El estado del cuadrotor se define por su posición $p = [x, y, z]^T \in R^3$ y su orientación, representada por la matriz de rotación $R \in SO(3)$ (o cuaterniones q en la implementación computacional), que mapea vectores de $\mathcal{F}_\mathcal{B}$ a $\mathcal{F}_\mathcal{B}$. Las ecuaciones dinámicas de traslación y rotación vienen dadas por:

$$\begin{cases} m\ddot{\mathrm{p}} = f_T \mathrm{Re}_3 - m\mathrm{g} + \mathrm{f}_{ext} \\ \mathrm{J}\dot{\omega} = \tau - \omega \times (\mathrm{J}\omega) + \tau_{ext} \end{cases}$$

donde:

- m: masa total del sistema.
- $f_T = \sum_{i=1}^{4} f_i$: empuje total generado por los cuatro rotores en la dirección del eje z_B, con $e_3 = [0,0,1]^T$.
- J: tensor de inercia del sistema.
- ω: velocidad angular del cuerpo.
- τ: par de control generado por los motores (roll, pitch, yaw).
- $fext, \tau ext$: fuerzas y pares externos, que incluyen perturbaciones aerodinámicas y las fuerzas de reacción generadas por el movimiento de los brazos robóticos.

2.3.1.1 Diferenciación entre gravedad terrestre (1g) y microgravedad (0g)

Un aspecto distintivo de este proyecto es la evaluación del sistema en dos regímenes gravitatorios, lo cual altera fundamentalmente el término g en la ecuación de traslación.

Régimen de gravedad terrestre (1g): en este escenario, $g = [0,0,9.81]^T \ \mathrm{m/s^2}$. El cuadrotor debe generar un empuje constante $f_T \approx mg$ para mantener el vuelo estacionario. Esto impone una carga continua en los actuadores y acopla fuertemente la actitud con la aceleración lateral. Para moverse lateralmente, el dron debe inclinar su vector de empuje, perdiendo componente vertical que debe ser compensada aumentando la potencia total.

Régimen de microgravedad (0g): en el escenario de simulación espacial, $g = [0,0,0]^T$. La dinámica cambia drásticamente:

$$m\ddot{\mathrm{p}} = f_T \mathrm{Re}_3 + \mathrm{f}_{ext}$$

En ausencia de peso, no es necesario un empuje de sustentación. Cualquier aplicación de fuerza f_T genera una aceleración pura sin necesidad de vencer la gravedad. Esto libera a los actuadores de la carga de sustentación, pero introduce un nuevo desafío de control: la inercia se convierte en el factor dominante y el sistema es mucho más sensible a las perturbaciones de los brazos, ya que no existe una fuerza restauradora gravitacional.

2.3.2 Cinemática de los manipuladores robóticos

El sistema de manipulación LiCAS A1 consta de dos brazos robóticos antropomórficos idénticos (izquierdo y derecho). Basándose en la definición del modelo físico, cada brazo posee cuatro grados de libertad rotacionales.

Sea $q_{brazo} = [q_1, q_2, q_3, q_4]^T$ el vector de variables articulares de un brazo, donde:

- q_1: flexión/extensión de hombro
- q_2: $abducción/aducción\ de\ hombro$
- q_3: rotación media/lateral de hombro

- q_4: flexión extensión de codo

La posición y orientación del efector final respecto al marco del cuerpo del dron $\mathcal{F}_{\mathcal{B}}$ se obtiene mediante la cinemática directa, representada como una composición de transformaciones homogéneas:

$$\mathrm{T}_{ee}^{B}(\mathrm{q}_{brazo}) = \mathrm{T}_{base}^{B} \cdot \mathrm{T}_{1}^{base}(q_1) \cdot \mathrm{T}_{2}^{1}(q_2) \cdot \mathrm{T}_{3}^{2}(q_3) \cdot \mathrm{T}_{ee}^{3}(q_4)$$

donde Tee^B es la matriz homogénea 4×4 que contiene la rotación Ree y la posición p_{ee} del gancho.

2.3.2.1 Acoplamiento cinemático y dinámico

El principal desafío abordado por el agente de aprendizaje por refuerzo es que la configuración de los brazos $q_{arms} = [q_L, q_R]$ modifica en tiempo real las propiedades inerciales del sistema aéreo completo.

El centro de masa total del sistema p_{CoM} varía en función de la posición de los brazos:

$$\mathrm{p}_{CoM}(\mathrm{q}_{brazos}) = \frac{1}{m_{total}}\left(m_{base}\mathrm{p}_{base} + \sum_{i} m_i \mathrm{p}_i(\mathrm{q}_{brazos})\right)$$

Dado que el controlador de vuelo asume un centro de masa estático o geométricamente centrado, el desplazamiento Δp_{CoM} inducido por la extensión de los brazos genera un par gravitacional parásito:

$$\tau_{dist} = \Delta \mathrm{p}_{CoM} \times m\mathrm{g}$$

Este efecto es percibido por el dron como una perturbación externa. El agente de RL debe aprender a predecirlo y compensarlo, ya sea modificando la distribución de empuje de los motores o reconfigurando los brazos para mantener el equilibrio.

Para modelar matemáticamente este comportamiento, el modelo dinámico completo del manipulador aéreo se formula habitualmente agrupando las fuerzas y pares en tres términos principales. En primer lugar, los términos inerciales, donde la aceleración lineal y angular de la base aérea y la aceleración articular de los brazos se afectan mutuamente de forma cruzada, reflejando el acoplamiento físico. En segundo lugar, los términos no lineales de Coriolis y fuerzas centrífugas, que dependen de las velocidades del sistema. Por último, el término de gravedad, responsable de generar los pares perturbadores debido a los desplazamientos continuos del centro de masa (CoM) mencionados anteriormente.

La obtención de este modelo dinámico completo se puede abordar mediante dos enfoques clásicos de la mecánica: la formulación de Euler-Lagrange o la de Newton-Euler. Si se emplea el método analítico de Euler-Lagrange, el estado del sistema se define mediante un vector de coordenadas generalizadas cuya dimensión es de $6 + 2N_{dof}$, donde los 6 primeros grados de libertad corresponden a la posición y actitud de la base flotante (dron), y los $2N_{dof}$ representan las articulaciones de los dos brazos robóticos, siendo N_{dof} el número de grados de libertad de cada brazo individual.

No obstante, en el contexto de este proyecto, la resolución de esta compleja interacción dinámica se delega en el motor físico. Para ello, MuJoCo no resuelve las ecuaciones simbólicas de Euler-Lagrange, sino que calcula el modelo dinámico del sistema completo en cada paso de simulación aplicando el método recursivo de Newton-Euler, mucho más eficiente para calcular las aceleraciones directas e inversas en cadenas cinemáticas complejas.

3 MODELADO Y SIMULACIÓN DEL SISTEMA

El desarrollo y validación de algoritmos de control para robots aéreos manipuladores conlleva riesgos operativos y costes elevados si se realizan directamente sobre plataformas físicas. Por ello, la simulación de alta fidelidad se presenta como una etapa intermedia indispensable. Este capítulo detalla la implementación virtual del sistema robótico, describiendo las herramientas de software empleadas, el modelado físico de los componentes y la arquitectura de control integrada.

3.1 Entorno de simulación: MuJoCo

Para la simulación física del entorno y del agente robótico se ha seleccionado el motor **MuJoCo** (*Multi-Joint dynamics with Contact*). Desarrollado originalmente por Emo Todorov en la Universidad de Washington y posteriormente adquirido y liberado como código abierto por Google DeepMind, MuJoCo se ha establecido como el estándar de facto en la comunidad de investigación en RL y robótica.

La elección de MuJoCo frente a otros motores físicos convencionales o motores de videojuegos (Unity, Unreal Engine) se fundamenta en tres ventajas técnicas críticas para la manipulación aérea:

3.1.1 Dinámica en coordenadas generalizadas

A diferencia de los motores de juegos que suelen utilizar coordenadas máximas (donde cada eslabón tiene 6 grados de libertad y se unen mediante restricciones algebraicas que pueden violarse numéricamente), MuJoCo opera en coordenadas generalizadas.

Esto implica que el sistema se modela matemáticamente como un árbol cinemático ramificado. Las articulaciones no son restricciones que se resuelven a posteriori, sino que definen la topología del sistema. Para el manipulador aéreo multi-rotor + LiCAS A1, esto garantiza que los brazos robóticos nunca se "separen" del chasis del dron, incluso bajo aceleraciones extremas o movimientos bruscos durante la fase de exploración del entrenamiento. Esta estabilidad numérica es fundamental para que el algoritmo PPO converja, ya que elimina el ruido estocástico derivado de errores de simulación.

3.1.2 Modelo de contacto avanzado para manipulación

El desafío central de este proyecto es el agarre de una caja mediante ganchos rígidos. La simulación de contactos rígidos es tradicionalmente problemática en computación, propensa a inestabilidades o penetraciones irreales de objetos.

MuJoCo aborda esto mediante un modelo de contacto único basado en optimización convexa. En lugar de tratar los contactos como restricciones duras e instantáneas, utiliza un modelo de "contacto suave" definible mediante los parámetros *solref* (referencia de solver) y *solimp* (impedancia de solver). Esto permite simular:

1. **Fricción realista:** fundamental para que los ganchos en L retengan el asa de la caja sin deslizarse irrealmente.

2. **Fuerzas normales estables:** permite al agente aprender a aplicar la fuerza necesaria para levantar la carga sin generar colisiones explosivas entre el gancho y el asa.

En la definición del modelo, se ha hecho uso explícito de estas capacidades definiendo las geometrías de los ganchos y el asa con primitivas de colisión de tipo cápsula (*capsule*), las cuales son computacionalmente eficientes y proporcionan superficies de contacto continuas y suaves.

3.1.3 Eficiencia computacional para DRL

El entrenamiento de agentes mediante DRL requiere millones de pasos de simulación. MuJoCo está escrito en C/C++ altamente optimizado, permitiendo velocidades de simulación muchas veces superiores al tiempo real.

Además, su integración con Python a través de los *bindings* oficiales permite el acceso directo a las estructuras de datos de bajo nivel (mjData y mjModel). Esto ha facilitado la implementación del código, permitiendo la lectura exacta de estados internos (posiciones, velocidades, matrices de rotación) para el cálculo de la recompensa y la reinicialización determinista del entorno, algo difícil de lograr con simuladores de caja negra.

3.2 Modelado físico del agente

La fidelidad de la simulación depende intrínsecamente de la precisión con la que se modelen las propiedades físicas del agente robótico. En este trabajo, el sistema se compone de dos subsistemas acoplados mecánicamente: la plataforma aérea (el cuadrotor) y el sistema de manipulación bimanual (los brazos "LiCAS A1"). A continuación, se detalla la caracterización inercial de ambos componentes, cuyos parámetros han sido incorporados al entorno MuJoCo para minimizar la brecha entre simulación y realidad.

3.2.1 Plataforma aérea (multi-rotor)

La plataforma aérea utilizada es un multirrotor de configuración cuadrotor en X, diseñado para soportar la carga útil del sistema de manipulación. Para garantizar una dinámica de vuelo realista, especialmente en maniobras que requieren cambios rápidos de actitud, es fundamental modelar correctamente la distribución de masa y los momentos de inercia del vehículo.

A partir del análisis CAD del modelo "Multirotor X4", se han extraído los siguientes parámetros inerciales que definen la respuesta del dron ante los pares generados por los motores y las perturbaciones externas:

- **Masa total de la base:** 4.65 kg.
- **Tensor de Inercia:** El tensor de inercia, calculado respecto al Centro de Masa (CoM) y expresado en el sistema de referencia del cuerpo, presenta una simetría característica de este tipo de aeronaves, donde los términos fuera de la diagonal (I_{xy}, I_{xz}, I_{yz}) son despreciables frente a los momentos principales de inercia.

Los valores implementados en el modelo físico son:

$$I_{base} = \begin{bmatrix} 0.3045 & 0 & 0 \\ 0 & 0.2156 & 0 \\ 0 & 0 & 0.3945 \end{bmatrix} [\mathrm{kg} \cdot \mathrm{m}^2]$$

Estos valores reflejan una distribución de masa donde la inercia en el eje Z ($I_{zz} \approx 0.39\ \text{kg} \cdot \text{m}^2$) es mayor que en los ejes de cabeceo y alabeo, consistente con la geometría plana y extendida del chasis.

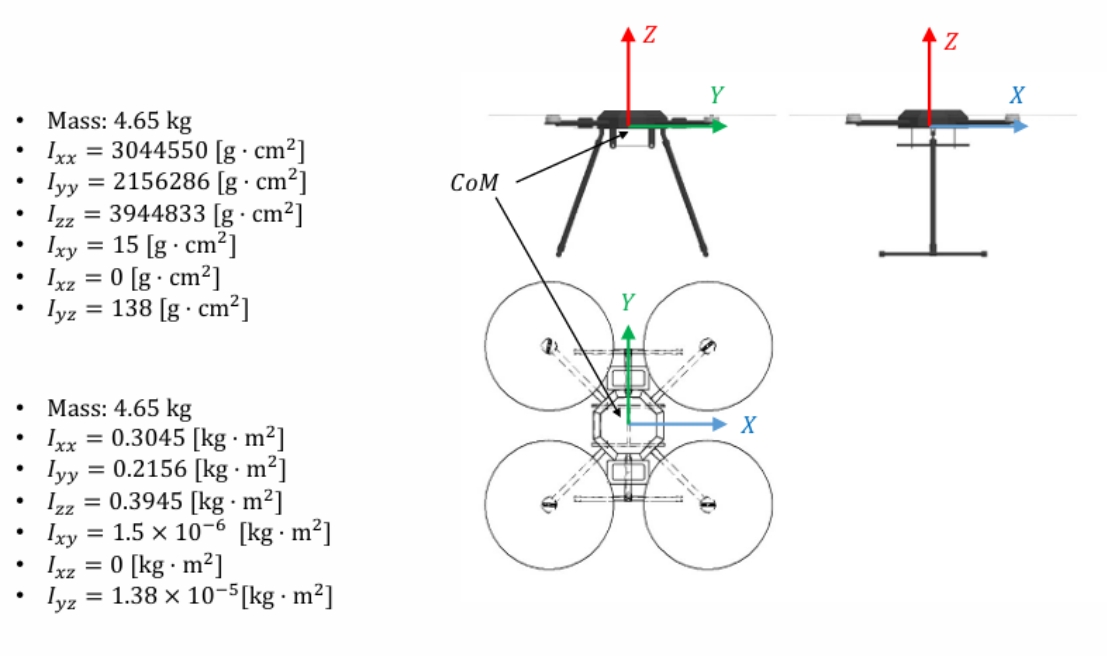


Imagen 1: Parámetros dinámicos multi-rotor

3.2.2 Sistema de manipulación (LiCAS A1)

El sistema de manipulación integrado es el **LiCAS A1**, un sistema robótico dual antropomórfico diseñado específicamente para aplicaciones de manipulación (rigidez variable o adaptabilidad mecánica).

El modelado de los brazos es imprescindible debido al acoplamiento dinámico: el movimiento de los eslabones altera el centro de gravedad total del sistema aéreo y genera pares de reacción que el controlador de vuelo debe compensar. Cada brazo se ha modelado como una cadena cinemática serie compuesta por eslabones rígidos conectados por articulaciones revolucionarias.

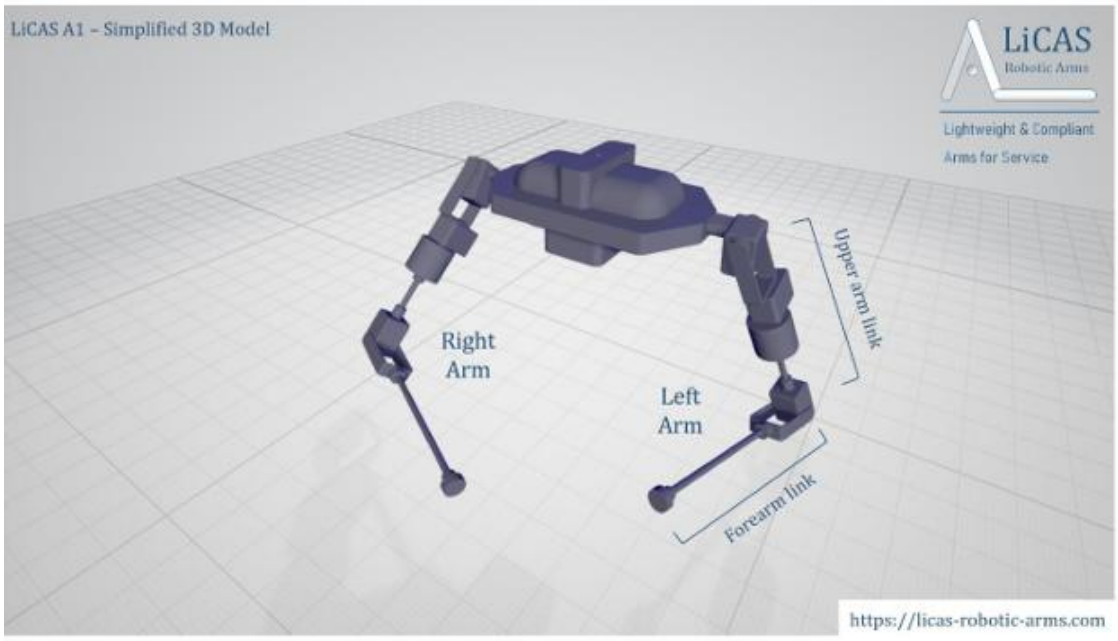


Imagen 2: Modelo 3D LiCAS A1

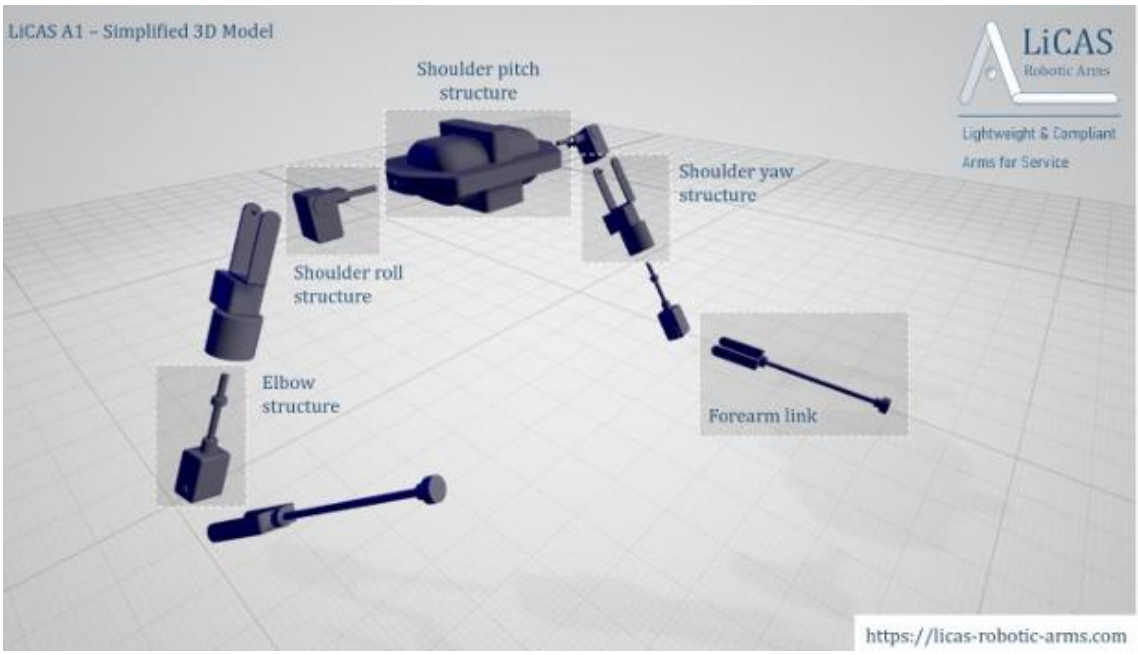


Imagen 3: Modelo 3D LiCAS A1 (2)

A continuación, se presentan los parámetros dinámicos (masa, inercia y posición del CoM) para cada eslabón de la cadena cinemática:

Tabla 1: Parámetros dinámicos LiCAS A1

Eslabón (link)	**Masa (kg)**	**Inercia Principal (Ixx ,Iyy,Izz) [kg·m2]**	**Centro de Masa (CoM) local [m]**
Estructura hombro (Pitch)	0.639	$I_{xx}: 3.02 \cdot 10^{-3}$ $I_{yy}: 8.64 \cdot 10^{-4}$ $I_{zz}: 3.39 \cdot 10^{-3}$	$\Delta z: -0.012$
Estructura hombro (Roll)	0.233	$I_{xx}: 3.68 \cdot 10^{-4}$ $I_{yy}: 1.16 \cdot 10^{-4}$ $I_{zz}: 3.76 \cdot 10^{-4}$	$\Delta y = 0.0236,$ $\Delta z = 0.0095$
Estructura hombro (Yaw)	0.246	$I_{xx}: 3.49 \cdot 10^{-4}$ $I_{yy}: 4.46 \cdot 10^{-4}$ $I_{zz}: 1.49 \cdot 10^{-4}$	$\Delta x = -0.015,$ $\Delta z = -0.100$
Estructura codo (Pitch)	0.214	$I_{xx}: 4.21 \cdot 10^{-4}$ $I_{yy}: 4.26 \cdot 10^{-4}$ $I_{zz}: 5.01 \cdot 10^{-5}$	$\Delta z: 0.039$
Antebrazo + gripper	0.106	$I_{xx}: 3.94 \cdot 10^{-4}$ $I_{yy}: 3.75 \cdot 10^{-4}$ $I_{zz}: 3.05 \cdot 10^{-5}$	$\Delta z: 0.092$

Es importante destacar la ligereza de los eslabones distales (el antebrazo apenas supera los 100 gramos), una característica de diseño fundamental en manipulación aérea para minimizar la inercia rotacional y reducir el impacto de los movimientos rápidos del brazo sobre la estabilidad del dron. Estos valores se han definido en el archivo XML de MuJoCo para asegurar que la simulación respete la física del hardware real.

3.3 Arquitectura de control de bajo nivel

El control de un sistema de manipulación aérea presenta un desafío dual: por un lado, requiere una

planificación de alto nivel para resolver la tarea estratégica (navegar hacia la caja, alinear los ganchos y levantarla); por otro, exige una gestión continua y precisa de los actuadores para mantener la estabilidad frente a la gravedad y las perturbaciones dinámicas.

En las fases iniciales de este proyecto, se exploró una arquitectura de control "End-to-End" pura, donde la red neuronal del agente interactuaba directamente con los comandos de los motores. En esta configuración, el espacio de acción del agente consistía en modificaciones porcentuales directas sobre el empuje de los cuatro rotores y los torques de las articulaciones.

Sin embargo, los experimentos preliminares demostraron que este enfoque conlleva severas limitaciones:

1. Dificultad de convergencia**:** el agente debe aprender desde cero la dinámica de vuelo, el mezclado de motores y la compensación de gravedad antes de poder siquiera aproximarse a la caja.

2. Comportamiento oscilatorio: las redes neuronales estocásticas tienden a generar salidas con ruido de alta frecuencia, lo que se traduce en vibraciones mecánicas inaceptables en los rotores y en movimientos espasmódicos de los brazos, impidiendo la precisión necesaria para insertar los ganchos en el asa.

3. Brecha simulación-realidad: en la robótica aérea real, los pilotos automáticos (como PX4) gestionan la estabilización básica. Un agente que aprende a controlar voltajes de motores directamente difícilmente sería transferible a un hardware físico.

Para superar estos obstáculos, se ha diseñado e implementado una arquitectura de control jerárquica o en cascada. En este esquema, el sistema se divide en dos bucles de control:

- Bucle externo (política de alto nivel): es el agente de DRL (PPO). Su función no es mover los motores, sino generar consignas de alto nivel (velocidad lineal deseada, tasa de yaw y posición objetivo de los brazos).

- Bucle interno (controlador de bajo nivel): es un conjunto de controladores clásicos (PID y control proporcional) que reciben las consignas del agente y calculan las fuerzas y torques exactos necesarios para cumplirlas, garantizando la estabilidad del sistema.

Esta abstracción permite que el agente se centre en la "toma de decisiones", delegando la ejecución dinámica al controlador de bajo nivel. A continuación, se detalla la implementación de estos controladores para la plataforma aérea y el sistema de manipulación.

3.3.1 Control de vuelo: mezclador y PID en cascada.

La implementación final del controlador de vuelo opera bajo la siguiente lógica de control en cascada:

3.3.1.1 Espacio de acción y referencias de control

El agente PPO genera un vector de acción continuo normalizado $a \in [-1,1]$. Las primeras cuatro componentes de este vector ($a_{0:3}$) se mapean a referencias físicas de velocidad en el marco del cuerpo del dron (Body Frame, F_B):

$$v_{(ref)}^{B} = \left[v_x^d; v_y^d; v_z^d\right] = \alpha_v \cdot a_{(0:2)}$$

$$\dot{\psi}_{ref} = \alpha_{\psi} \cdot a_3$$

Donde $\alpha_v = 1.5$ m/s y $\alpha_{\psi} = 1.0$ rad/s son los factores de escalado definidos en el entorno. Esto limita la agresividad del vuelo, garantizando que el agente opere dentro de una envolvente de vuelo segura.

3.3.1.2 Estimación de estado y cambio de marco de referencia

El simulador MuJoCo proporciona las velocidades lineales v^W en el marco inercial del mundo (world frame). Sin embargo, el control de vuelo de un cuadrotor se realiza intrínsecamente en su propio sistema de referencia (un sensor IMU real mide aceleraciones y velocidades angulares en F_B). Por tanto, el primer paso del bucle de control es transformar la velocidad del mundo al cuerpo mediante la matriz de rotación R_{WB}:

$$v^B = R_{WB}^T \cdot v^W$$

Esta operación es crítica para que el controlador PID pueda corregir errores relativos a la orientación actual del dron ("hacia adelante", "hacia la derecha"), independientemente de su orientación global.

3.3.2 Lazos de control PID

El sistema implementa tres lazos de control desacoplados para gestionar los grados de libertad del vehículo:

Control de Altitud (Eje Z)

Para el control vertical, se utiliza un controlador PID completo sobre el error de velocidad en el eje Z ($e_{vz} = v_z^d - v_z^B$). Dado que el mantenimiento de la altitud es crítico para la seguridad, este controlador incluye un término integral para eliminar el error en estado estacionario y un término derivativo para amortiguar oscilaciones:

$$u_z = K_{p,z} e_{vz} + K_{i,z} \int e_{vz} dt - K_{d,z} v_z^B$$

Para evitar el problema del *integral windup* (saturación del integrador) cuando el dron está lejos del objetivo, se implementó un *clamping* o saturación de la integral en el rango $[-0.2, 0.2]$. La salida final de empuje (u_{thrust}) suma este término de control a un valor *feedforward* de sustentación base (hover_throttle), calculado para contrarrestar exactamente la gravedad:

$$u_{thrust} = u_{hover} + u_z$$

Los parámetros sintonizados experimentalmente son $K_p = 6.0, K_i = 2.0, K_d = 0.2$

Control Horizontal (Ejes X-Y) y Actitud

El movimiento lateral utiliza una estructura en cascada:

- Lazo externo (velocidad): un controlador proporcional (P) convierte el error de velocidad lateral en ángulos de inclinación deseados (Roll ϕ_d y Pitch θ_d). Por ejemplo, si el dron necesita avanzar (velocidad X positiva), debe inclinar el morro hacia abajo (Pitch negativo).

$$\theta_d = K_{p,v}(v_x^d - v_x^B), \quad \phi_d = -K_{p,v}\left(v_y^d - v_y^B\right)$$

Estos ángulos objetivo se saturan a ± 0.35 radianes ($\approx 20°$) para evitar vuelcos incontrolables.

- Lazo Interno (actitud): un controlador PD compara los ángulos deseados con la orientación actual del dron (obtenida de la matriz de rotación) para calcular los torques de control necesarios:

$$u_{pitch} = K_{p,att}(\theta_d - \theta) - K_{d,att}\dot{\theta}$$

$$u_{roll} = K_{p,att}(\phi_d - \phi) - K_{d,att}\dot{\phi}$$

Con ganancias $K_{p,att} = 5.0$ y $K_{d,att} = 0.15$.

Control de Yaw

El control de dirección (Yaw) es un controlador P simple que actúa sobre la velocidad angular en el eje Z del cuerpo:

$$u_{yaw} = K_{p,yaw}(\dot{\psi_{ref}} - \dot{\psi})$$

Este término permite al agente alinear el dron con la caja objetivo para facilitar la inserción de los ganchos.

Frecuencias de ejecución

Conviene señalar que, en un controlador de vuelo real, estos lazos en cascada no funcionan todos a la vez, sino que cada uno va a su propio ritmo, marcado por la dinámica del dron y las limitaciones de los sensores. El lazo más interno, el de control de velocidad angular, tiene que reaccionar muy rápido ante perturbaciones y ruido de los motores, así que suele ejecutarse alrededor de 1 kHz. El siguiente nivel, el de actitud, que se basa en la fusión de datos de acelerómetros y giroscopios, trabaja a una frecuencia más baja, típicamente en torno a 250 Hz. Por último, el lazo externo, encargado de la velocidad y la posición, depende de sensores más lentos como el GPS o las cámaras, por lo que normalmente se queda sobre los 50 Hz.

En este entorno de simulación con MuJoCo, adaptado para RL, se ha optado por simplificar un poco este esquema usando una aproximación síncrona. Es decir, todos los lazos se calculan a la vez en cada paso de decisión del agente (según el tiempo de paso). Aunque es una simplificación, en la práctica permite acelerar mucho el entrenamiento de la red neuronal sin perder suficiente fidelidad como para mantener la estabilidad del sistema y poder validar correctamente la parte de manipulación.

Mezclador de motores

Finalmente, las salidas abstractas de los controladores ($u_{thrust}, u_{roll}, u_{pitch}, u_{yaw}$) deben traducirse en comandos individuales para los cuatro motores. Para una configuración de cuadrotor en "X", se utiliza la siguiente matriz de mezcla estándar:

$$\begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & +1 \\ 1 & +1 & -1 & -1 \\ 1 & +1 & +1 & +1 \\ 1 & -1 & +1 & -1 \end{bmatrix} \begin{bmatrix} u_{\{thrust\}} \\ u_{\{roll\}} \\ u_{\{pitch\}} \\ u_{\{yaw\}} \end{bmatrix}$$

Esta función distribuye las fuerzas de manera que un comando de *pitch*, por ejemplo, incremente la velocidad de los motores traseros y disminuya la de los delanteros, generando el par necesario sin alterar el empuje total neto. El resultado de esta matriz de mezcla es un vector de comandos que se limita estrictamente al rango normalizado de $[0,1]$. En la práctica, esta señal representa el porcentaje de uso de cada motor (una especie de PWM virtual) respecto a su empuje máximo.

En cuanto al dimensionamiento físico, la masa total del sistema (el chasis X4 de ≈ 4.5 kg más los brazos manipuladores) llega a rondar los 8 kg al despegue. Para que el control sea viable, en MuJoCo se han configurado los motores con un empuje máximo de unos 50 N cada uno (≈ 5 kg de tracción por motor). Así, para compensar los ≈ 80 N del peso en vuelo estacionario, el empuje base se queda en torno al 40% de la capacidad total (es decir, valores de 0.4 a la salida del mezclador).

Este sobredimensionamiento no es casual, es algo básico en el diseño de multirrotores. Mantener el empuje por debajo del 50% deja margen de maniobra suficiente para que el sistema pueda responder cuando se le exige más, por ejemplo, al generar momentos para inclinar el dron. Si los motores ya estuvieran cerca del límite, no podrían aportar ese extra de empuje sin saturarse, lo que acabaría provocando pérdidas de altura o incluso inestabilidad durante la manipulación.

Esta arquitectura robusta libera al algoritmo de RL de la carga de aprender la estabilización básica, permitiéndole centrarse en la coordinación compleja requerida para la maniobra de agarre.

3.3.3 Control de vuelo: mezclador y PID en cascada

El subsistema de manipulación LiCAS A1, compuesto por dos brazos antropomórficos de 4 grados de libertad cada uno, presenta un reto de control fundamental: la gestión de la impedancia mecánica. Mientras que el dron requiere un control dinámico agresivo para volar, los brazos requieren una precisión geométrica estricta para alinear los ganchos con el asa de la caja.

La evolución del proyecto atravesó dos paradigmas de control claramente diferenciados, cuya comparación justifica la arquitectura final adoptada.

1. **Enfoque inicial: control de torque directo**

En la primera fase del desarrollo, los actuadores de los brazos se definieron como motores de par puro. En la nomenclatura de MuJoCo, la entrada de actuación se exige en fuerza neta (N) o par (Nm). Esto se modeló mediante la etiqueta <motor>, que aplica un par en la articulación proporcional a la señal de control normalizada enviada por el agente $a_t \in [-1,1]$. Matemáticamente, la acción del agente se traducía directamente en un torque aplicado τ:

$$\tau_{joint} = gear \cdot a_t$$

En este simulador, el parámetro *gear* funciona como el multiplicador de fuerza máxima. Para los grados de libertad de los brazos, este atributo se define como un escalar. Se estableció el valor de 20 mediante ensayo y error empírico; aunque los servomotores reales LiCAS poseen relaciones de reducción mecánicas superiores (e.g., 250:1), una ganancia de 20 Nm generaba el par necesario para levantar los eslabones contra la gravedad, evitando a su vez explosiones de gradiente e inestabilidades numéricas durante las primeras épocas de entrenamiento del modelo.

Por el contrario, el atributo *gear* aplicado a los rotores del dron (etiqueta <general>) se definió como un vector de seis dimensiones (tres componentes de fuerza y tres de momento). Esto se debe a que un rotor aplica fuerzas espaciales sobre un punto del chasis. Por ejemplo, la asignación de valores gear="0

0 50 0 0 1.5" para un motor especifica que, al máximo de capacidad ($a_t = 1$), el rotor genera un empuje de 50 N en el eje Z y, simultáneamente, un par de reacción aerodinámica de 1.5 Nm en el eje de guiñada (*Yaw*). El signo de este último término se alternó positiva y negativamente entre los motores para simular los sentidos de giro opuestos de la configuración en X.

Este enfoque, aunque físicamente válido, presentó insuficiencias críticas para la tarea de manipulación aérea:

1. **El problema de la compensación de gravedad:** al no existir un bucle de control interno, si el agente emitía una acción cercana a cero ($a_t \approx 0$), el torque aplicado era nulo. Esto provocaba que los brazos "cayeran" bajo su propio peso debido a la gravedad, comportándose como un sistema pasivo o "muerto". El agente de RL se veía obligado a aprender, mediante prueba y error, a emitir un torque constante solo para sostener los brazos en el aire, desperdiciando capacidad de la red neuronal en reaprender la gravedad en lugar de focalizarse en la tarea de agarre.

2. **Inestabilidad estocástica:** dado que la política estocástica del agente (PPO) introduce ruido durante la exploración, pequeñas variaciones en la salida de la red neuronal resultaban en variaciones directas de la fuerza. Esto generaba movimientos espasmódicos y vibraciones de alta frecuencia en los efectores finales, haciendo imposible la inserción precisa de los ganchos en un asa de apenas 12 mm de diámetro.

2. Solución implementada: control de posición

Para solventar las limitaciones anteriores, la arquitectura final sustituyó los actuadores de tipo <motor> por actuadores de tipo <position>.

Este cambio de paradigma transforma el problema de control: el agente ya no decide cuánta fuerza aplicar, sino qué configuración geométrica (ángulo) desea alcanzar. El simulador, emulando la electrónica de un servomotor digital real, asume la responsabilidad de generar los torques necesarios para alcanzar y mantener dicha posición.

Formulación matemática del actuador de posición:

En el modelo final, cada articulación está gobernada por una ley de control proporcional-derivativa (PD) interna implementada nativamente por el solver de MuJoCo:

$$\tau_{aplicado} = K_p(q_{ref} - q_{actual}) - K_d \dot{q_{actual}}$$

Donde:

- q_{ref}: es el ángulo objetivo dictado por el agente de RL.

- K_p: es la ganancia proporcional o rigidez del servo. En el modelo final, se ha configurado un valor de Kp = 20, lo que proporciona la rigidez suficiente para soportar el peso del propio brazo sin ceder ante la gravedad.

- K_d: es la ganancia derivativa o amortiguamiento (damping). Se ha definido explícitamente en las articulaciones (<joint damping="2.0" .../>) para disipar la energía cinética y evitar oscilaciones o sobreimpulsos al llegar a la posición deseada.

3. Integración en el espacio de acción

La implementación en el entorno de aprendizaje gestiona la interfaz entre la red neuronal y estos actuadores de posición. Dado que la red neuronal emite acciones normalizadas en el rango $[-1,1]$, es necesario escalarlas a los límites físicos reales de las articulaciones del robot LiCAS A1.

El mapeo se realiza de la siguiente manera:

$$q_{ref}^{i} = a_{RL}^{i} \cdot \text{Limit}_i$$

Donde Limit_i corresponde a los rangos articulares definidos en el modelo XML (por ejemplo, ± 1.57 rad para los hombros y ± 2.6 rad para la rotación de muñeca/gancho).

Ventajas críticas de este control:

- **Abstracción cinemática:** el agente "piensa" en poses. Si el agente decide "extender el brazo", el controlador de bajo nivel garantiza que el brazo se extienda, independientemente de las perturbaciones externas o la orientación del dron.

- **Rigidez artificial:** el uso de una ganancia K_p elevada permite simular el comportamiento de bloqueo de los servos. Esto es vital durante la fase de levantamiento de la carga: cuando el dron asciende, el peso de la caja tiende a abrir los brazos. El control de posición resiste pasivamente esta fuerza externa, manteniendo el agarre firme sin que el agente deba calcular activamente la fuerza de fricción necesaria.

- **Reducción del sim-to-real gap:** los manipuladores aéreos reales suelen operar con servomotores controlados por posición (PWM). Al entrenar el agente con este esquema, la política aprendida es directamente transferible a un robot físico, ya que las señales de salida son compatibles con los protocolos de control de servos estándar.

En conclusión, la transición a un esquema de control de posición ha sido determinante para dotar al sistema de la estabilidad y precisión milimétrica requeridas, delegando la dinámica de bajo nivel al motor físico y permitiendo al algoritmo de DRL especializarse en la planificación de la maniobra de agarre.

4 DISEÑO DEL ENTORNO DE APRENDIZAJE (RL) EN 1G

La formulación de una tarea de control robótico como un problema de DRL, requiere una traducción precisa de la física del sistema a un Proceso de Decisión de Markov (MDP). En esta fase del proyecto, el objetivo principal es que el agente aprenda una política de control óptima que le permita navegar en un entorno simulado bajo gravedad terrestre (1g), estabilizar el sistema aéreo-manipulador híbrido, y realizar un agarre de precisión sobre una carga objetivo.

Para estandarizar la interacción entre el algoritmo de optimización (PPO) y el simulador dinámico de cuerpos rígidos (MuJoCo), se ha desarrollado un entorno virtual a medida utilizando la interfaz Gymnasium (la evolución del estándar OpenAI Gym). Esta API define el contrato de comunicación fundamental: en cada instante de tiempo discreto t, el entorno proporciona al agente una observación del estado s_t, el agente emite una acción a_t, y el entorno devuelve el siguiente estado s_{t+1} junto con una señal de recompensa r_t.

El diseño meticuloso de este entorno es el núcleo metodológico del proyecto. A continuación, se detalla la arquitectura de los espacios de observación y acción, elementos que determinan la "visión" y la "capacidad de actuación" del agente robótico.

4.1 Arquitectura del entorno (Gymnasium)

4.1.1 Qué es Gymnasium

Gymnasium (la evolución directa de la API estándar OpenAI Gym) es una biblioteca de código abierto en Python diseñada específicamente para el desarrollo, evaluación y estandarización de algoritmos RL. Su propósito fundamental es actuar como una interfaz unificada (un *middleware*) que aísla la lógica del algoritmo de aprendizaje (el agente PPO) de las complejidades del simulador dinámico subyacente (el motor físico MuJoCo). Para este proyecto, se ha implementado en el código una clase personalizada, la cual hereda directamente de la clase base gymnasium.Env, garantizando así la compatibilidad total con librerías modernas de RL.

El funcionamiento de Gymnasium se fundamenta en la ejecución computacional del bucle clásico de un MDP. La dinámica de interacción se orquesta principalmente a través del método step(action). En cada instante de tiempo, el agente envía un vector de acción al entorno; a continuación, Gymnasium traslada esta orden al motor físico MuJoCo para que avance la simulación matemática un número determinado de pasos (definido por el parámetro frame_skip). Inmediatamente después, el método retorna una tupla estandarizada compuesta por cinco elementos críticos: la nueva observación del sistema (obs), la señal escalar de refuerzo (reward), una bandera booleana que indica si el episodio ha concluido por un estado terminal como una colisión o el éxito (terminated), otra bandera para límites de tiempo máximos (truncated), y un diccionario con métricas adicionales de diagnóstico (info).

A nivel de funcionalidades, Gymnasium proporciona un sistema de tipado riguroso mediante su módulo gymnasium.spaces. Esto permite definir matemáticamente la topología y los límites exactos de las entradas y salidas del sistema. En el caso de la robótica continua, se hace un uso extensivo de la clase gymnasium.spaces.Box, la cual define tensores de dimensiones específicas (arrays n-

dimensionales) acotados entre valores máximos y mínimos. Adicionalmente, la API gestiona el ciclo de vida de la simulación mediante el método reset(), encargado de reinicializar las posiciones del dron, los brazos y la carga objetivo introduciendo variabilidad estocástica al inicio de cada episodio para mejorar la generalización del modelo, así como el método render() para la extracción y visualización de los frames RGB del entrenamiento.

Para garantizar la convergencia de las redes neuronales que parametrizan al Actor y al Crítico en el algoritmo PPO, es imperativo que los datos de entrada (observaciones) y salida (acciones) posean una dimensionalidad manejable, estén correctamente delimitados y posean un significado físico unívoco.

4.1.2 Espacio de observación

El espacio de observación define toda la información sensorial y exteroceptiva a la que el agente tiene acceso para tomar decisiones. En este proyecto, se ha definido un espacio continuo gymnasium.spaces.Box compuesto por un vector unidimensional de 43 variables de estado ($s \in R^{43}$). Para evitar gradientes explosivos durante el entrenamiento, todas las observaciones se limitan artificialmente (clipping) al rango $[-10,10]$.

Este vector de 43 dimensiones se construye mediante la concatenación estratégica de tres bloques de información:

1. Estado cinemático del multi-rotor (21 variables):

Describe la situación espacial y dinámica del cuadrotor en el marco inercial del mundo.

- **Posición global (3 variables):** coordenadas cartesianas (x, y, z) del centro de masa del dron (pos).
- **Matriz de rotación (9 variables):** a diferencia de usar ángulos de Euler o cuaterniones (cuyo doble recubrimiento q y $-q$ representan la misma rotación, confundiendo a la red neuronal), se ha extraído la matriz de rotación completa de 3×3 y se ha aplanado a un vector de 9 elementos. Esta representación continua y única facilita drásticamente el aprendizaje de la actitud.
- **Velocidad lineal (3 variables):** vector de velocidad de traslación del dron en los tres ejes.
- **Velocidad angular (3 variables):** tasas de rotación del dron (Roll, Pitch, Yaw).

2. Estado propioceptivo de los manipuladores (16 variables):

Informa al agente sobre la configuración geométrica y cinética instantánea de los dos brazos robóticos LiCAS A1.

- **Posiciones articulares (8 variables):** ángulos actuales de los 8 servomotores (q_{11} a q_{14} para el brazo izquierdo, y q_{21} a q_{24} para el derecho).
- **Velocidades articulares (8 variables):** tasas de cambio angular de cada articulación.

3. Información relativa y geometría del objetivo (6 variables):

Este es el bloque más crítico para el éxito de la tarea, ya que guía la precisión milimétrica del agarre. En lugar de proporcionar coordenadas globales absolutas de los objetivos, se calculan vectores relativos, lo que otorga invarianza traslacional a la red (el agente sabe cómo agarrar independientemente de dónde esté la caja en el mundo).

- **Vector dron-objetivo (3 variables):** vector que apunta desde el centro de masa del dron hacia el centro del asa de la caja objetivo (rel_target = target_pos - pos).

- **Vectores de error de los ganchos (6 variables):** vectores relativos directos desde la punta de cada gancho (l_hook, r_hook) hacia sus respectivos puntos de inserción óptimos en el asa (rel_l, rel_r). Cabe destacar que los puntos objetivos (t_{left} y t_{right}) se calculan aplicando un desplazamiento artificial al centro del objetivo en el eje Y (± 0.12 m). Esta separación guía a cada gancho hacia los extremos del asa cilíndrica.

4.1.3 Espacio de acción

El espacio de acción dicta cómo el agente puede intervenir sobre el entorno. Siguiendo las mejores prácticas en DRL continuo, se ha definido un espacio simétrico y normalizado gymnasium.spaces.Box de 12 dimensiones ($a \in [-1,1]^{12}$). Esta normalización estricta es esencial, ya que garantiza que las capas de salida de la red neuronal (activadas típicamente por funciones como *Tanh*) operen en un régimen de varianza controlada.

Dentro del entorno, este vector adimensional de 12 elementos se decodifica y escala a magnitudes físicas reales, dividiendo la orden en comandos de vuelo y comandos de manipulación:

1. Comandos de vuelo (4 variables):

Las primeras cuatro componentes de la acción ($a_{0:3}$) sirven como referencias para el controlador PID de bajo nivel del dron.

- **Referencias de velocidad lineal ($a_{0:2}$):** se interpretan como la velocidad deseada en el marco del cuerpo del cuadrotor. Estos valores normalizados se multiplican por un factor de escala de 1.5, traduciéndose en una velocidad máxima permitida de ± 1.5 m/s en los ejes X, Y y Z. Esto evita comportamientos agresivos indeseados y asegura aproximaciones estables.

- **Referencia de guiñada (a_3):** controla la tasa de rotación (Yaw rate) del dron, escalada por un factor de 1.0 rad/s. Esta acción es vital para alinear perpendicularmente la orientación del robot con el asa de la caja antes del agarre.

2. Comandos de manipulación (8 variables):

Las ocho componentes restantes ($a_{4:11}$) corresponden a las referencias de control de posición para los servos de los brazos LiCAS A1. Como se justificó en la sección de control de bajo nivel, el agente no comanda torques, sino configuraciones geométricas.

- **Escalado de límites físicos:** para mapear la acción $[-1,1]$ a radianes reales, se utiliza un vector de límites articulares, extraído de la caracterización cinemática del hardware. El vector aplicado es [1.57, 1.57, 1.57, 2.6, 1.57, 1.57, 1.57, 2.6].

- Esto significa que las tres primeras articulaciones de cada brazo (hombro y codo) tienen un rango de operación de $\pm\frac{\pi}{2}$ radianes (90°), mientras que la rotación final de la muñeca que porta el gancho tiene un rango extendido de ± 2.6 radianes (150°) para permitir ajustes finos en la inserción. La multiplicación matricial arm_action_scaled = action[4:] * arm_limits restringe intrínsecamente al agente a no generar colisiones de auto-intersección imposibles.

Esta arquitectura híbrida de observación y acción proporciona al agente un conocimiento perfecto y continuo de la métrica de la tarea, a la vez que abstrae las complejidades oscilatorias de los motores, estableciendo un marco de interacción sólido para la fase de diseño de recompensas.

4.2 Ingeniería de la función de recompensa ("reward shaping")

4.2.1 La función de recompensa

En el marco del RL modelado como un MDP, la función de recompensa, denotada formalmente como $R(s_t, a_t, s_{t+1})$, constituye el único mecanismo de retroalimentación a través del cual el agente evalúa la calidad de sus acciones. Es el canal de comunicación exclusivo para transmitir al algoritmo PPO cuál es el objetivo de la tarea.

Para un problema clásico y simple, a menudo basta con definir una función de recompensa dispersa, otorgando un valor positivo de $+1$ si se alcanza la meta y 0 en cualquier otro caso. Sin embargo, el sistema robótico híbrido abordado en este Trabajo de Fin de Grado presenta una complejidad importante: el agente debe controlar simultáneamente 12 grados de libertad continuos basándose en un espacio de observación de 43 dimensiones. En un entorno de tan alta dimensionalidad, la probabilidad de que el agente ejecute estocásticamente la secuencia exacta de comandos para volar, estabilizarse, alinear los ganchos y levantar la caja es virtualmente nula. Una recompensa dispersa resultaría en un gradiente de aprendizaje plano, impidiendo cualquier tipo de convergencia.

Para superar esta barrera teórica, se ha tenido que trabajar mucho el ámbito de la Ingeniería de la Función de Recompensa (Reward Shaping). Esta técnica es uno de los componentes más importantes y complejos en el desarrollo de este proyecto. Consiste en el diseño meticuloso de una función de recompensa densa que proporcione retroalimentación continua e incremental, guiando al agente paso a paso a través de las distintas fases operativas de la misión: navegación, aproximación al objetivo, alineación geométrica de los ganchos, agarre y elevación de la carga.

El diseño de esta función no es trivial y es altamente susceptible a un fenómeno conocido en la literatura de IA como *Reward Hacking* (explotación o hackeo de la recompensa). Esto ocurre cuando el agente descubre lagunas matemáticas en la función que le permiten maximizar su puntuación realizando comportamientos anómalos o físicamente indeseados que no resuelven la tarea real. Cabe destacar que, en cualquier poryecto de RL, siempre se va a transcurrir por esta etapa, es decir, el agente siempre, en el 100% de los casos, va a intentar hackear la recompensa (de hecho, esto tiene interesantes aplicaciones en otros sectores y es ampliamente utilizado, por ejemplo, para encontrar *glitches* o *bugs* en muchos sistemas, programas, etc). Por ello, la función de recompensa debe actuar como un delicado sistema de pesos y contrapesos que no solo premie el progreso, sino que también penalice activamente comportamientos como inestabilidades de vuelo, movimientos articulares bruscos, vibraciones o colisiones estructurales.

La formulación final de la recompensa refleja la necesidad vital de priorizar la precisión y la estabilidad frente a la velocidad bruta. Para lograr este comportamiento convergente y evitar transiciones de estado abruptas (sentencias *if/else* duras que "rompen" el gradiente de la política de PPO), se ha hecho un uso

extensivo de funciones de recompensa continuas y penalizaciones proporcionales.

A continuación, se detalla la evolución empírica de este diseño, analizando los modos de fallo conductuales observados durante el entrenamiento y las soluciones algorítmicas implementadas en cada iteración para esculpir la política de control final.

4.2.2 Estrategias de aproximación: el problema de la aproximación impulsiva

La primera fase operativa de la misión consiste en la navegación desde el punto de aparición estocástico del dron hasta las proximidades de la carga objetivo. Aunque conceptualmente simple, enseñar a un agente a volar hacia un punto en el espacio sin programar una trayectoria explícita es un desafío en la formulación de la recompensa.

En las versiones preliminares del entorno, la recompensa de aproximación se definió mediante una función lineal inversamente proporcional a la distancia euclídea entre el dron y la caja. La lógica matemática dictaba que: a menor distancia, mayor recompensa instantánea.

Sin embargo, el algoritmo PPO, en su estricta búsqueda matemática de maximizar la suma acumulada del retorno esperado $J(\pi_\theta)$, encontró una laguna (Reward Hacking). Si la recompensa aumenta cuanto más cerca se está, la política óptima descubierta por la red neuronal es alcanzar el objetivo en el menor número de *timesteps* (pasos de simulación) posibles. Al no existir ningún incentivo para la conservación del momento o la seguridad estructural, el agente aprendió a inclinar agresivamente el vector de empuje del dron para acelerar a fondo hacia la caja, ignorando por completo cualquier maniobra de frenado. Este comportamiento resultaba invariablemente en colisiones catastróficas a alta velocidad contra la estructura de la caja base.

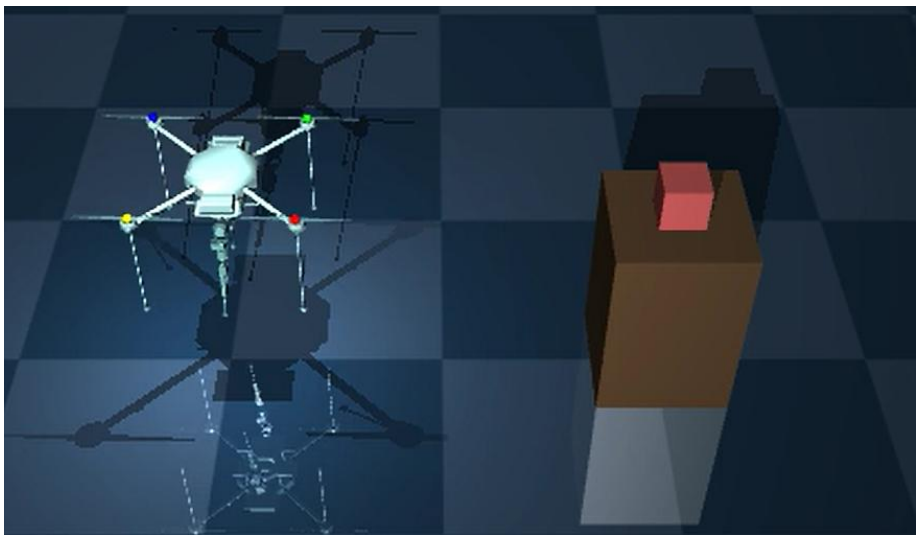


Imagen 4: Situación inicial

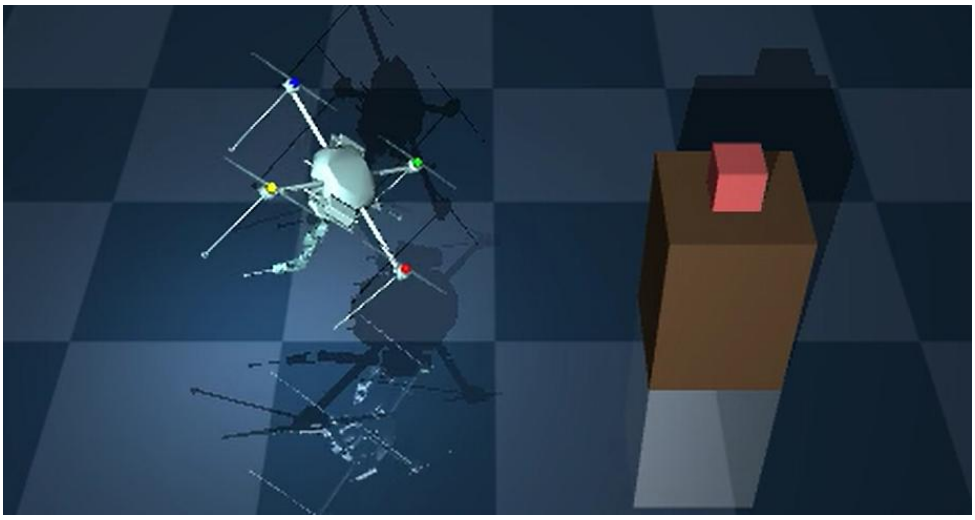


Imagen 5: el dron se inclina para ganar la máxima velocidad

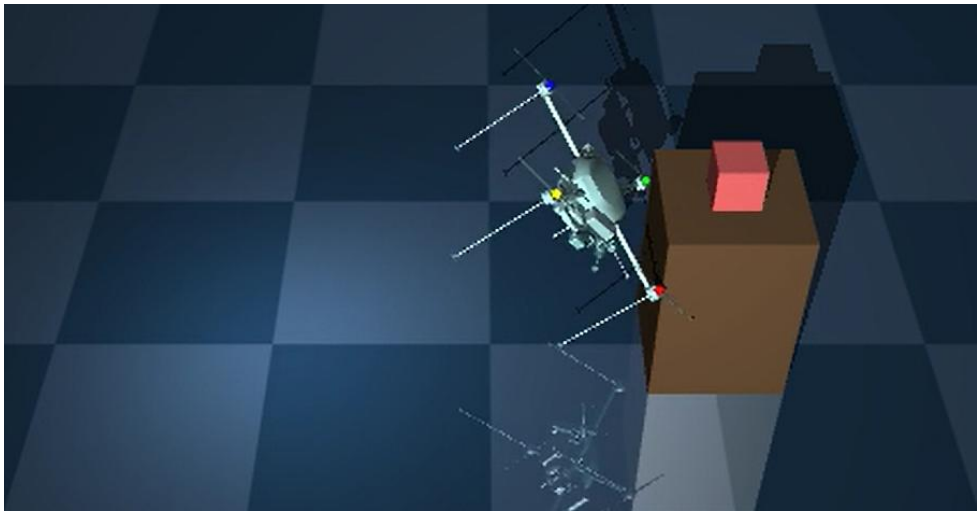


Imagen 6: el dron choca con la caja base a máxima velocidad

Para mitigar este fallo conductual, era evidente la necesidad de incorporar la velocidad del sistema a la ecuación de recompensa. No obstante, este ajuste desencadenó un segundo problema teórico de optimización.

4.2.2.1 El "miedo" a la aproximación y el estancamiento en mínimos locales

El primer intento para frenar al dron consistió en añadir un término de penalización estricto y constante basado en la magnitud de la velocidad lineal del vehículo ($|v|$). La intención era castigar los desplazamientos bruscos. Sin embargo, la asignación de un peso excesivo a esta penalización (por ejemplo, coeficientes de -3.0) provocó un colapso en la política del agente.

El agente desarrolló un comportamiento que podríamos definir de forma coloquial como "miedo a acercarse". En la dinámica de un MDP, si el castigo instantáneo incurrido por adquirir velocidad supera el beneficio marginal obtenido por reducir la distancia a la caja unos pocos centímetros, el gradiente de la política dicta que la mejor decisión es no moverse. El cuadrotor aprendió a estabilizarse en un vuelo estacionario perfecto exactamente en su punto de aparición, rehusando acercarse al objetivo para no ser penalizado por la velocidad necesaria para el tránsito. El sistema había convergido hacia un

mínimo local subóptimo del que le resultaba imposible escapar mediante simple exploración estocástica.

4.2.2.2 La solución: puntos de aproximación y penalización dinámica

Para resolver esta dicotomía entre el impacto agresivo e incontrolado y la parálisis por penalización, en la iteración final del entorno se rediseñó por completo la estrategia de navegación mediante dos innovaciones matemáticas clave:

1. Desplazamiento del objetivo (punto de aproximación):

En lugar de recompensar al agente por volar directamente hacia el asa de la caja (lo que propiciaba colisiones inminentes), se estableció un punto de aproximación virtual. Este punto se define con un offset geométrico respecto al asa: situado 30 cm hacia atrás en el eje X, centrado en el eje Y, y 40 cm por encima en el eje Z (approach_offset = np.array([-0.3, 0.0, 0.40])). Además, la recompensa lineal por distancia se sustituyó por una función exponencial decreciente:

$$R_{dist} = 3.0 \cdot e^{-1.0 \cdot d_{approach}}$$

Esta formulación matemática suaviza los gradientes, otorgando recompensas significativas solo cuando el dron se posiciona de forma estable en una "zona segura" de observación previa al agarre, actuando como un punto de preparación. Adicionalmente, se incluyó un bonus temporal de $+1.0$ puntos solo si el agente superaba su mejor distancia histórica en el episodio, forzando una exploración progresiva.

2. Penalización de velocidad condicionada:

Para solucionar el estancamiento en el mínimo local sin recuperar el comportamiento descontrolado, se programó una penalización dinámica por zonas. La magnitud del castigo por la velocidad ya no es estática, sino que depende estrictamente de la distancia al objetivo:

- **Zona lejana ($d_{approach} > 1.0$ m):** la penalización es leve ($-0.5 \cdot |v|$). Esto permite al dron adquirir la velocidad necesaria para cruzar el espacio de simulación rápidamente sin que el algoritmo "tema" penalizaciones desproporcionadas.
- **Zona de transición ($0.5 \leq d_{approach} \leq 1.0$ m):** la penalización se incrementa a $-1.0 \cdot |v|$, advirtiendo a la red neuronal que debe comenzar a reducir la inclinación de actitud.
- **Zona de aproximación fina ($d_{approach} < 0.5$ m):** la penalización se vuelve severa ($-1.5 \cdot |v|$). Si el dron entra rápido en este radio crítico, la resta masiva de puntos anula la recompensa ganada por proximidad.

Esta estructuración por intervalos actúa en la práctica como un sistema de guiado, obligando al agente a desarrollar una política de vuelo en dos tiempos: una navegación inicial de crucero rápida y decidida, seguida de un frenado suave y progresivo antes de detenerse de manera estable en el punto de aproximación, preparado para la inserción cuidadosa de los brazos robóticos.

4.2.3 Precisión y geometría del agarre

Una vez resuelto el problema de la navegación segura hacia la zona de operación, el siguiente desafío

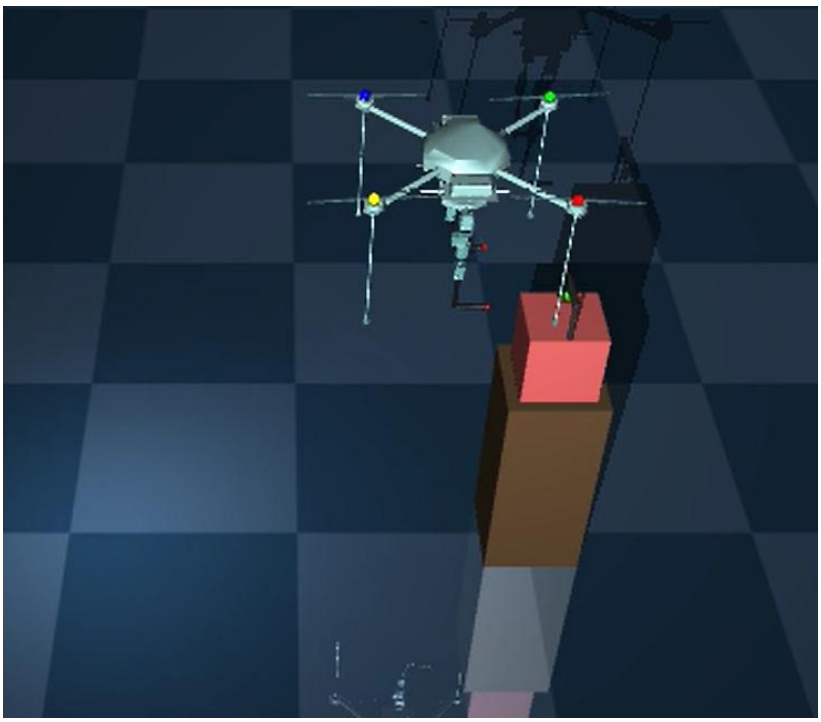
algorítmico consiste en guiar los efectores finales (los ganchos de los brazos LiCAS A1) hasta su posición de enclavamiento en la caja objetivo. Esta fase es mecánicamente crítica debido a la estricta tolerancia del sistema: el asa de la caja es un cilindro de apenas 12 mm de radio y 23 cm de longitud, y los efectores finales son ganchos rígidos en forma de L (hook_L_tip, hook_R_tip) que deben insertarse con precisión milimétrica.

4.2.3.1 El fracaso de la aproximación directa

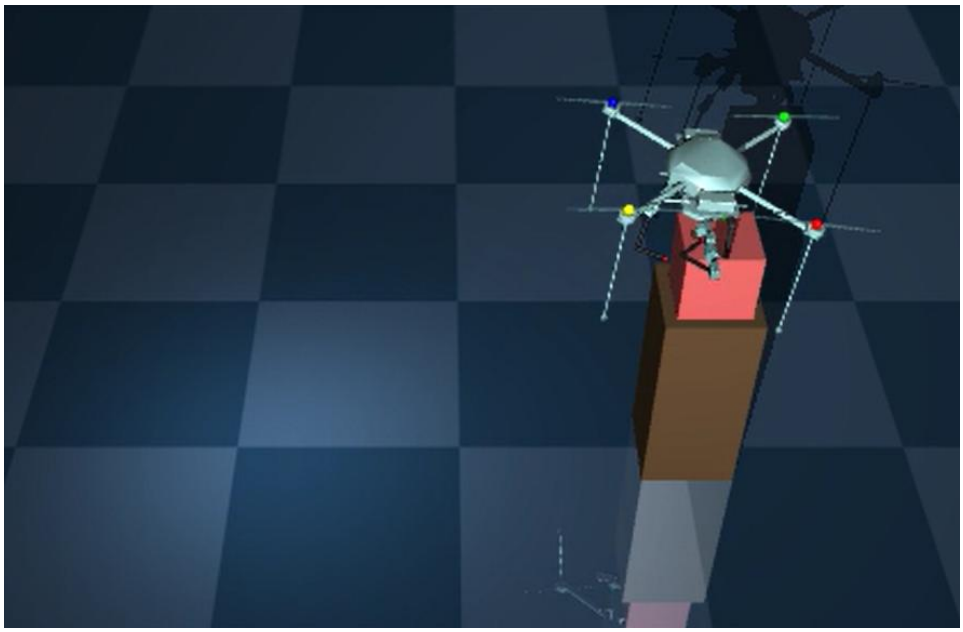
En las fases intermedias del diseño de la función de recompensa, se intentó guiar los brazos robóticos recompensando al agente por minimizar la distancia euclídea tridimensional absoluta entre la punta de los ganchos y el centro del asa. Esta estrategia dictaba una aproximación directa.

Bajo este esquema, la política descubierta por el algoritmo PPO consistía en mover el dron y los brazos en la línea recta más corta posible hacia el objetivo. Sin embargo, la cinemática de un gancho en forma de L hace que esta aproximación en línea recta sea mecánicamente inviable. Al volar directamente hacia el asa, la parte vertical del gancho (hook_L_vertical) o la propia punta colisionaban lateralmente contra la estructura de la caja o contra el pilar del asa antes de poder posicionarse por debajo.

En la simulación física gestionada por MuJoCo, este impacto lateral generaba fuerzas de reacción instantáneas que se transmitían a través de la cadena cinemática de los brazos hasta el chasis del dron, desestabilizando fatalmente el vuelo. Además, el agente terminaba "empujando" la caja fuera de su base en lugar de agarrarla, fracasando en el objetivo final de la tarea.



Aproximación 1: el dron se acerca a la caja objetivo



Aproximación 2: el chasis del dron se choca con el asa de la caja. Los ganchos chocan con la caja

4.2.3.2 La solución geométrica: la "maniobra de cuchara"

Para lograr un agarre exitoso con este tipo de efectores rígidos, la trayectoria no puede ser un vector directo. Requiere una secuencia geométrica específica en tres fases:

1. **Descenso y posicionamiento:** los ganchos deben situarse a una altitud (Z) estrictamente inferior a la del asa.

2. **Alineación coplanar:** los brazos deben abrazar el asa lateralmente, alineándose en el plano horizontal (XY).

3. **Elevación activa:** un movimiento puramente vertical hacia arriba para que la "L" del gancho enclave el asa desde abajo.

Esta trayectoria compleja se denomina comúnmente en manipulación robótica como la “maniobra de cuchara”. El reto en el contexto del DRL continuo es cómo enseñar esta secuencia sin programar una máquina de estados rígida, manteniendo una función de recompensa densa y diferenciable. En la versión final del entorno, esto se resolvió mediante una arquitectura matemática de recompensas condicionadas.

1. Desdoblamiento del objetivo

En lugar de un único punto central, la función de recompensa desdobla el objetivo en dos coordenadas virtuales independientes (t_left y t_right). A estos puntos se les aplica un *offset* geométrico de ± 0.12 metros en el eje Y respecto al centro del asa (target_handle). Esta separación matemática de 24 cm coincide intencionalmente con la anchura del asa, forzando al agente a abrir los brazos y buscar los extremos del cilindro, evitando colisiones centrales. La recompensa principal de precisión se calcula mediante una función exponencial de decaimiento (gaussiana) sobre la distancia media de ambos ganchos a estos objetivos:

$$R_{hooks} = 5.0 \cdot e^{-4.0 \cdot dist_{hooks}}$$

Esta formulación premia fuertemente la exactitud sin penalizar en exceso al dron cuando se encuentra lejos, estabilizando el gradiente.

2. Desplazamiento vertical

Para asegurar que la trayectoria fuerce al gancho a pasar por debajo del asa, se introdujo una modificación en el eje vertical de los objetivos. Se sumó artificialmente 0.05 metros a la coordenada Z de t_left y t_right (t_left[2] += 0.05). Al ubicar la máxima recompensa teórica ligeramente por encima del asa física, el agente se ve obligado a hacer que los ganchos atraviesen el plano geométrico del asa desde abajo hacia arriba, la esencia misma del movimiento de cuchara.

3. El trigger de subida

La parte más importante para inducir la maniobra de cuchara es la recompensa condicional basada en la proyección en el plano XY. El código calcula la distancia estrictamente horizontal entre los ganchos y sus objetivos (dist_xy_hooks).

Si el sistema detecta que los ganchos están alineados horizontalmente (tolerancia geométrica de $dist\_xy < 0.15$ m) y que la altitud del dron es lo suficientemente baja para que los ganchos estén bajo el asa (pos[2] < target_handle[2] + 0.1), se activa dinámicamente un nuevo incentivo de recompensa:

$$R_{scoop} = 2.0 \cdot v_z$$

Donde v_z es la velocidad vertical del cuadrotor (vel[2]). Esta sentencia matemática condicionada consigue algo muy importante y fundamental en este proyecto desde el punto de vista del RL, ya que le dice al agente que solo se le premiará por subir si, y solo si, sus ganchos están correctamente posicionados debajo de la carga.

4. Transición al agarre físico

Finalmente, una vez que el motor de simulación MuJoCo detecta un contacto físico válido entre las geometrías de colisión de los ganchos y el asa, la variable booleana good_grip se vuelve verdadera.

En este estado, el agente abandona la fase de precisión geométrica y la recompensa cambia su ponderación drásticamente: se otorga una inyección constante de $+5.0$ puntos por mantener el contacto, y la recompensa por velocidad vertical se multiplica ($+5.0 \cdot v_z$), incitando al dron a aplicar empuje máximo para vencer la gravedad de la caja y elevarla. Esta transición fluida entre la recompensa espacial (movimiento de cuchara) y la recompensa física (elevación de la caja) ha sido la clave teórica para resolver la complejidad del agarre autónomo.

4.2.4 Restricciones estéticas y cinemáticas

Además de los problemas e inconvenientes mencionados anteriormente, en el entrenamiento de sistemas robóticos redundantes —aquellos que poseen más grados de libertad de los estrictamente necesarios para posicionar el efector final en el espacio tridimensional— surge un desafío fundamental conocido teóricamente como el problema de la cinemática inversa mal planteada. Al disponer de múltiples combinaciones articulares para alcanzar un mismo punto objetivo $[x, y, z]^T$, el algoritmo PPO explorará libremente el espacio nulo del Jacobiano del manipulador.

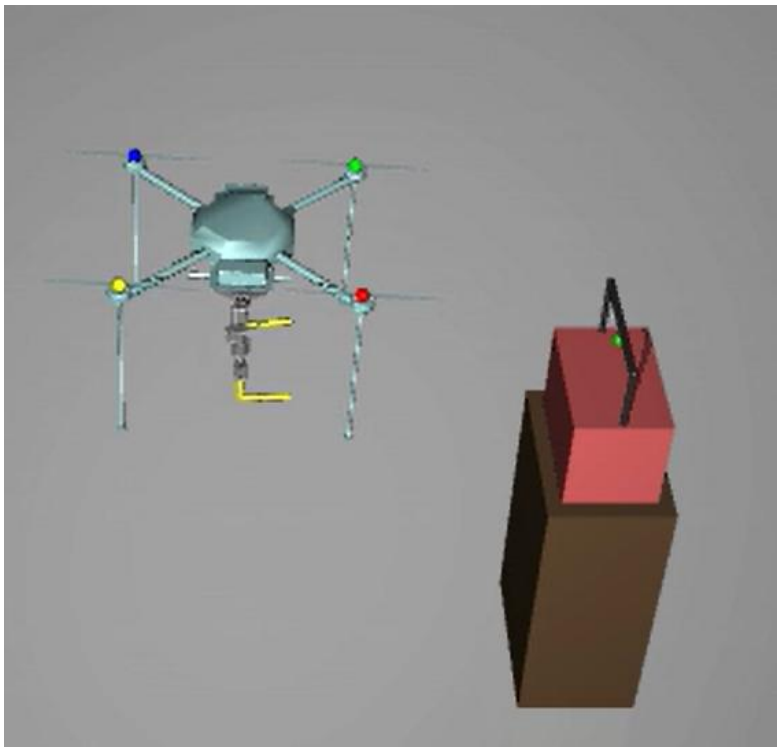
Si la función de recompensa se limita a premiar exclusivamente la precisión posicional (acercar el gancho al asa), el agente tenderá a descubrir configuraciones articulares que, si bien son matemáticamente válidas para la red neuronal, resultan físicamente aberrantes, ineficientes o estructuralmente peligrosas para su homólogo real. Durante las iteraciones de entrenamiento en el entorno simulado, se identificaron varios de estos comportamientos anómalos, cuya mitigación requirió la implementación de un sistema de regularizaciones cinemáticas (penalizaciones de comportamiento).

A continuación, se analizan los modos de fallo conductual detectados y la formulación matemática introducida para esculpir una política de control que sea no solo efectiva, sino también estética, predecible y segura.

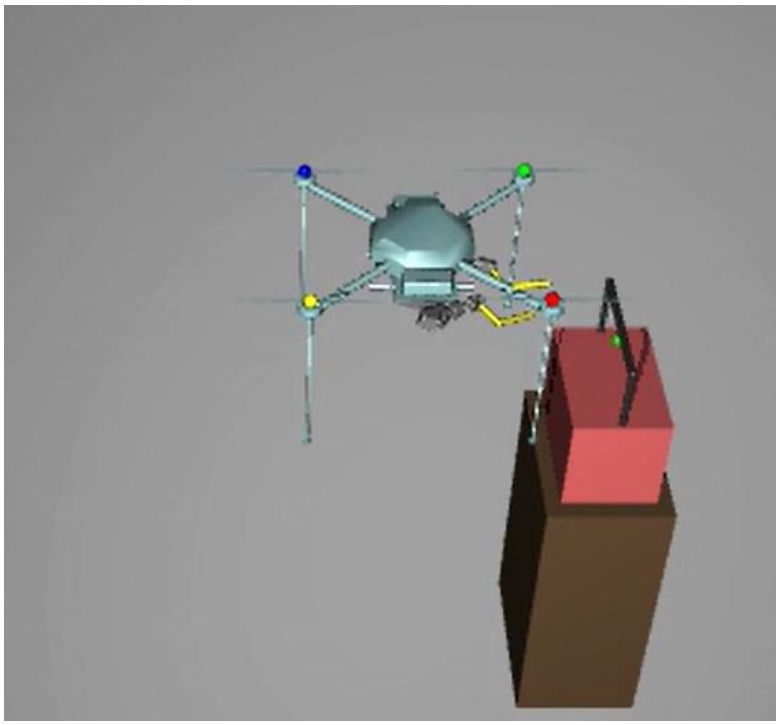
4.2.4.1 El problema de la retracción excesiva

El comportamiento emergente más notable descubierto por el agente de RL fue un problema estético, en el que el dron retraía demasiado los brazos robóticos, algo que, además de resultar poco estético, podría ocasionar problemas en un dron real. Dada la sensibilidad del cuadrotor a los cambios en su centro de masa, el algoritmo PPO aprendió rápidamente un principio básico de la dinámica de cuerpos rígidos: cuanto más extendidos estén los brazos (mayor radio de giro), mayor será el momento de inercia y mayor será el torque perturbador gravitacional ($\tau = \Delta p_{CoM} \times mg$) que desestabiliza el vuelo.

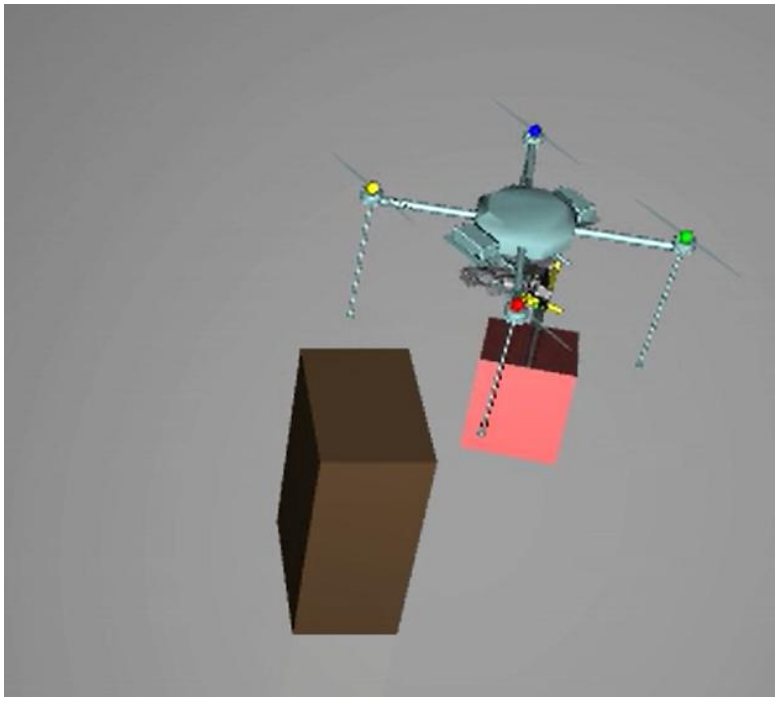
Para minimizar este esfuerzo de control aerodinámico, la política convergía invariablemente hacia una configuración donde el agente encogía las articulaciones de los hombros (q11, q12, q21, q22) y los codos (q13, q23) al máximo, pegando los ganchos al chasis del dron. Al intentar la maniobra de agarre en esta postura replegada, el dron se veía obligado a volar peligrosamente cerca de la caja objetivo, incrementando exponencialmente el riesgo de colisión de las hélices o el tren de aterrizaje contra la estructura externa.



Problema retracción 1: Situación inicial



Problema retracción 2: el dron se aproxima a la caja iniciando la retracción excesiva de los brazos



Problema retracción 3: el dron agarra la caja, pero con los brazos retraídos excesivamente

Para erradicar este comportamiento, se introdujo una penalización continua basada en la distancia vertical cuerpo-ganchos. El sistema calcula la distancia vertical promedio entre el centroide del dron y las puntas de los ganchos. La lógica impuesta dicta que los brazos deben operar extendidos, asumiendo una separación óptima de al menos 0.25 metros. Si el agente retrae los brazos por debajo de este umbral crítico, se le aplica un castigo proporcional a la infracción:

$$\text{Si } d_{body\_hooks} < 0.25 \Rightarrow R_{body\_hooks} = -2.0 \cdot \left(\frac{0.25 - d_{body\_hooks}}{0.25}\right)$$

Esta formulación lineal castiga severamente posturas hiper-retraídas, forzando al agente a asumir el coste aerodinámico de extender los brazos en favor de mantener un margen de seguridad adecuado entre las hélices y el obstáculo.

4.2.4.2 Control de apertura lateral (guía del eje Roll)

Un segundo problema geométrico radicaba en la separación lateral de los efectores finales. El asa objetivo tiene una longitud longitudinal finita de 0.23 metros. Para que la maniobra de cuchara (descrita en la sección anterior) sea exitosa, los ganchos deben envolver el asa por sus extremos. En las primeras pruebas, el agente tendía a juntar demasiado los brazos en el centro (maximizando la colisión mutua de las geometrías de los ganchos) o a abrirlos de forma asimétrica.

Para corregir esto, se monitoriza continuamente la separación en el eje Y entre ambos ganchos. Sabiendo que la separación ideal ronda los 0.24 metros (ligeramente superior a la longitud del asa), se implementó una función de castigo asimétrica que solo penaliza si el agente cruza o estrecha demasiado los brazos por debajo de 0.22 metros:

$$\text{Si } sep_y < 0.22 \Rightarrow R_{sep} = -2.0 \cdot \left(\frac{0.30 - sep_y}{0.30}\right)$$

Esta regularización suave actúa como un "muelle virtual" que repele los ganchos entre sí si se acercan demasiado al centro, garantizando la amplitud necesaria para el abrazo bimanual.

4.2.4.3 Anti-rotación y rigidez articular (desviación de Yaw)

Desde una perspectiva puramente estética y de rigidez mecánica, las rotaciones sobre el eje longitudinal del brazo (Yaw) suelen ser indeseables durante el acercamiento, ya que disminuyen la rigidez estructural de la cadena cinemática frente a cargas verticales. El agente a menudo retorcía las articulaciones de los hombros y codos (específicamente q13 y q23, correspondientes al Elbow_Pitch_Structure) en ángulos extraños que, matemáticamente, lograban posicionar la punta del gancho en el objetivo, pero dejaban el brazo en una singularidad o en una postura débil para el levantamiento de carga.

Para inducir rectitud y naturalidad en el brazo, se extraen las posiciones angulares de las articulaciones de Yaw de los codos izquierdo y derecho para calcular una métrica de desviación total (yaw_deviation). A diferencia de las penalizaciones anteriores, aquí se diseñó un incentivo exponencial positivo:

$$\text{Si } \text{dev}_{yaw} > 0.3 \Rightarrow R_{yaw} = +2.0 \cdot e^{-5.0 \cdot \text{dev}_{yaw}}$$

El uso de la función exponencial significa que, si la desviación es máxima, el premio es insignificante. A medida que el agente aprende a mantener los codos rectos (desviación cercana a 0), la recompensa crece dramáticamente hacia un máximo de $+2.0$ puntos. Esta técnica de psicología conductual aplicada a redes neuronales demostró ser altamente eficaz para lograr posturas limpias y ortogonales.

4.2.4.4 Amortiguamiento activo y suavidad de comandos

Finalmente, se abordó el problema de la vibración. Una red neuronal que opera sin memoria a corto plazo (como las arquitecturas perceptrón multicapa estándar utilizadas en la base de PPO) puede emitir comandos que cambien de extremo a extremo (de -1 a $+1$) en fracciones de segundo. En simulación esto es instantáneo, pero en la realidad, exigir a un servomotor y a un rotor cambiar de dirección a 50 Hz podría destruir los engranajes y ocasionar graves daños en el dron.

Para forzar la suavidad operativa en el comportamiento estético, el entorno almacena en memoria la acción del paso de simulación inmediatamente anterior (self.last_action). Posteriormente, penaliza severamente la derivada discreta (la tasa de cambio) de los comandos enviados por el agente:

$$R_{smooth} = -0.3 \cdot (|\Delta a_{vuelo}| + |\Delta a_{brazos}|)$$

Adicionalmente, se penaliza directamente la energía cinética de las articulaciones de los brazos mediante la norma de sus velocidades, restando $0.01 \cdot \text{vel}_{arm}$ de la recompensa. Este término actúa como un freno viscoso virtual que suprime los temblores espasmódicos, garantizando que cuando el dron se alinee frente a la caja objetivo, sus manipuladores robóticos se muevan con la fluidez y precisión milimétrica necesaria para una manipulación apropiada.

4.2.5 Restricciones estéticas y cinemáticas

A pesar de haber logrado estabilizar el vuelo, regularizar la cinemática de los brazos y definir la trayectoria de aproximación mediante la "maniobra de cuchara", las versiones intermedias del entrenamiento revelaron un último y crítico modo de fallo conductual justo en el instante de la transición entre el vuelo libre y la manipulación física.

El síntoma observado en la simulación era paradójico: el cuadrotor navegaba con elevada precisión, alineaba sus brazos robóticos correctamente y realizaba el descenso colocándose exactamente debajo del asa de la caja objetivo; sin embargo, en el rango muy cercano al asa de la caja, el dron se detenía abruptamente y se mantenía en vuelo estacionario milímetros por debajo de la carga, rehusando enclavar los ganchos y levantarla. El agente parecía tener un muro invisible que le impedía tocar el objeto, fallando sistemáticamente la misión a pesar de su excelente precisión posicional.

El diagnóstico de este comportamiento requirió un análisis bastante profundo de la intersección entre la definición geométrica del entorno en MuJoCo y la función de distancia programada en Gymnasium.

4.2.5.1 Diagnóstico del error: desacoplamiento entre referencia y colisión

En la formulación del RL continuo, los agentes no "ven" los objetos sólidos como los percibe un ser humano a través del renderizado visual; perciben distancias matemáticas hacia coordenadas espaciales. En el diseño de la caja objetivo, la estructura física del asa con la que los ganchos deben colisionar está definida por la primitiva geométrica <geom name="handle_bar" ... pos="0 0 0.34" .../>. Esto sitúa el núcleo físico del asa a una coordenada local de $Z = 0.34$ metros respecto al centro de la caja.

Sin embargo, para calcular la distancia de los ganchos al objetivo, el entorno utilizaba un marcador de referencia visualizado como una esfera verde translúcida, definido mediante la etiqueta <site name="target_site" pos="0 0 0.29" .../>. Este punto de referencia (site) estaba situado intencionalmente 5 centímetros por debajo del asa física para guiar a los ganchos hacia la zona inferior de la misma.

Desde la perspectiva del algoritmo PPO, el objetivo a minimizar era la distancia euclídea entre los

ganchos y este target_site. La red neuronal optimizó su política hasta lograr un error de distancia cercano a cero. En este estado, los ganchos se encontraban exactamente en $Z = 0.29$ metros.

El problema radicaba en el gradiente de recompensa. Si desde esa posición óptima el agente decidía aplicar empuje vertical para subir hacia el asa física ($Z = 0.34$), la distancia respecto al target_site ($Z = 0.29$) comenzaba a aumentar matemáticamente. Dado que la función de recompensa penalizaba el alejamiento ($R_{hooks} \propto e^{-dist}$), el agente recibía un castigo numérico inmediato por intentar agarrar la caja. La inteligencia artificial, actuando de forma lógicamente correcta respecto a su función objetivo, aprendió que "tocar el asa reduce la recompensa", generando el estancamiento estacionario.

4.2.5.2 La corrección geométrica

Para solucionar este conflicto destructivo entre la geometría de colisión y el centroide de recompensa, era necesario realinear los gradientes de manera que la reducción del error posicional garantizara inevitablemente la penetración física de las mallas de colisión. En la versión final integrada del código, se implementó un offset vertical artificial directamente sobre el cálculo de las metas locales de los ganchos.

El código extrae inicialmente la posición del marcador de referencia y, en lugar de usarlo como objetivo directo, aplica una transformación tridimensional:

$$t_{left} = p_{site} + [0; 0.12; 0.05], \quad t_{right} = p_{site} + [0; (-0.12); 0.05]$$

Como se detalla en el script de simulación: t_left[2] += 0.05 y t_right[2] += 0.05.

Esta simple adición de 0.05 metros es matemáticamente profunda: eleva el punto de máxima recompensa desde $Z = 0.29$ hasta $Z = 0.34$ en el marco local de la caja. Este nuevo punto coincide exactamente con el núcleo del cilindro físico del asa (handle_bar).

Al ubicar el objetivo de recompensa dentro del material sólido de la caja, se obliga al dron a atravesar el plano del asa desde abajo hacia arriba. Dado que los ganchos de los brazos son evaluados por el motor de colisiones continuas de MuJoCo (condim="4" para contactos precisos con fricción rotacional), es físicamente difícil que los ganchos alcancen el objetivo de recompensa máxima sin antes chocar de forma ascendente contra la parte inferior del asa.

C. Transición discreta: el trigger del agarre (good grip)

Esta corrección geométrica actúa en perfecta sinergia con el sistema de detección de colisiones de bajo nivel que se implementó en el código para poder detectar cuando se producía un contacto. Al forzar la colisión física como único medio para maximizar la recompensa posicional, el agente consigue desencadenar el contacto entre los identificadores handle_geom_id y good_contact_geoms.

En el instante exacto en que MuJoCo resuelve la ecuación de fuerza normal entre estas dos geometrías, la variable good_grip se evalúa como True. Esto activa un gran bonus de recompensa: un cambio radical de estado.

$$\text{Si } good\_grip = True \Rightarrow R_{instant} \mathrel{+}= 5.0 + 5.0 \cdot v_z$$

Una vez que el sistema verifica el agarre, la recompensa posicional por proximidad deja de ser la máxima prioridad. La función de recompensa inyecta un premio estático masivo de $+5.0$ que eclipsa

cualquier pequeña penalización geométrica y, lo más importante, premia agresivamente la velocidad vertical positiva ($5.0 \cdot v_z$). El dron, liberado de la necesidad de mantener posiciones estacionarias, satura los actuadores verticales para iniciar el levantamiento de la carga.

Si el conjunto dron-brazos logra elevar la masa de 1.0 kg de la caja objetivo a una altura superior a 0.05 metros respecto a su posición inicial (is_lifted = lift_height > 0.05), entra en el régimen final de recompensa continua. En esta fase, el agente recibe sumas extraordinarias proporcionales a la altura de elevación ($30.0 + \text{lift\_height} \cdot 200.0$).

La resolución de este importante problema demostró que, en el control de manipuladores aéreos mediante DRL, una formulación espacial ligeramente desalineada respecto a la realidad física de las mallas de colisión puede arruinar por completo una política entrenada. El ajuste de objetivos geométricos permitió cerrar la brecha final, convirtiendo un vuelo preciso pero inútil en una manipulación autónoma robusta y exitosa.

4.3 Configuración del entrenamiento

La transición de un entorno simulado teóricamente sólido a una política de control funcional depende casi en su totalidad de la configuración del algoritmo de optimización y de la canalización de los datos. Para la implementación de este proyecto, se descartó la programación de algoritmos de bajo nivel desde cero en favor de la librería Stable-Baselines3 (SB3), construida sobre el framework PyTorch. Esta decisión responde a la necesidad de utilizar implementaciones estables y matemáticamente verificadas de algoritmos del estado del arte, permitiendo focalizar el esfuerzo de ingeniería en el diseño del entorno y el ajuste de hiperparámetros.

El entrenamiento se orquestó mediante un script principal en python, apoyado por un archivo de configuración externo que facilita la iteración rápida sin modificar el código fuente. A continuación, se desglosan las decisiones de diseño arquitectónico y paramétrico que permitieron la convergencia del agente.

4.3.1.1 Vectorización y normalización asimétrica del entorno

En el aprendizaje profundo, las redes neuronales son extremadamente sensibles a la escala de los datos de entrada. Si una variable de estado (como la altitud en metros) tiene un rango de $[0,3]$, y otra (como la velocidad angular de los motores) opera en cientos de radianes por segundo, los gradientes de la red se desequilibran, paralizando el aprendizaje.

Para solucionar esto, el entorno se encapsuló en un contenedor DummyVecEnv y se pasó por el filtro VecNormalize de SB3. Esta herramienta mantiene un registro de la media y la varianza móvil de todas las observaciones, normalizándolas en tiempo real para que sigan una distribución estándar ($\mu = 0, \sigma = 1$). Además, se aplicó un truncamiento (clip_obs=10.0) para evitar que picos numéricos en la simulación física desestabilizaran los pesos de la red.

Sin embargo, durante las pruebas se tomó una decisión de diseño contraintuitiva pero vital para el éxito de la tarea: desactivar la normalización de la recompensa (norm_reward=False). Por defecto, muchas implementaciones de RL normalizan las recompensas para estabilizar el valor de la función de ventaja. No obstante, la función de recompensa diseñada en este proyecto (Sección 4.2) posee una estructura fuertemente escalonada: premios de $+1$ o $+2$ para la aproximación, $+5$ por el contacto, y un salto masivo de $+8000$ por levantar la caja (recompensa final de éxito). Si VecNormalize aplasta estos valores dinámicamente, el agente pierde la noción de magnitud absoluta, por lo que para la red, agarrar

la caja empieza a parecer estadísticamente similar a simplemente acercarse a ella. Desactivar esta normalización garantizó que los premios mayores conservaran su peso en la actualización de la política.

4.3.1.2 Hiperparámetros del algoritmo PPO

El PPO es robusto por naturaleza, pero el control continuo de un cuadrotor requiere un ajuste fino de su dinámica de exploración y actualización. Los parámetros finales se configuraron de la siguiente manera:

1. Horizonte y tamaño de Batch (n_steps=4096, batch_size=512):

El parámetro n_steps define cuántas transiciones recoge el agente en el entorno antes de actualizar la red. Se fijó en 4096 pasos. Considerando que el episodio máximo dura 1500 pasos (max_steps = 1500), esto asegura que cada recolección de datos (Rollout) contenga trayectorias completas, desde el despegue hasta el intento de agarre, proporcionando al algoritmo una visión global del problema. Este conjunto de datos se divide en minibatches de 512 muestras para calcular los gradientes, un tamaño suficientemente grande para estimar la dirección del gradiente con baja varianza, pero lo bastante pequeño para mantener la eficiencia estocástica en la memoria.

2. Optimización y prevención del colapso (n_epochs=7, target_kl=0.05):

En PPO, una vez recolectados los datos, la red se actualiza varias veces sobre el mismo conjunto (Epochs). Inicialmente, se usaban valores altos (10-12 iteraciones), lo que provocaba un sobreajuste drástico (lo que se conoce como overfitting) a las trayectorias recientes, destruyendo políticas de vuelo previamente aprendidas. Se redujo n_epochs a 7 (y en algunas iteraciones del código a 8) para suavizar la asimilación de conocimiento.

Como medida de seguridad adicional, se introdujo el parámetro target_kl = 0.05. Este mecanismo calcula la divergencia de Kullback-Leibler entre la política vieja y la nueva durante las épocas. Si la red se está actualizando demasiado rápido y cambia su comportamiento más de un 5% de un paso al siguiente, el entrenamiento aborta prematuramente esa fase de optimización. Esto fue clave para evitar que una mala racha de exploración (por ejemplo, el dron chocando repetidamente por azar) destruyera los pesos de una red que ya sabía volar.

3. Exploración vs. explotación (ent_coef=0.008):

El coeficiente de entropía obliga a la red a mantener cierta aleatoriedad en sus decisiones. Para tareas de control robótico suele fijarse en órdenes más pequeños al usado en este proyecto, sin embargo, debido a la tendencia del agente a estancarse en mínimos locales (como quedarse en estático por miedo a acercarse a la caja), se inyectó una entropía moderada de 0.008. Esto forzó al dron a seguir intentando micro-movimientos oscilatorios con los brazos incluso cuando creía haber encontrado una postura "segura", permitiéndole descubrir accidentalmente la maniobra de cuchara, por ejemplo.

4. Arquitectura de la red neuronal (Actor-Crítico):

Se utilizó un Perceptrón Multicapa (Multi-Layer Perceptron, MLP por sus siglas en inglés) desacoplado para el Actor (que emite las acciones) y el Crítico (que predice el valor del estado). Ambos comparten una topología densa de dos capas ocultas con 256 neuronas cada una (net_arch=dict(pi=[256, 256], vf=[256, 256])).

Dos detalles arquitectónicos marcaron la diferencia en la estabilidad inicial:

- **Activación tanh:** a diferencia de ReLU, la tangente hiperbólica está acotada entre $[-1,1]$, alineándose de forma natural con el espacio de acción normalizado de Gymnasium.
- **Inicialización de la desviación estándar (log_std_init=-1):** en control continuo, la acción se muestrea desde una campana de Gauss. Por defecto, SB3 inicia con un $\sigma = 1.0$, lo que provocaba que en el primer episodio el dron metiera los motores al 100% y se estrellara inmediatamente contra el suelo, sin generar datos útiles. Al fijar el logaritmo de la desviación estándar en -1 (lo que equivale a $\sigma \approx 0.36$), el dron comenzó sus primeros intentos con movimientos suaves y controlados.

4.3.1.3 Monitorización y callbacks

El ecosistema de entrenamiento no está completo sin una canalización de evaluación que permita juzgar el progreso de forma objetiva, aislada del ruido de la exploración y mientras se está llevando a cabo el entrenamiento, lo cual aporta una información muy importante en tiempo real de cómo se está desarrollando el mismo.

Para ello, se configuró un entorno paralelo puramente determinista (eval_env). A través del "EvalCallback", cada 50.000 pasos de simulación, el entrenamiento se pausa y el agente realiza 5 episodios de prueba donde siempre escoge la acción con mayor probabilidad (sin muestreo estocástico). Esto permite trazar las verdaderas curvas de aprendizaje en Tensorboard (herramienta de visualización de TensorFlow que permite monitorizar y analizar el entrenamiento de modelos, mostrando métricas, gráficos y estructuras de la red de forma interactiva, como por ejemplo la recompensa o longitud medias por episodio) y, de manera crucial, guardar automáticamente el modelo que obtiene la mayor recompensa, protegiendo el progreso frente a posibles degradaciones catastróficas posteriores.

Paralelamente, se implementó un "CheckpointCallback" para crear copias de seguridad de los pesos de la red y, muy importante, del estado interno del normalizador VecNormalize cada 100.000 pasos.

A modo de síntesis de la configuración paramétrica del algoritmo y la arquitectura de la red neuronal detallada, se presenta la siguiente tabla resumen. Esta tabla compila los hiperparámetros definitivos extraídos de los scripts de configuración (config.yaml) y entrenamiento (train.py), proporcionando una referencia rápida y estandarizada para la replicabilidad del experimento:

Tabla 2: Hiperparámetros entrenamiento 1g

Hiperparámetro	**Valor**	**Descripción**
total_timesteps	3×10^6	Pasos de simulación totales para completar el entrenamiento.
learning_rate (α)	3×10^{-4}	Tasa de aprendizaje constante para el optimizador

Hiperparámetro	**Valor**	**Descripción**
		(Adam).
n_steps	4096	Número de pasos recolectados (Rollout) antes de actualizar la red.
batch_size	512	Tamaño del subconjunto de datos (minibatch) para el cálculo del gradiente.
n_epochs	$7 - 8$	Iteraciones de optimización realizadas sobre el mismo conjunto de datos recolectado.
gamma (γ)	0.99	Factor de descuento; prioriza la estabilidad a largo plazo sobre recompensas inmediatas.
gae_lambda (λ)	0.95	Factor de decaimiento para la Estimación de Ventaja Generalizada (GAE).
clip_range (ϵ)	0.2	Rango de recorte de la función objetivo de PPO para evitar cambios de política drásticos.
ent_coef	0.008	Coeficiente de entropía; inyecta estocasticidad para evitar mínimos locales.
vf_coef	0.5	Ponderación de la función de pérdida del Crítico (Value Function) frente al Actor.
max_grad_norm	0.5	Umbral de recorte para la norma del gradiente (mitiga

Hiperparámetro	**Valor**	**Descripción**
		gradientes explosivos).
target_kl	0.05	Límite máximo de divergencia de Kullback-Leibler para abortar una época destructiva.
net_arch (pi, vf)	[256, 256]	Topología densa (MLP) de dos capas ocultas para las redes del Actor y del Crítico.
activation_fn	Tanh	Función de activación; acota las salidas intermedias y favorece la normalización continua.
log_std_init	-1.0	Desviación estándar inicial reducida ($\sigma \approx 0.36$) para un arranque de exploración suave.

5 ADAPTACIÓN Y DISEÑO DEL ENTORNO EN MICROGRAVEDAD (0G)

Una vez consolidado y validado el modelo de control para el entorno bajo gravedad terrestre (1g), el presente capítulo aborda la evaluación del sistema en el régimen de microgravedad (0g), un aspecto distintivo de este proyecto que altera fundamentalmente las ecuaciones de movimiento. En este escenario de simulación espacial, el vector gravitacional es nulo, lo que provoca que la dinámica del conjunto dron-brazos cambie drásticamente respecto a las fases de entrenamiento anteriores.

En ausencia de peso, el multi-rotor no necesita aplicar un empuje vertical continuo para mantener el vuelo estacionario. Si bien esta condición libera a los actuadores de la carga aerodinámica de sustentación, introduce un nuevo desafío de control severo: la inercia mecánica se convierte en el factor dominante y el sistema es mucho más sensible a las perturbaciones generadas por el movimiento de los brazos, ya que deja de existir una fuerza restauradora gravitacional. En un entorno de vacío simulado, cualquier par de torsión generado por los efectores finales o cualquier impulso de aproximación no frenado a tiempo se traduce en una deriva traslacional o en un giro descontrolado perpetuo.

Bajo este nuevo paradigma físico, el cuadrotor deja de comportarse funcionalmente como un helicóptero para operar como un vehículo espacial de base flotante libre de seis grados de libertad. Para lograr que el PPO lograse la convergencia resolviendo la tarea de agarre autónomo, fue indispensable acometer una reingeniería casi total del entorno simulado.

A lo largo de este capítulo se desgrana este proceso de adaptación. En primer lugar, se detallan las modificaciones estructurales en el motor de física MuJoCo (Sección 5.1). Seguidamente, se analiza críticamente el intento de emplear metodologías de transferencia de conocimiento (Transfer Learning) desde la política optimizada en 1g, justificando el descarte de esta vía a favor de un entrenamiento iterativo desde cero (Sección 5.2). A continuación, se expone el rediseño del sistema de actuación de bajo nivel y la consecuente reescritura de la función de recompensa para penalizar activamente la conservación del momento (Sección 5.3). Finalmente, se documentará la configuración hiperparamétrica que permitió estabilizar el aprendizaje en ingravidez (Sección 5.4).

5.1 Reestructuración del modelo físico y simulación

La transición de un entorno de simulación terrestre a uno espacial exige una adaptación rigurosa que trasciende la simple anulación del vector gravitacional en el motor físico. Para garantizar la fidelidad de la dinámica multicuerpo y asegurar que el agente aprenda políticas de control viables, ha sido necesario someter el archivo de definición del entorno (XML) de MuJoCo a una profunda reestructuración geométrica y matemática. En este nuevo régimen operativo, el sistema queda gobernado exclusivamente por su propia inercia y por las fuerzas de interacción de los contactos rígidos, desapareciendo por completo el efecto estabilizador del peso y la resistencia aerodinámica.

En consecuencia, el presente subcapítulo documenta las alteraciones implementadas a bajo nivel en el simulador para emular con precisión las condiciones de vacío y microgravedad. En las siguientes subsecciones se desgranará, en primer lugar, la reconfiguración de los parámetros globales del entorno integrador y la redefinición matemática de las dinámicas de contacto (Sección 5.1.1); y, en segundo

lugar, se analizará el tratamiento aplicado a las fuerzas disipativas de los actuadores y las propiedades inerciales, factores que cobran un protagonismo crítico ante la ausencia de arrastre fluido (Sección 5.1.2).

5.1.1 Modificaciones en la física del XML

La fidelidad de la simulación en robótica depende intrínsecamente de cómo el motor computacional resuelve las ecuaciones de Newton-Euler y las restricciones de contacto. Al trasladar el sistema acoplado cuadrotor-LiCAS A1 a un entorno de microgravedad, no basta con simplemente anular la aceleración gravitacional. El vacío y la ingravidez alteran de forma radical la manera en que los cuerpos rígidos colisionan, se rozan y transfieren energía cinética. En un escenario bajo gravedad terrestre (1g), el propio peso del dron y de la caja objetivo actúa como un amortiguador natural frente a las perturbaciones, manteniendo los objetos asentados sobre las superficies y facilitando el cierre de la cadena cinemática durante el agarre. Sin embargo, en 0g, cualquier fuerza de contacto genera una reacción puramente inercial, lo que exige una reconfiguración profunda del archivo de definición del modelo físico (XML) en MuJoCo para garantizar la estabilidad numérica y evitar comportamientos divergentes.

A continuación, se desglosan las modificaciones estructurales y paramétricas implementadas en la transición del modelo base hacia su iteración espacial final.

5.1.1.1 Anulación de variables ambientales y cambio de integrador numérico

El primer paso en la adaptación consistió en la redefinición del bloque de opciones globales <option>. Se estableció el vector gravitacional en cero ($g = [0,0,0]^T$), pero, además, para emular las condiciones del vacío orbital, se anularon los parámetros de densidad y viscosidad del medio (density="0", viscosity="0"). Esta eliminación del arrastre aerodinámico tiene un impacto profundo en la dinámica de vuelo: el dron ya no sufre pérdida de velocidad por la fricción del aire, convirtiendo la tarea de control en un problema de estabilización de un vehículo espacial con inercia pura.

Bajo estas condiciones no disipativas, la conservación del momento lineal y angular se vuelve matemáticamente crítica. El integrador numérico utilizado por defecto en fases previas (“implicitfast”) presentaba ligeras derivas de energía (ganancia o pérdida espuria de momento) tras miles de pasos de simulación sin gravedad. Para subsanar esta inestabilidad, se sustituyó por el algoritmo de Runge-Kutta de cuarto orden (integrator = "RK4"). Aunque el método RK4 tiene un coste computacional mayor, garantiza una mayor fidelidad en la integración temporal de las ecuaciones diferenciales, acoplado a un paso de simulación estricto de 2 milisegundos (timestep = 0.002), lo cual es indispensable para resolver los contactos rígidos sin que los objetos salgan despedidos de forma explosiva al interactuar en ingravidez.

5.1.1.2 Dinámica de contacto suave: solref y solimp

El modelado del contacto en MuJoCo se desmarca de los enfoques tradicionales de restricciones duras, empleando en su lugar una formulación de optimización convexa basada en "contactos suaves". Esta suavidad está gobernada por dos parámetros fundamentales que fueron reajustados para los efectores finales y la carga en 0g: la referencia del solver (solref) y la impedancia del solver (solimp).

En el bloque de valores por defecto (<default>), se creó una clase específica denominada “manipulable” aplicada a la geometría del asa. En ingravidez, al insertar los ganchos bajo el asa, no existe el peso de la caja de 1 kg "tirando" hacia abajo. Si el contacto es demasiado rígido, la mínima penetración geométrica calculada en un paso de tiempo genera una fuerza repulsiva gigante que dispara

la caja en dirección opuesta. Para evitar esto, se configuraron los siguientes valores:

Impeditividad (solimp="0.99 0.995 0.001"): este parámetro define una función de transición que dicta cómo de "duro" se vuelve el contacto a medida que aumenta la penetración. Al fijar el valor base inicial en 0.99, se garantiza que la superficie del asa se perciba extremadamente sólida desde el primer instante del toque físico, evitando que los ganchos se hundan irrealmente en la malla cilíndrica.

Referencia temporal (solref="0.002 1"): el primer valor establece la constante de tiempo (en segundos) que tarda el simulador en resolver la restricción de contacto. Al reducirlo a 0.002 s (coincidente con el timestep), se instruye al motor para que absorba y disipe la fuerza de penetración casi instantáneamente, amortiguando los rebotes elásticos que arruinarían el agarre en vacío.

5.1.1.3 Dimensionalidad de la fricción (condim) y márgenes de colisión (margin)

El éxito de la maniobra de agarre no solo depende de la fuerza normal, sino, críticamente, de la fricción. En el XML, se definió explícitamente el parámetro condim="4" para las interacciones de los ganchos y el asa. Una dimensión de contacto de 4 implica que el solver no solo calcula la fricción de deslizamiento tangencial (como en un modelo estándar de Coulomb), sino que también computa la fricción rotacional (torsión y rodadura). Dado que en 0g el dron puede aplicar fuerzas omnidireccionales, la falta de gravedad que fije la caja contra los ganchos hace que esta sea extremadamente propensa a pivotar o resbalar lateralmente por el interior de la "L" de los efectores. La configuración del vector de fricción friction="1.0 0.05 0.01" garantiza un coeficiente de fricción estática altísimo (1.0), complementado con resistencia a la torsión (0.05) y a la rodadura (0.01), "bloqueando" eficazmente el asa dentro del gancho una vez que el control de posición de los servomotores aplica fuerza.

Finalmente, para dotar al algoritmo PPO de un margen de maniobra numérico viable durante la transición de vuelo libre a contacto sólido, se implementó un margen perimetral virtual (margin="0.0005") en la clase manipulable. Esta instrucción obliga a MuJoCo a activar la detección de la matriz de colisión y comenzar a calcular fuerzas repulsivas suaves medio milímetro antes de que las mallas geométricas tridimensionales se crucen físicamente. En 0g, este colchón de medio milímetro actúa como una capa de amortiguamiento, permitiendo que el controlador detecte el incremento de fuerza y el agente PPO asimile la entrada en la fase de manipulación física antes de que una colisión dura contamine las observaciones de velocidad con ruido.

5.1.2 El desafío del vacío: impacto del damping y la inercia rotacional

La supresión de la atmósfera en el simulador mediante la anulación de la densidad y la viscosidad del medio introduce una física radicalmente distinta a la experimentada en la primera fase del proyecto. En un régimen de gravedad terrestre (1g), la interacción geométrica de los eslabones del robot con el aire circundante genera un arrastre aerodinámico (drag) que actúa como una fuerza disipativa continua. Esta resistencia fluida, que escala con el cuadrado de la velocidad, ejerce un efecto de frenado pasivo que amortigua las oscilaciones de los brazos robóticos y ayuda a estabilizar la plataforma aérea tras la ejecución de maniobras agresivas.

Al transicionar al vacío absoluto simulado en el entorno de microgravedad, este sumidero de energía desaparece por completo. De acuerdo con la primera ley de Newton, cualquier impulso cinético impartido al sistema —ya sea una traslación del centro de masa o una rotación de los eslabones— se conservará indefinidamente a menos que una fuerza externa o interna se oponga a

él. Esta invariancia del momento somete al algoritmo de Aprendizaje por Refuerzo (RL) a un escenario de control extremadamente hostil: un comando de motor que en 1g apenas generaría un pequeño desplazamiento compensado por el aire, en 0g provoca una deriva perpetua que termina arruinando la alineación geométrica necesaria para el agarre de la caja.

Para dotar al modelo de viabilidad física y permitir la convergencia de la red neuronal de PPO, fue estrictamente necesario modelar con alta fidelidad las propiedades mecánicas internas de los actuadores del sistema LiCAS A1, concretamente su amortiguamiento viscoso (damping) y su inercia rotacional equivalente (armature), configurados en la clase <default class="servo_arm"> del archivo XML.

Modelado de la inercia rotacional del rotor (armature)

En la robótica física, los servomotores están compuestos por un tren de engranajes reductor y un motor de corriente continua cuyo rotor interno gira a altas velocidades. Aunque la masa de este rotor es pequeña, al estar sometido a una alta relación de reducción, su momento de inercia aparente "reflejado" en el eje de la articulación es considerable. En MuJoCo, este fenómeno se modela a través del atributo armature.

En el archivo de definición final, se asignó un valor de armature = "0.02" a todas las articulaciones de los brazos manipuladores. Matemáticamente, este parámetro se suma directamente a los elementos de la diagonal principal de la matriz de masa (o matriz de inercia en coordenadas generalizadas) del robot. En un entorno 1g, el efecto del armature suele quedar enmascarado por la fuerza abrumadora de la gravedad. Sin embargo, en el espacio exterior, se convierte en un factor crítico de acoplamiento dinámico.

Cuando el agente de RL comanda una extensión rápida del brazo para alcanzar el asa de la caja, la aceleración de la articulación exige superar esta inercia rotacional del rotor. Según la tercera ley de Newton, el par necesario para acelerar el servo genera un par de reacción igual y opuesto sobre el cuerpo del dron (la base flotante). Al no existir sustentación aerodinámica que ancle el cuadrotor, este par de reacción induce inmediatamente una rotación indeseada (spinning) en el chasis principal. El agente de RL se vio obligado a descubrir, mediante millones de pasos de exploración estocástica, que cada movimiento articular en 0g conlleva un coste de actitud, debiendo emitir comandos compensatorios de par directamente sobre el dron de forma síncrona al movimiento del brazo para mantener el objetivo en su campo de acción.

Amortiguamiento mecánico interno (damping) y pérdida por fricción (frictionloss). El segundo pilar para la estabilización del entorno sin arrastre aerodinámico es la disipación activa de la energía cinética a nivel de la articulación. En el diseño del XML, las articulaciones de la clase servo_arm se configuraron con un coeficiente de amortiguamiento excepcionalmente alto (damping="2.0") y una fricción seca (frictionloss="0.1").

El parámetro damping implementa una ley de fricción viscosa lineal $\tau_{damp} = -c \cdot \dot{q}$, donde c es el coeficiente de amortiguamiento y $\dot{q}$ la velocidad angular de la articulación. La magnitud 2.0 no fue elegida de manera arbitraria; emula el freno electromagnético característico de los servomotores digitales reales cuando se les exige mantener una posición o cuando reducen su velocidad.

Sin este amortiguamiento severo, la red neuronal (cuyas salidas estocásticas inherentemente contienen ruido de alta frecuencia) induciría vibraciones incontrolables en los ganchos. Al carecer

de aire que frene el eslabón, cualquier exceso de energía cinética provocaría que el brazo sobrepasara su ángulo objetivo (overshoot) y comenzara a oscilar perpetuamente alrededor del setpoint. El alto valor de damping fuerza a las articulaciones a comportarse como un sistema mecánicamente sobreamortiguado. Esto obligó a la política del agente PPO a ser deliberada y constante en sus salidas: si el agente deja de aplicar esfuerzo de control, el brazo se detiene en el vacío debido a su propia fricción interna, facilitando enormemente la maniobra de "cuchara" milimétrica requerida para enclavar el asa.

Amortiguamiento artificial por software. Pese a la exhaustiva parametrización electromecánica de los servos, los ensayos preliminares revelaron que el ruido de la exploración matemática del agente seguía acumulando derivas microscópicas en el chasis del dron, alejándolo de la caja base a lo largo de los extensos episodios de 1500 timesteps.

Para resolver esta inestabilidad sin alterar la física central de MuJoCo, se implementó en la arquitectura del entorno de Gymnasium una atenuación cinemática discreta directamente sobre el vector de estado en cada frame skip. Se aplica un decaimiento multiplicativo a las velocidades lineales (self.data.qvel[0:3] *= 0.999) y angulares (self.data.qvel[3:6] *= 0.995).

Esta intervención algorítmica simula un "arrastre residual suave". Desde la perspectiva del RL, este ligero damping de software fue la clave final para suprimir los micro-temblores del chasis, permitiendo que la red neuronal correlacionara de manera determinista sus acciones de control de vuelo con recompensas geométricas densas, logrando así estabilizar el vuelo en flotación libre antes de proceder a la inserción de los ganchos en el objetivo.

5.2 El reto de la transferencia de conocimiento (Transfer Learning)

En el ámbito del DRL, el entrenamiento de agentes desde cero en entornos de alta dimensionalidad computacional es un proceso que consume una cantidad masiva de recursos y tiempo. Una vez validada empíricamente la convergencia de una política óptima en el simulador bajo gravedad terrestre (1g), la transición metodológica natural hacia el entorno espacial no consistía en desechar el modelo, sino en intentar reaprovechar las representaciones internas ya aprendidas por la red neuronal.

Este enfoque, conocido en la literatura de inteligencia artificial como transferencia de conocimiento (*Transfer Learning*), parte de la premisa de que muchas de las habilidades fundamentales adquiridas por el agente (como la codificación espacial de los cuaterniones, la percepción de las distancias relativas hacia el objetivo, o la coordinación cinemática básica para cerrar los ganchos) son invariantes respecto a la aceleración de la gravedad y pueden usarse como base para entrenar un agente en el entorno de microgravedad. Las siguientes subsecciones documentan y analizan críticamente el proceso de intentar aprovechar este bagaje conductual previo, evaluando las barreras físicas que convirtieron una hipótesis teórica en uno de los mayores retos técnicos del proyecto.

5.2.1 La hipótesis inicial: Intentos de aplicar fine-tuning a la política ganadora del entorno 1g

La hipótesis de partida para abordar el control en microgravedad se fundamentó en la técnica de *fine-tuning* (ajuste fino). El planteamiento dictaba que, al instanciar el nuevo entorno de 0g y alimentar al algoritmo PPO con los pesos del modelo multicapa Actor-Crítico que ya había resuelto

la tarea en 1g, el agente solo necesitaría adaptar marginalmente su política de vuelo, reduciendo drásticamente las épocas de exploración estocástica. Después de todo, el agente ya sabía cómo aproximarse a la caja y cómo ejecutar la delicada "maniobra de cuchara" para insertar los efectores finales bajo el asa.

Para llevar a cabo este experimento, se cargó el archivo comprimido del mejor modelo validado de la fase anterior empleando la librería Stable-Baselines3. El entrenamiento se reanudó configurando una tasa de aprendizaje significativamente menor a la original (3.5×10^{-4}). Esta reducción tenía como objetivo evitar el fenómeno del "olvido catastrófico" asegurando que los gradientes de las primeras colisiones en 0g no destruyeran bruscamente los pesos de la red neuronal preentrenada, permitiendo una adaptación suave a la nueva dinámica.

No obstante, la ejecución de las primeras trayectorias de evaluación del modelo preentrenado en el simulador de 0g reveló un modo de fallo conductual inmediato, sistemático e irrecuperable. Lejos de adaptar su vuelo, el cuadrotor se comportaba de manera errática y explosiva.

A. El sesgo del vuelo estacionario

El diagnóstico de este fallo requirió volver a las ecuaciones dinámicas fundamentales descritas en el Capítulo 2 de esta memoria. En el entorno de gravedad terrestre, la ecuación de traslación establece que la masa por la aceleración es igual a la suma del empuje vectorial, las fuerzas externas y la gravedad ($m\ddot{p} = f_T R e_3 - mg + f_{ext}$). Para mantener un simple vuelo estacionario, el agente de 1g había interiorizado en su red neuronal la necesidad absoluta de generar un empuje continuo ($f_T \approx mg$).

Al transferir esta política al entorno espacial, donde $g = [0,0,0]^T$, la dinámica cambia drásticamente a $m\ddot{p} = f_T R e_3 + f_{ext}$. La red neuronal, sesgada por millones de iteraciones de entrenamiento previo, emitía como primera acción un comando de empuje vertical para "no caerse". En ausencia de un peso que contrarrestar, este empuje continuo dejaba de ser una fuerza de sustentación para convertirse en una aceleración ascendente pura e infinita. En cuestión de fracciones de segundo, el dron salía disparado hacia la cota superior del simulador (eje Z), cruzando el límite penalizado del espacio de trabajo e interrumpiendo prematuramente el episodio iteración tras iteración.

B. Acoplamiento no lineal de la traslación lateral

El segundo síntoma del fracaso del *fine-tuning* se manifestó en los comandos de navegación en el plano XY. En la física atmosférica bajo gravedad, el desplazamiento lateral de un multirrotor está fuertemente acoplado a su actitud. Para avanzar, el dron debe inclinar su vector de empuje (cabeceo o *pitch* negativo), perdiendo la componente vertical del mismo, lo que a su vez le obliga a aumentar la potencia total de los motores para no perder altitud. La red preentrenada en 1g dominaba esta compleja coreografía de alabeo, cabeceo y compensación de empuje.

Sin embargo, en el archivo xml final, esta técnica es contraproducente. Inclinar el cuadrotor y aplicar potencia máxima sin un vector de gravedad que vectorice el movimiento resultaba en trayectorias parabólicas descontroladas o en un giro continuo en espiral. El agente intentaba aplicar correcciones mediante el lazo PID heredado, pero los errores de velocidad se acumulaban rápidamente, provocando la saturación de los actuadores y la inestabilidad de la matriz de rotación.

C. Transferencia negativa

Desde el punto de vista analítico del RL, se concluyó que el experimento había incurrido en un caso de "transferencia negativa". La política base no era un conocimiento subyacente generalizable, sino que estaba sobreajustada (*overfitted*) a una restricción ambiental muy específica: la fuerza de $9.81\ m/s^2$ en el eje Z negativo.

El esfuerzo computacional que la red requería para "desaprender" el miedo a caer, para dejar de compensar empujes laterales y para entender que la inercia dominaba ahora el sistema, era superior al esfuerzo de construir un mapeo de estados desde una inicialización de pesos aleatoria. Lejos de acelerar la convergencia, el *fine-tuning* ancló al optimizador en un mínimo local destructivo. Esta constatación empírica obligó a detener la línea de investigación de transferencia de conocimiento y justificó técnica y temporalmente la necesidad de desechar la arquitectura de control en cascada empleada hasta el momento, apostando por un rediseño desde los cimientos y un entrenamiento desde cero plenamente adaptado a las leyes físicas orbitales en 0g.

5.2.2 Análisis de fallos

La constatación empírica del fracaso del transfer learning exigió realizar un estudio de las trayectorias ejecutadas por el agente preentrenado al ser instanciado en el entorno de microgravedad. El análisis de los registros de simulación y el renderizado de los episodios evidenció que el modelo de la red neuronal en la fase anterior no solo era ineficaz, sino físicamente destructivo. Las estrategias que constituían el núcleo de la política óptima en la 1g no servían para nada en 0g.

A continuación, se diseccionan los tres modos de fallo conductual primarios que explican el colapso del agente, demostrando la incompatibilidad fundamental entre la heurística de vuelo atmosférico y la dinámica de base flotante en el vacío.

A. El colapso del paradigma cinemático

El mecanismo intrínseco de locomoción horizontal de un multirrotor bajo gravedad terrestre depende de una delicada ruptura del equilibrio de fuerzas. Para generar un desplazamiento lateral, el cuadrotor inclina su actitud (modificando los ángulos de pitch o roll), lo que proyecta una fracción del vector de empuje principal sobre el plano horizontal. Durante este proceso, la red neuronal de 1g aprendió a incrementar simultáneamente la potencia global de los motores para compensar la pérdida de la componente de sustentación vertical y evitar la pérdida de altitud.

En la simulación orbital ($g = [0,0,0]^T$), esta política de vuelo resulta aberrante. Al no existir una fuerza gravitatoria que el empuje vertical deba contrarrestar, la orden de inclinar el morro hacia abajo y aplicar potencia se traduce en una aceleración lineal pura y masiva en la dirección del eje Z local de la aeronave. El agente, al no percibir resistencia, continuaba incrementando los comandos espaciales esperando una estabilización aerodinámica que nunca llegaba, provocando que el sistema acelerase exponencialmente hasta abandonar el volumen de trabajo permitido del entorno en apenas unos cientos de milisegundos.

B. La ausencia de frenado aerodinámico y el retorno de la aproximación impulsiva orbital

En el Capítulo 4 se detalló la superación del problema de la aproximación impulsiva, un comportamiento donde el dron colisionaba a alta velocidad contra la estructura de la caja por no

aprender a reducir su energía cinética. La solución en 1g radicó en una penalización dinámica por velocidad que forzaba al agente a nivelar su actitud antes de llegar al objetivo. Al nivelar el dron en la Tierra, el empuje se vuelve puramente vertical, y la fricción del aire junto con la inercia residual se encargan de frenar suavemente el vehículo.

En microgravedad, la anulación paramétrica del medio disipativo (densidad y viscosidad) elimina cualquier sumidero externo de energía. Si un cuerpo adquiere una velocidad de $1.5\ m/s$, mantendrá esa velocidad indefinidamente a menos que se aplique una fuerza equivalente en sentido contrario. La red preentrenada en 1g carecía por completo del concepto de "retropropulsión". Su estrategia de aproximación consistía en acelerar hacia la caja y, a un metro de distancia, cortar los comandos de inclinación horizontal esperando que la fricción detuviera el avance. En el vacío, esta acción pasiva resultaba en impactos cinéticos severos contra la caja base, desestabilizando tanto la carga útil como el propio agente, y demostrando que la penalización de velocidad empleada en 1g era insuficiente para forzar el descubrimiento de maniobras de frenado activo en 0g.

C. Conservación del momento angular y la perturbación bimanual

El tercer fallo, y quizás el más complejo a nivel dinámico, derivó de la cinemática de los propios brazos manipuladores LiCAS A1. Según la tercera ley de Newton, los torques aplicados por los servomotores para mover las articulaciones generan torques de reacción opuestos sobre el cuerpo principal.

En la versión terrestre, el controlador PID de vuelo en cascada, apoyado por la fuerza restauradora que ejerce el vector de empuje para mantener el dron en el aire, absorbía eficientemente estas perturbaciones. El agente había aprendido que podía extender los brazos o realizar ajustes rápidos de Yaw en los codos instantes antes del agarre sin que la plataforma aérea sufriera desviaciones críticas.

En el espacio, la inercia se convierte en el factor dinámico dominante. Al estar el sistema mecánicamente aislado, el momento angular total debe conservarse. Cuando la política de 1g ordenaba una extensión simultánea y rápida de los brazos para abrazar la caja (la "maniobra de cuchara" aprendida), el chasis del dron sufría una contrarrotación inmediata y violenta para conservar el equilibrio de momento.

Este giro incontrolado alteraba por completo el marco de referencia sensorial del agente. En un solo paso de simulación, la posición relativa del objetivo y los cuaterniones de orientación cambiaban de manera drástica, inyectando observaciones caóticas a la red neuronal. El agente intentaba corregir este giro comandando correcciones asimétricas en los brazos y el empuje, lo que invariablemente introducía más energía rotacional al sistema cerrado, degenerando en un bucle de retroalimentación inestable que imposibilitaba cualquier intento de sincronización fina con el asa de la caja.

La suma de estas inestabilidades cinemáticas y dinámicas cerró definitivamente la puerta a la reutilización de la red neuronal previa, sentando las bases teóricas que justificaron la reingeniería total de la arquitectura de acción y de la función de recompensa abordadas en las siguientes secciones.

5.2.3 Cambio de paradigma: justificación del descarte del Transfer Learning

A la vista de las inestabilidades descritas en el apartado anterior, quedó patente que insistir en la adaptación de la política de 1g mediante fine-tuning no era una línea de trabajo viable. El problema de fondo no residía únicamente en una cuestión de ajuste de pesos neuronales, sino en que la propia formulación del problema de control era incompatible con las leyes físicas que rigen el movimiento en microgravedad.

La decisión de descartar la transferencia de conocimiento y optar por un entrenamiento desde cero se fundamentó en dos motivos principales: la incompatibilidad geométrica del espacio de acciones y el coste temporal del proceso de convergencia.

A. Incompatibilidad arquitectónica del espacio de control

En el entorno de gravedad terrestre, la arquitectura de control se diseñó en base a un esquema jerárquico. La red neuronal emitía un vector de 12 dimensiones, donde las 4 primeras componentes actuaban como referencias de velocidad (lineal y tasa de guiñada) para un controlador PID de bajo nivel, y las 8 restantes controlaban la posición de los servos de los brazos. Este esquema dependía del uso de un mezclador de motores clásico para mantener la actitud y contrarrestar el peso del dron.

Sin embargo, al analizar la dinámica del vuelo en el nuevo entorno simulado, se evidenció que intentar emular el comportamiento de un cuadrotor clásico en el vacío carece de sentido práctico. En microgravedad no existe la necesidad de mantener un empuje constante ni de inclinar la aeronave para generar un desplazamiento lateral. Como se documenta en las iteraciones finales del código, lo físicamente correcto y óptimo en un entorno orbital es gobernar el sistema mediante un control omnidireccional, asimilando el dron a un satélite o nave espacial.

Este cambio de enfoque obligó a modificar desde los cimientos el espacio de acciones de Gymnasium. Se eliminó el lazo PID intermedio y se sustituyó por una aplicación directa de fuerzas y pares sobre el centro de masa del vehículo utilizando la función *xfrc_applied* del motor MuJoCo, la cual permite aplicar directamente fuerzas y torques externos sobre el centro de masa de un cuerpo, sin pasar por actuadores ni controladores intermedios. De este modo, el simulador integra estas fuerzas en la dinámica del sistema como entradas físicas puras, equivalentes a empujes y momentos ejercidos desde el exterior. El nuevo espacio de acción pasó a tener 14 dimensiones:

- Fuerzas lineales: 3 variables continuas para comandar los empujes en los ejes X, Y y Z del mundo.
- Pares de torsión: 3 variables para comandar los torques de roll, pitch y yaw.
- Control de manipuladores: 8 variables manteniendo el control de posición de los servomotores.

Desde el punto de vista del aprendizaje profundo, esta reestructuración hace inviable el transfer learning. Una red neuronal entrenada para mapear estados a una capa de salida de 12 dimensiones orientadas a referencias de velocidad no puede transferir sus pesos a una arquitectura que exige 14 salidas orientadas a fuerzas y torques directos. La naturaleza física de los comandos es completamente distinta.

Justificación temporal frente al olvido catastrófico El PPO, aunque robusto, requiere una cantidad considerable de iteraciones (millones de pasos de simulación) para modificar comportamientos que han sido fuertemente arraigados durante el entrenamiento.

Al intentar adaptar la red de 1g, el algoritmo destinaba la mayor parte de la computación a intentar "desaprender" sus propios sesgos. Forzar a la red a olvidar la necesidad de aplicar un empuje constante en el eje Z y a dejar de usar el cabeceo (pitch) para avanzar estaba requiriendo un volumen de muestras mucho mayor que el necesario para enseñar las mecánicas básicas a una red "en blanco". El coste de combatir este sobreajuste (overfitting) a la gravedad terrestre superaba con creces el tiempo de cómputo de empezar el proceso desde cero.

En definitiva, la redefinición del problema de control aéreo hacia un control puro de fuerzas y torques hizo que la inicialización aleatoria de la red neuronal fuera la opción más eficiente. Esta decisión marcó un punto de inflexión en el desarrollo de esta segunda fase del TFG, obligando no solo a iniciar un nuevo ciclo de entrenamiento, sino a replantear la ingeniería de la función de recompensa para guiar al agente bajo este nuevo paradigma de actuación.

5.3 Reingeniería del control y la función de recompensa

El rediseño estructural del entorno de simulación y el cambio hacia un modelo de control omnidireccional (tipo nave espacial) invalidaron por completo la función de recompensa empleada durante la fase terrestre del proyecto. La función matemática de evaluación, que antes premiaba la navegación rápida y la estabilización horizontal, se enfrentaba ahora a un sistema dinámico radicalmente distinto: un vehículo de base flotante en el vacío, sin fuerzas restauradoras y con una inercia rotacional y traslacional crítica.

Para lograr que el algoritmo convergiera en estas nuevas condiciones, fue necesario replantear desde cero la estrategia de Reward Shaping. El objetivo dejó de ser simplemente guiar al cuadrotor hacia la carga superando la gravedad, para convertirse en el reto de enseñar a una red neuronal a gestionar la conservación de la cantidad de movimiento, obligándola a predecir la inercia del sistema y a ejecutar maniobras de frenado activo. Las siguientes subsecciones detallan la evolución matemática de estas nuevas penalizaciones y recompensas condicionales implementadas en la versión final del entorno, diseñadas específicamente para contrarrestar las patologías conductuales emergentes en microgravedad.

5.3.1 El control de la inercia

Uno de los primeros comportamientos anómalos documentados al iniciar el entrenamiento desde cero en el entorno espacial fue la reaparición, en una variante mucho más destructiva, del problema de la aproximación impulsiva. En la configuración inicial de la recompensa para 0g, se incluyó un incentivo lineal estándar para minimizar la distancia entre el dron y el punto de aproximación. La lógica de optimización de la red neuronal dictaminó, de forma matemáticamente correcta respecto a su función objetivo, que la manera de maximizar el retorno acumulado era aplicar la fuerza máxima permitida (50 N) de forma ininterrumpida hacia las coordenadas de la caja.

A. La física de la aproximación impulsiva orbital

En el entorno terrestre (1g), esta estrategia agresiva era mitigada parcialmente por el arrastre aerodinámico y por la propia maniobra de nivelación del dron, la cual redirigía el empuje hacia el eje vertical y actuaba como un freno natural. En el vacío simulado de la versión espacial, dictado por las leyes de Newton, la aplicación continua de fuerza resulta en una aceleración constante. Al no existir fricción con el medio, la velocidad del agente crecía linealmente a lo largo de la trayectoria.

Cuando el dron alcanzaba las proximidades de la caja, su energía cinética era tan elevada que el tiempo disponible para procesar cualquier cambio de trayectoria era inferior al tiempo de respuesta de los actuadores. El resultado sistemático era una colisión de alta velocidad que expulsaba la caja de su base, terminando el episodio abruptamente y generando un gradiente de recompensa negativo que paralizaba el aprendizaje.

B. El fracaso de las penalizaciones estáticas

El primer intento de corrección consistió en penalizar la magnitud absoluta de la velocidad lineal durante todo el episodio. Sin embargo, asignar un peso negativo constante a la velocidad indujo un mínimo local idéntico al observado en las primeras fases del proyecto: la red neuronal aprendió que cualquier movimiento restaba puntos, por lo que la política óptima descubierta fue no aplicar ninguna fuerza y quedarse flotando estáticamente en el punto de aparición.

C. La solución: penalización dinámica condicional y retropropulsión

Para resolver la dicotomía entre la inacción y la colisión a alta velocidad, se implementó en la versión definitiva del código una estructura de recompensa en dos niveles.

En primer lugar, se suavizó drásticamente el gradiente de aproximación. En lugar de recompensas exponenciales agresivas, se optó por un castigo lineal muy tenue por la distancia:

$$R_{dist} = -0.05 \cdot d_{approach}$$

Esta reducción restó urgencia a la maniobra, indicando al agente que llegar rápido no era la prioridad absoluta.

En segundo lugar, se programó una regla de penalización de velocidad estrictamente ligada a la posición relativa del dron respecto al objetivo, creando una "zona de frenado obligatorio". Como se refleja en la lógica del entorno, el sistema evalúa continuamente si la distancia al punto de aproximación es inferior a 1.0 metro. Si el dron entra en este radio y su velocidad supera un umbral de seguridad de $0.5\ \mathrm{m/s}$, se desencadena un castigo proporcional al exceso de velocidad:

$$\text{Si } d_{approach} < 1.0 \text{ y velocidad} > 0.5 \Rightarrow R = -2.5 \cdot (v - 0.5)$$

Esta formulación es clave para la viabilidad de la misión orbital. Le permite al dron acelerar libremente durante la primera fase del tránsito (cuando $d_{approach} > 1.0$), aprovechando la inercia para cruzar el espacio de trabajo sin ser castigado. Sin embargo, al cruzar la frontera de 1.0 metro, la red neuronal percibe una caída súbita en la recompensa esperada si no ha reducido su vector de velocidad.

Puesto que en el espacio no existe rozamiento para perder esa velocidad, la única forma que tiene el agente de evitar esta penalización es descubrir la maniobra de retropropulsión. A través de la optimización del gradiente, PPO aprendió a invertir el signo de las acciones de los propulsores justo antes de cruzar la barrera del metro de distancia, emitiendo fuerzas de empuje en sentido opuesto a su vector de movimiento. Esta configuración de la recompensa forzó el desarrollo de una política de control que replica las maniobras reales de aproximación de satélites, logrando que el cuadrotor llegue al punto pre-agarre con una velocidad residual prácticamente nula, preparado para iniciar la inserción precisa de los brazos robóticos.

5.3.2 Precisión de agarre en ingravidez

En el escenario de simulación terrestre, el éxito mecánico del agarre se veía facilitado por un condicionante físico implícito: la aceleración de la gravedad $(1g)$ mantenía la carga objetivo firmemente apoyada sobre su estructura base. La fuerza normal ejercida por el soporte generaba una fricción estática que proporcionaba un alto margen de tolerancia ante pequeños impactos o

desalineaciones durante la fase de aproximación de los brazos LiCAS A1. Si un gancho golpeaba ligeramente el lateral del asa, la caja absorbía la perturbación sin desplazarse.

En el entorno de microgravedad, la caja objetivo pasa a ser un cuerpo libre de base flotante con una masa inercial de 1.0 kg. Al anularse el peso, desaparece la fuerza normal y el anclaje por fricción. Cualquier error de precisión milimétrica que provoque una colisión prematura del efector final contra el pilar del asa transfiere cantidad de movimiento de forma instantánea al objeto. La caja, al no oponer resistencia vertical ni rozamiento, responde desplazándose a través del espacio vacío o adquiriendo un momento angular propio, lo que frustra sistemáticamente el episodio de simulación al alterar las coordenadas relativas antes de que el agente pueda cerrar la cadena cinemática bimanual.

Limitaciones de la heurística original: la "maniobra de cuchara" desarrollada en el Capítulo 4 forzaba al agente a descender por debajo de la cota del asa, alinear los brazos coplanarmente y aplicar un empuje vertical ascendente para lograr el enclavamiento. En 0g, el concepto geométrico de evitar la colisión frontal sigue siendo un imperativo operativo, pero la dinámica de ejecución pierde su anclaje inercial. Durante las primeras pruebas de entrenamiento desde cero en el entorno espacial, la réplica estricta de esta secuencia provocaba que el impulso traslacional comandado por el dron para "subir" e interceptar el asa terminara simplemente empujando la carga a lo largo del eje local Z, sin lograr la retención simultánea de ambos ganchos.

A. Desacoplamiento algorítmico: control del chasis vs. efectores finales

Para mitigar la extrema sensibilidad a las colisiones espurias, la estrategia de diseño en la versión definitiva del entorno requirió reestructurar el sistema de incentivos, trasladando el peso de la optimización desde la navegación global del chasis hacia la cinemática directa de los efectores finales.

En las iteraciones terrestres, el agente recibía su mayor flujo de recompensas al minimizar la distancia del centro de masa del dron a la zona de operación. En ingravidez, esta lógica propiciaba acercamientos peligrosos de todo el bloque del robot. La solución técnica implementada consistió en desacoplar ambas métricas posicionales. Como se observa en el código del entorno espacial, se premia explícitamente la reducción de la distancia media entre las puntas de los ganchos y el centro del asa, supeditándolo a un umbral de seguridad geométrica:

$$\text{Si } d_{approach} < 0.5 \Rightarrow R = 2.0 \cdot \left(1.0 - d_{hook\_avg}\right)$$

Esta formulación condicionada establece un cambio de fase en el comportamiento del optimizador. Obliga al agente a detener la traslación de la plataforma aérea a una distancia prudencial (aproximadamente medio metro) y a utilizar exclusivamente la extensión de las articulaciones rotacionales de los hombros y codos para cubrir los milímetros finales de la maniobra. Al restringir el movimiento únicamente a los eslabones distales de los brazos, se minimiza drásticamente la masa efectiva en movimiento y, en consecuencia, la energía cinética que podría transferirse a la caja en caso de un contacto subóptimo.

B. Condicionantes geométricos del pre-agarre

De forma simultánea, se integraron restricciones espaciales adicionales en la función Reward Shaping para garantizar que la configuración articular abordara el cilindro del asa de manera ortogonal y con la envolvente de apertura adecuada. La arquitectura monitoriza continuamente la separación euclídea entre el efector izquierdo y el derecho en el eje transversal. Si el agente opera dentro del radio de 0.5 metros manteniendo una separación estricta de entre 0.15 y 0.35 metros, recibe un refuerzo positivo

directo continuo de $+0.5$ puntos. Esta banda de tolerancia asimétrica exige matemáticamente que los brazos adopten una postura abierta que supera la cota longitudinal de diseño del asa (0.23 m), esquivando el choque contra los pilares extremos.

Asimismo, el paralelismo se evalúa calculando el vector unitario relativo entre las posiciones tridimensionales de ambos ganchos. La extracción del valor absoluto de la componente transversal de este vector proporciona la desviación angular respecto a la geometría de la caja. La recompensa proporcional obtenida a este valor induce una penalización implícita a cualquier configuración articular asimétrica o desviación de guiñada (yaw) en los codos. En 0g, este paralelismo es imprescindible, ya que una inserción sesgada deriva en la aplicación de un par de torsión indeseado sobre la carga en el instante mismo del contacto, lo que induciría a la caja a iniciar un giro libre irrecuperable.

C. El trigger inercial del agarre estable

Finalmente, la confirmación de la validez de la maniobra no se asume por simple proximidad espacial, sino que se subordina al motor de colisiones continuas de MuJoCo. La transición de estado interno solo ocurre cuando el sistema contabiliza un mínimo de 5 frames consecutivos de presión entre las mallas de colisión de los gancho y el material del asa. Es en este momento exacto cuando la función de recompensa habilita los premios de tracción de la carga útil, asegurando que cualquier orden posterior de los actuadores mueva al bloque en conjunto, evitando perturbaciones cinéticas asimétricas sobre la caja en flotación libre.

5.3.3 Estabilidad post-contacto: rediseño de las penalizaciones de esfuerzo de control

El instante exacto en el que los efectores finales establecen contacto sólido con el asa de la caja marca una frontera en la dinámica del sistema acoplado. En el entorno terrestre, la fase de levantamiento de la carga introducía una fuerza externa constante: el peso de la propia caja ($mg \approx 9.81$ N), el cual actuaba como un tensor natural para los brazos. Esta carga gravitacional inducía una rigidez mecánica pasiva en las articulaciones de los brazos LiCAS A1, forzando a los servomotores a operar bajo un régimen de par continuo que amortiguaba inherentemente cualquier ruido de alta frecuencia proveniente de las acciones del agente de RL.

Al transicionar al vacío orbital simulado, esta tensión desaparece por completo. La caja objetivo posee una masa inercial de 1.0 kg, pero carece de peso. En consecuencia, el desplazamiento del conjunto ensamblado dron-caja requiere únicamente vencer la inercia traslacional (ma). Esta carencia de pre-carga mecánica en los eslabones destapó una patología conductual severa durante las primeras etapas del entrenamiento en microgravedad: la inducción de oscilaciones constantes y vibraciones de alta amplitud en los efectores finales.

A. El fenómeno del Reward Hacking vibratorio

El origen de este comportamiento inestable se rastreó hasta una formulación heredada de la función de recompensa del entorno 1g. En la versión inicial de la tarea, una vez confirmado el agarre, se inyectaba una recompensa continua proporcional a la velocidad vertical del sistema para incentivar un levantamiento rápido. En las adaptaciones preliminares a 0g, esta línea de código se mantuvo intacta.

Sin la gravedad para anclar el movimiento, la red neuronal estocástica de PPO descubrió una laguna matemática. El agente aprendió que, comandando cambios de posición de extremo a extremo en los servos de los hombros y codos a altísima frecuencia, lograba que el centro de masa de la caja

experimentara picos instantáneos de velocidad vertical positiva. Aunque el desplazamiento neto del sistema era prácticamente nulo, la suma acumulada de las recompensas instantáneas por esos picos de velocidad maximizaba el retorno del episodio. Este temblor no solo era estéticamente inaceptable y físicamente irrealizable, sino que provocaba que la caja resbalara lateralmente y se escapara de los ganchos tras pocos frames de simulación.

La corrección de esta anomalía exigió razonar sobre los incentivos basados en la derivada de la posición. De esta manera, se decidió tras varias pruebas empíricas que lo correcto era eliminar por completo el premio a la velocidad vertical tras el contacto, con lo cual se consiguió eliminar estas vibraciones.

B. Reingeniería de las penalizaciones de estado

Para forzar una postura rígida y eliminar el ruido de las acciones post-contacto, se implementaron castigos dinámicos orientados a la estabilidad de la plataforma base. Dado que cualquier vibración asimétrica en los brazos transfiere momento angular al chasis principal en 0g, se introdujo una penalización directa sobre la magnitud de la velocidad angular del dron. Esta métrica se ponderó negativamente en la ecuación de recompensa (reward -= 0.25 * ang_speed), castigando cualquier maniobra articular brusca que indujera rotaciones en la base flotante. Asimismo, se penalizó la desviación de la actitud del cuadrotor respecto a la orientación ideal mediante el error del cuaternión.

C. Implementación de la restricción virtual

A pesar de la purga de incentivos vibratorios y la penalización del esfuerzo rotacional, la naturaleza de los contactos suaves de MuJoCo en ausencia de fuerzas normales continuas seguía siendo propensa a micro deslizamientos. Para garantizar la fidelidad de la maniobra de arrastre espacial una vez que el agente lograba la alineación perfecta, se desarrolló una solución computacional de bajo nivel: el muelle virtual.

Se definió un método que se invoca exclusivamente cuando la lógica de validación confirma un agarre bilateral estable. Este método calcula el error de posición entre el punto medio de ambos ganchos y el centro de masa de la caja. A continuación, aplica una ley de control Proporcional-Derivativo (PD) interna, con una rigidez extrema (kp=150.0) y un alto amortiguamiento (kd=30.0), para inyectar una fuerza correctora directa sobre la dinámica de la caja.

La fuerza calculada se introduce en el motor físico mediante el vector de perturbaciones externas. Esta técnica emula el bloqueo mecánico de una pinza robótica real. Evita que la estocasticidad de la red neuronal rompa la cadena cinemática cerrada, permitiendo que el entrenamiento se concentre en la traslación del conjunto masivo sin que fallos puramente numéricos de la resolución de contactos arruinen trayectorias exitosas.

D. Recompensa final por desplazamiento masivo

Finalmente, con la estabilidad garantizada por el control PD virtual y las penalizaciones angulares, el incentivo supremo de la misión se redefinió como una función estricta del desplazamiento relativo neto. Solo si el estado del sistema certifica un agarre seguro, el agente recibe una inyección estática de +20.0 puntos. Superada una cota de arrastre mínima de 0.05 m de altura, se desencadena la recompensa final masiva escalada por la distancia recorrida (reward += 100.0 + (altura_levantada * 300.0)). Este enfoque obliga al PPO a descubrir políticas de actuación suaves, constantes y unidireccionales para movilizar la carga útil a través del espacio, erradicando por completo el comportamiento oscilatorio.

5.4 Configuración del entrenamiento definitivo en 0g

Una vez superados los problemas dinámicos del vacío y tras haber rediseñado por completo la función de recompensa para penalizar la inercia descontrolada, el proyecto alcanza la última etapa de esta segunda fase: la configuración del entrenamiento definitivo. Tener un entorno simulado físicamente correcto en MuJoCo y una buena lógica de penalizaciones en Gymnasium es imprescindible, pero es solo una parte del trabajo. El siguiente paso es ajustar el algoritmo de inteligencia artificial para que sea capaz de asimilar toda esta información y aprender una política de control útil sin volverse inestable.

Al igual que en la primera fase del proyecto bajo gravedad terrestre, para este entrenamiento se ha continuado utilizando el algoritmo PPO implementado a través de la librería Stable-Baselines3. PPO ya demostró ser una herramienta muy robusta para el control continuo en 1g, pero el escenario de microgravedad plantea exigencias muy distintas a la red neuronal subyacente. En este nuevo entorno, el agente ya no controla un cuadrotor convencional que debe mantenerse en el aire compensando su propio peso, sino que gobierna un vehículo que opera bajo un paradigma de control espacial, desplazándose libremente por el vacío mediante la aplicación directa de fuerzas y pares.

Este cambio de comportamiento obliga a replantear cómo se comunica la red neuronal con el simulador. Por un lado, la información que el agente necesita "ver" para entender su entorno (las observaciones) debe modificarse para reflejar las nuevas prioridades de la tarea. Por otro lado, la forma en que el agente actúa sobre el mundo (las acciones) cambia radicalmente frente a la versión terrestre, obligando a reestructurar las salidas de la red neuronal. Además, la falta de rozamiento y gravedad hace que el sistema sea extremadamente sensible a la propia configuración interna del algoritmo PPO. Si se eligen unos parámetros de entrenamiento demasiado agresivos, el dron tomará decisiones bruscas y perderá el control rápidamente debido a la conservación del momento; por el contrario, si los parámetros son demasiado conservadores, el agente no explorará lo suficiente y el entrenamiento se estancará.

Por todos estos motivos, la configuración del aprendizaje en esta fase no podía ser una simple copia de la anterior, sino que ha requerido un ajuste metódico. En las siguientes subsecciones se detallará cómo se ha estructurado finalmente este proceso para asegurar la convergencia del modelo y el éxito de la misión. Primero, en la Sección 5.4.1, se explicarán los cambios estructurales aplicados a los espacios de acción y observación dentro de Gymnasium. Posteriormente, en la Sección 5.4.2, se presentarán los hiperparámetros definitivos del algoritmo PPO y las estrategias de normalización de datos que finalmente permitieron que el dron aprendiera a aproximarse y agarrar la caja de forma estable en condiciones de ingravidez.

5.4.1 Ajustes de la arquitectura del entorno

Para que el agente de RL pueda interactuar con el nuevo mundo sin gravedad, la interfaz de comunicación entre el algoritmo PPO y el simulador MuJoCo (definida a través de la librería Gymnasium) tuvo que reescribirse casi por completo. En el contexto del aprendizaje automático, y como ya se ha comentado anteriormente, esta interfaz se divide en dos bloques fundamentales: el espacio de acciones (lo que el dron puede hacer) y el espacio de observaciones (lo que el dron puede percibir). Adaptar estos espacios fue un paso obligado, ya que el multirrotor dejó de comportarse como una aeronave atmosférica para pasar a pilotarse como un vehículo espacial.

5.4.1.1 Redefinición del espacio de acciones

Durante la fase de gravedad terrestre (1g), el agente controlaba el dron de una forma indirecta: enviaba comandos a un controlador PID de vuelo (que regulaba el empuje de los cuatro motores para mantener la altitud y la inclinación) y comandos de posición a los servos de los brazos. Sin embargo, en el vacío orbital este enfoque perdía todo el sentido. Un dron en ingravidez no necesita motores apuntando siempre hacia abajo para no caerse; necesita poder empujar y girar en cualquier dirección del espacio en 3D, igual que hace un satélite con sus propulsores de gas.

Por este motivo, se eliminó el controlador de vuelo intermedio y se reestructuró el espacio de acciones para convertirlo en un vector continuo de 14 dimensiones, donde cada valor está normalizado entre -1 y 1 para facilitar el trabajo de la red neuronal. La distribución de estos comandos es la siguiente:

Fuerzas de traslación (acciones 0 a 2): los tres primeros valores dictan la fuerza directa que se aplica sobre el centro de masa del dron en los ejes X, Y y Z del mundo. El simulador escala estos valores normalizados multiplicándolos por una fuerza máxima permitida de 50 N, lo cual a posteriori ha llevado a pensar que es excesivo. Con esto, el agente aprende a aplicar impulsos lineales puros ("empujar" el dron hacia adelante o frenarlo) sin tener que inclinar el chasis. En MuJoCo, esto se implementó puenteando los actuadores tradicionales y escribiendo directamente sobre la variable xfrc_applied del cuerpo del dron.

Pares de rotación (Acciones 3 a 5): los tres siguientes valores controlan los torques de alabeo (roll), cabeceo (pitch) y guiñada (yaw), escalados a un máximo de 5 Nm. Esto permite al agente corregir su orientación o compensar los giros indeseados que provocan los brazos al moverse.

Comandos articulares (Acciones 6 a 13): las 8 dimensiones restantes se dedican a los brazos robóticos LiCAS A1. A diferencia del chasis, los brazos sí mantienen un control de posición (actuadores tipo servo). La red neuronal envía un valor normalizado que se multiplica por el límite físico de cada articulación (por ejemplo, entre -1.57 y 1.57 radianes para los hombros, y hasta 2.6 radianes para los codos), indicando a qué ángulo exacto debe moverse cada eslabón.

5.4.1.2 Ampliación del espacio de observaciones

Si el dron va a pilotarse de una manera más compleja, también necesita mejor información sensorial. En la versión espacial, el vector de observaciones se amplió hasta alcanzar las 42 dimensiones. El principio de diseño aquí fue proporcionar a la red neuronal únicamente los datos estrictamente necesarios para resolver la cinemática del problema, evitando saturarla con información redundante que pudiera ralentizar el aprendizaje.

El estado que recibe el agente en cada paso de simulación se compone de los siguientes bloques de información:

Estado cinemático del dron (13 dimensiones): incluye la posición tridimensional en el mundo (3), su orientación representada mediante cuaterniones (4), su velocidad lineal (3) y su velocidad angular (3). La inclusión precisa de las velocidades es vital en 0g, ya que, al no haber rozamiento, el agente necesita leer estos valores para saber cuándo debe aplicar la retropropulsión de frenado.

Estado propioceptivo de los brazos (16 dimensiones): se le pasa a la red la posición angular real (8) y la velocidad de giro (8) de cada uno de los servos. Esto permite al modelo saber exactamente en qué postura se encuentra el manipulador bimanual antes de intentar cerrarlo.

Vectores relativos de distancia (9 dimensiones): en lugar de darle al dron solo su posición global y la de la caja por separado (lo que obligaría a la red a aprender a restarlas), se le facilita el trabajo entregándole directamente el vector de distancia relativa entre el dron y el objetivo de aproximación (3). Además, y esto supuso una mejora notable para la maniobra final, se incluyeron los vectores relativos desde la punta del gancho izquierdo hasta el lado izquierdo del asa (3) y desde el gancho derecho hasta el lado derecho (3). Esto actúa como una "mira telescópica" matemática para el agarre de precisión.

Métricas de estado de contacto (4 dimensiones): este último bloque fue la pieza clave para la retención en microgravedad. Puesto que en el espacio la caja flota libremente si la tocas mal, el agente necesita estar absolutamente seguro de que la ha atrapado. Se introdujeron dos contadores que miden cuánto tiempo llevan los ganchos presionando el asa, un valor que indica la separación normalizada entre los ganchos (para saber si los brazos están lo suficientemente cerrados), y una variable booleana que se activa a 1.0 únicamente cuando el simulador confirma que el agarre es estable y seguro.

En resumen, esta nueva arquitectura de entorno convierte al sistema en un problema de control directo de fuerzas, dándole a la inteligencia artificial un lienzo sensorial mucho más rico enfocado en la velocidad y la distancia relativa. Gracias a esto, el algoritmo PPO dispone de los canales de entrada y salida adecuados para enfrentarse a las peculiaridades dinámicas que exige la captura de un objeto flotante.

5.4.2 Hiperparámetros

Configurar el algoritmo PPO para que resuelva una tarea en microgravedad supone un reto matemático considerable. Al no existir rozamiento ni un vector de gravedad que amortigüe los errores, cualquier acción subóptima por parte del dron tiene consecuencias permanentes en el episodio. Si la red neuronal se actualiza de forma demasiado agresiva durante el entrenamiento, el agente puede olvidar rápidamente cómo frenar o cómo estabilizar los brazos, arruinando horas de cómputo. Por tanto, fue indispensable establecer un conjunto de hiperparámetros y técnicas de preprocesamiento de datos que garantizaran un aprendizaje estable, lento pero seguro.

A continuación, se detalla la configuración definitiva que permitió a la red neuronal converger en el entorno espacial, dividida en dos pilares fundamentales: el tratamiento de los datos (normalización) y el ajuste del optimizador.

5.4.2.1 Estrategias de normalización de datos

Las redes neuronales artificiales, como la que utiliza PPO en su base, son muy sensibles a la escala de los números que reciben como entrada. En nuestro entorno de 0g, las observaciones mezclan magnitudes físicas muy diferentes: tenemos posiciones que pueden variar varios metros, velocidades en metros por segundo, cuaterniones que siempre están entre -1 y 1, y ángulos de los servomotores en radianes. Si introducimos estos datos crudos en la red, los pesos sinápticos se descompensarían intentando darle sentido a variables tan dispares.

Para solucionar esto, se implementó la herramienta VecNormalize proporcionada por Stable-Baselines3. Esta envoltura (wrapper) actúa de forma transparente sobre el entorno y realiza dos tareas críticas:

Normalización de las observaciones: VecNormalize calcula una media móvil y una varianza en tiempo real de todo lo que "ve" el dron. Antes de pasar el vector de 42 dimensiones a la red neuronal, le resta la media y lo divide por la desviación estándar. De esta forma, sin importar si el dron está a 3 metros o a 10 centímetros de la caja, los valores numéricos que procesa la red se mantienen siempre acotados (generalmente en un rango entre -5 y 5). Esto evita gradientes explosivos durante la fase de retropropagación (backpropagation).

Normalización de las recompensas: las recompensas que programamos en la sección 5.3 pueden tener picos altos (como los +8000 puntos por éxito final) o pequeñas sumas decimales por acercamiento. Normalizar estos retornos es vital para que la red del Crítico (la parte de PPO que estima el valor de un estado) no se sature y pueda predecir correctamente si un movimiento es bueno a largo plazo.

Por el lado de las acciones, la normalización se resolvió directamente desde la arquitectura del código. Como vimos, la red neuronal solo emite comandos en el rango de -1,1. Es dentro del propio paso del simulador (step) donde este valor normalizado se multiplica por los límites físicos reales (los 50 N de empuje máximo o los radianes de los brazos).

5.4.2.2 Ajuste de hiperparámetros del algoritmo PPO

Una vez asegurado que los datos entran y salen limpios de la red, el siguiente paso fue definir cómo debía aprender el agente a partir de su experiencia. Se mantuvo la estructura de red neuronal Perceptrón Multicapa (MLP) con dos capas ocultas de 256 neuronas cada una (net_arch = [256, 256]), ya que demostró tener la capacidad de abstracción suficiente en la fase terrestre sin penalizar en exceso el tiempo de cómputo.

Sin embargo, los parámetros de optimización se ajustaron cuidadosamente para lidiar con el vacío. Los valores definitivos más relevantes fueron los siguientes:

Tasa de aprendizaje (Learning Rate, α): 3.5×10-4 constante.

Aunque en algunos problemas complejos se utilizan tasas decrecientes, aquí se optó por mantener el estándar de Adam. Reducirla demasiado atascaba al agente en el punto de aparición por miedo a moverse y generar penalizaciones, mientras que aumentarla provocaba que la política de control de vuelo se desestabilizara al intentar aprender a mover los brazos.

Tamaño del lote y actualizaciones (n_steps, batch_size, n_epochs):

Se configuró n_steps = 4096, lo que significa que el dron debe interactuar con el entorno durante 4096 pasos de simulación antes de que PPO se detenga a calcular una actualización de la red. Teniendo en cuenta que los episodios máximos duran 1500 pasos, esto permite al algoritmo recopilar entre dos y tres trayectorias completas (o docenas de fallos rápidos) por cada ciclo. Esta memoria temporal se divide en mini-lotes (batch_size = 256) y se repasa 7 veces (n_epochs = 7). Este volumen de datos masivo asegura que la red no memorice un único impacto fortuito, sino que aprenda tendencias estadísticas reales sobre cómo funciona la inercia.

Factor de descuento (Gamma, γ): 0.99.

Este parámetro define cuánta importancia le da el agente a las recompensas futuras frente a las inmediatas. En 0g, un empuje excesivo al inicio del episodio arruina el agarre 10 segundos después. Un factor de 0.99 asegura que el dron tenga un "horizonte temporal" largo y sea capaz de entender que frenar ahora (aunque cueste energía) le permitirá ganar los puntos del agarre en el futuro.

Coeficiente de Entropía (ent_coef): 0.01.

La entropía controla cuánto de aleatorias son las acciones del dron. Si es muy alta, el agente prueba cosas nuevas constantemente; si es baja, se vuelve determinista y explota lo que ya sabe. Se ajustó a un valor ni muy alto ni muy bajo (0.01) porque, en microgravedad, añadir demasiado "ruido" a los comandos se traduce directamente en vibraciones de los brazos y giros descontrolados del chasis (como se explicó en la Sección 5.3.3). Queremos que explore, pero de forma suave.

Restricciones de estabilidad (clip_range = 0.2 y target_kl = 0.05):

Estas son las redes de seguridad de PPO. El clip_range impide que la política cambie más de un 20% en una sola actualización. El target_kl aborta el entrenamiento de esa época si detecta que la nueva red neuronal difiere estadísticamente demasiado de la anterior. En la ingravidez, donde encontrar la maniobra de "cuchara" es tan difícil, estas protecciones son vitales para evitar que una mala racha de colisiones borre el progreso acumulado de millones de pasos anteriores.

Con este marco de trabajo configurado, el entorno de aprendizaje quedó listo para comenzar la recopilación de millones de pasos de simulación, dando paso a la obtención de los resultados que se discutirán en el siguiente capítulo de la memoria.

A modo de síntesis y para facilitar la replicabilidad del experimento, la siguiente tabla condensa la configuración paramétrica definitiva del algoritmo PPO implementada para el entorno espacial, cerrando así la definición metodológica de esta fase.

Tabla 3: Hiperparámetros entrenamiento 0g

Hiperparámetro	**Valor**	**Descripción Breve**
learning_rate (α)	3×10^{-4}	Tasa de aprendizaje constante para el optimizador (Adam).
n_steps	4096	Pasos de simulación recolectados antes de cada actualización de la red.
batch_size	512	Tamaño del subconjunto de datos (mini-lote) para el cálculo del gradiente.
n_epochs	8	Número de pasadas de optimización sobre el mismo conjunto de datos recolectado.
gamma (γ)	0.99	Factor de descuento; da prioridad a la consecución del agarre a largo plazo.
ent_coef	0.01	Coeficiente de entropía; mantiene una exploración estocástica controlada.
clip_range (ϵ)	0.2	Recorte de la función objetivo; evita actualizaciones que destruyan lo ya aprendido.
target_kl	0.05	Límite de divergencia admisible para abortar de forma segura una

Hiperparámetro	Valor	Descripción Breve
		época inestable.
net_arch	[256, 256]	Topología de la red neuronal (dos capas ocultas de 256 neuronas para Actor y Crítico).

Con esta arquitectura de red parametrizada, los datos de observación correctamente normalizados y el simulador ajustado a las leyes inerciales, el entorno quedó listo para la ejecución computacional. Los resultados empíricos de este entrenamiento, así como la evaluación cualitativa y cuantitativa de las trayectorias de agarre en flotación libre, se analizan en profundidad en el siguiente capítulo de la memoria.

6 RESULTADOS Y DISCUSIÓN

El presente capítulo constituye la validación empírica de todo el trabajo de modelado dinámico, diseño de la arquitectura de control y configuración del algoritmo de aprendizaje por refuerzo detallado en los capítulos anteriores. Una vez que el optimizador PPO ha completado los millones de pasos de simulación estipulados para cada uno de los entornos (terrestre y espacial), es el momento de extraer la telemetría de MuJoCo y analizar cómo se comporta realmente el sistema acoplado cuadrotor-LiCAS A1 al enfrentarse a la tarea de manipulación aérea.

El objetivo de este análisis no se limita a certificar si el dron consigue, de forma binaria, agarrar la caja o no. Se busca profundizar en la cinemática de la maniobra, la suavidad del vuelo y el esfuerzo mecánico exigido a la plataforma. Para ello, se evaluarán los datos registrados durante los episodios de prueba mediante una batería de métricas que incluyen trayectorias 3D, evolución de la distancia al objetivo, perfiles de velocidad lineal, estabilidad de la actitud (medida a través de los ángulos de Euler) y la actividad de los actuadores.

Para abordar esta información de forma estructurada, el capítulo se divide en tres grandes bloques. En primer lugar, se analizará el desempeño del agente en el entorno bajo gravedad terrestre (1g), lo que servirá para establecer una línea base de comportamiento funcional. En segundo lugar, se evaluará el rendimiento en el entorno de microgravedad (0g), demostrando cómo el agente ha aprendido a lidiar con la conservación del momento y la inercia pura. Finalmente, se planteará una discusión comparativa que contrastará el desempeño en ambos regímenes físicos, poniendo en perspectiva la complejidad del proyecto y justificando las soluciones adoptadas frente a los fallos encontrados durante las fases iterativas de desarrollo.

6.1 Evaluación del modelo en gravedad terrestre (1g)

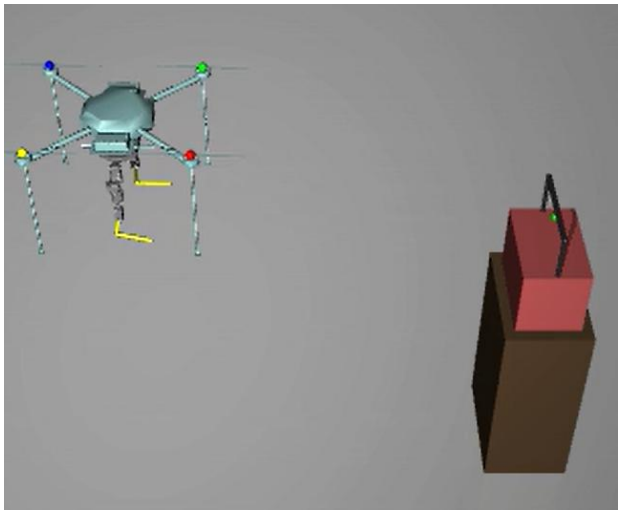
La validación operativa del sistema comienza con su evaluación en el entorno de gravedad estándar (g=9.81 m/s2). Este escenario actúa como el caso de control fundacional del proyecto. En estas condiciones, el reto principal para la red neuronal es la gestión simultánea de la sustentación y la navegación; el cuadrotor está obligado a comandar un empuje continuo y preciso en sus rotores para contrarrestar su propio peso, al mismo tiempo que debe inclinar su chasis para desplazarse hacia la carga útil.

Lograr que el algoritmo converja en una política de vuelo estable no es trivial, ya que el movimiento de los brazos manipuladores al preparar la postura de agarre desplaza el centro de masa del conjunto, introduciendo perturbaciones asimétricas que el controlador de actitud debe absorber en tiempo real. Los resultados que se exponen a continuación corresponden a la evaluación del mejor modelo de la fase terrestre, ejecutando trayectorias desde posiciones de aparición aleatorias para confirmar su capacidad de generalización.

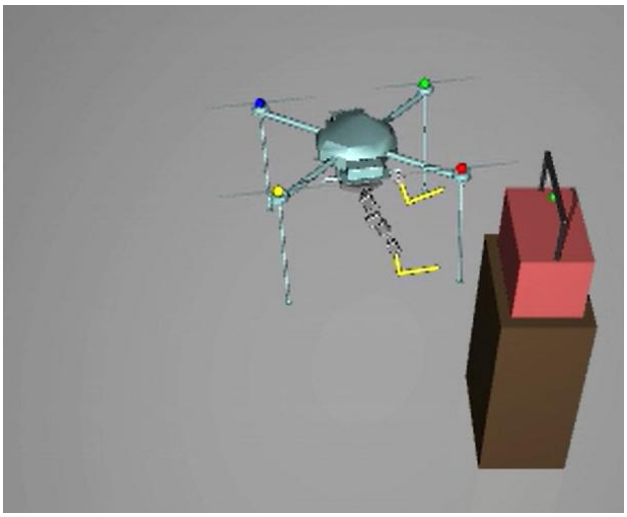
Para ofrecer una radiografía técnica completa del comportamiento del dron, el análisis de esta fase se ha subdividido en tres áreas clave. En primer lugar, se diseccionará la geometría de la aproximación mediante el análisis de la trayectoria tridimensional y el perfil de velocidades. A continuación, se examinará la robustez mecánica de la maniobra revisando la estabilidad de los ángulos de Euler y el esfuerzo demandado a los actuadores, verificando que no existan oscilaciones o saturaciones

indeseadas. Finalmente, se agruparán las métricas numéricas globales para cuantificar de forma objetiva la consistencia del aprendizaje y la tasa de éxito final del agarre.

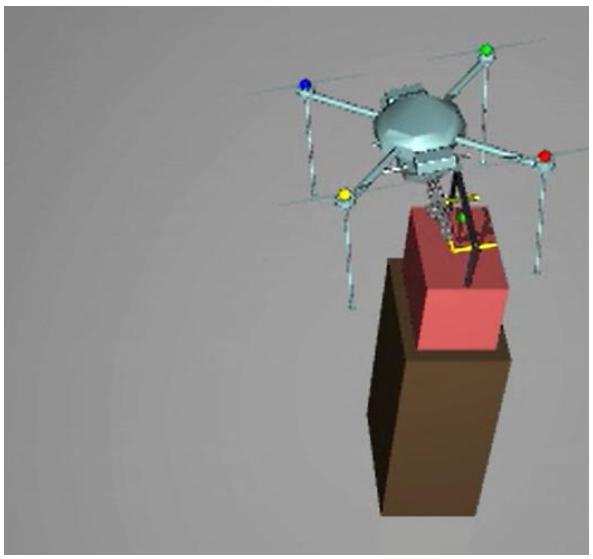
A continuación, se muestran los frames clave del video renderizado del modelo en gravedad terrestre:



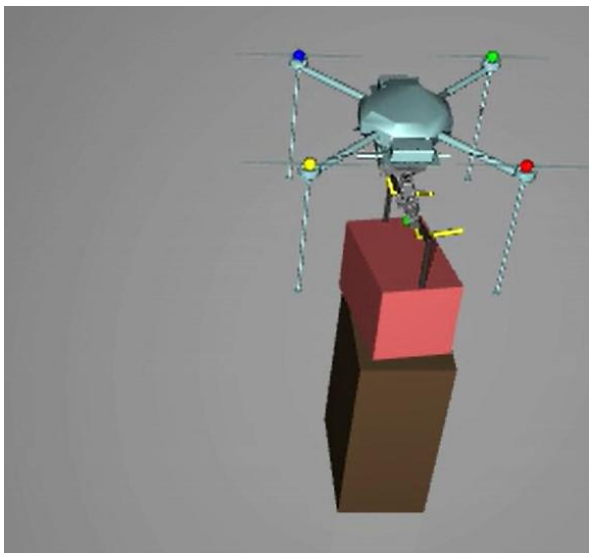
Frame 1: Posición inicial



Frame 2: aproximación, el dron empieza a frenar



Frame 3: dron parado, comienza el agarre



Frame 4: consecución del agarre

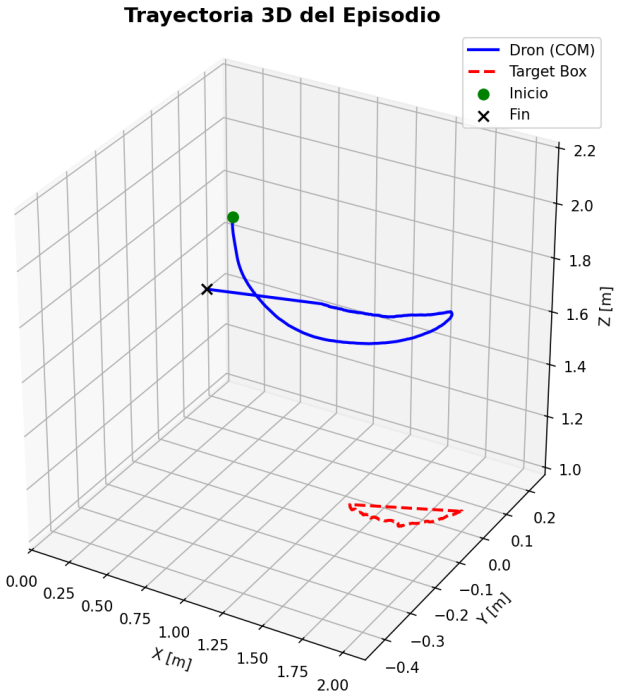
6.1.1 Análisis de trayectoria y cinemática de aproximación

El primer paso para validar la política de control adquirida por el agente de RL es evaluar cómo resuelve el problema de la navegación espacial desde su punto de aparición hasta la zona de manipulación. En un entorno sometido a la gravedad terrestre, esta maniobra exige que el cuadrotor rompa su estado de vuelo estacionario, incline su vector de empuje para generar avance lateral y, simultáneamente, gestione su altitud para no impactar contra el suelo ni contra la estructura que soporta la carga.

Para ilustrar este comportamiento, se ha extraído la telemetría completa de un episodio de evaluación

exitoso, permitiendo diseccionar la maniobra desde una perspectiva geométrica y cinemática.

6.1.1.1 Geometría de la aproximación: trayectoria 3D

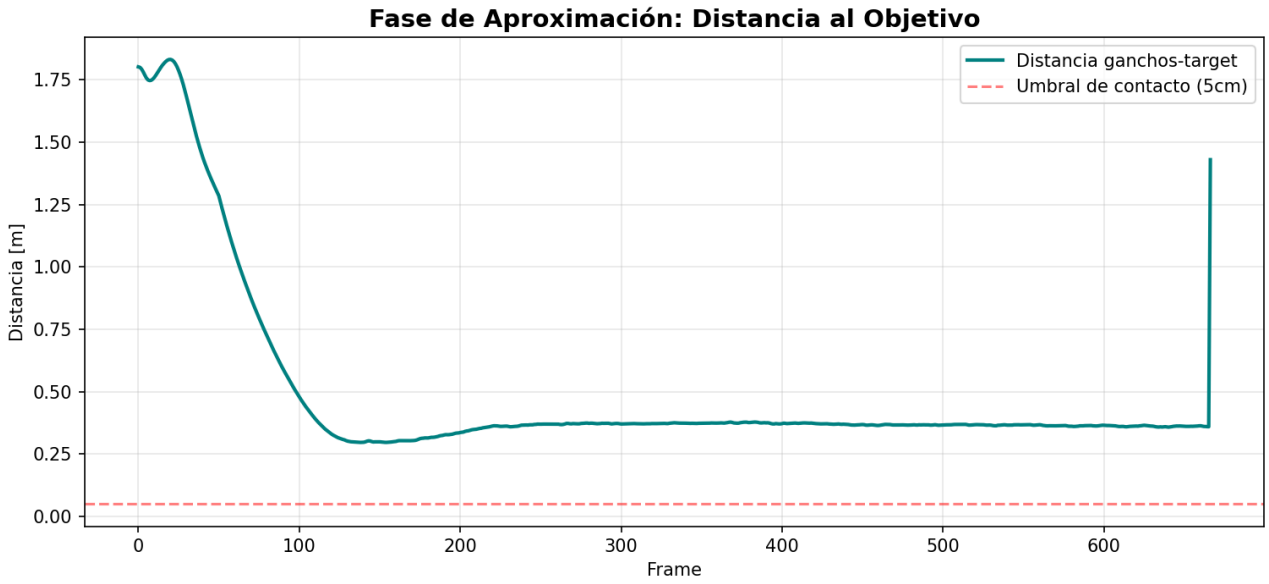


Resultado 1g 1: Trayectoria 3D

El registro tridimensional del centro de masa (CoM) del dron durante el episodio revela una estrategia de vuelo notablemente fluida y conservadora. Como se observa en la gráfica de la trayectoria 3D, el vehículo inicia su movimiento desde una cota elevada (marcador verde, aproximadamente a 1.9 m de altura). En lugar de trazar una línea recta descendente y agresiva hacia la caja objetivo (marcada en rojo discontinuo), el agente ejecuta una curva de descenso suave.

Esta trayectoria curva demuestra empíricamente el éxito de las modificaciones implementadas en la función de recompensa descritas en capítulos anteriores. El comportamiento de la aproximación impulsiva, que en iteraciones previas llevaba al dron a estrellarse frontalmente al intentar minimizar la distancia por la vía más rápida, ha sido completamente erradicado. El dron desciende perdiendo altitud de forma gradual mientras avanza en el plano horizontal (ejes X e Y), describiendo una parábola controlada que le permite acercarse a la zona de operación con un margen de seguridad estructural holgado, evitando así cualquier intersección indeseada con la caja base.

6.1.1.2 Evolución temporal de la distancia

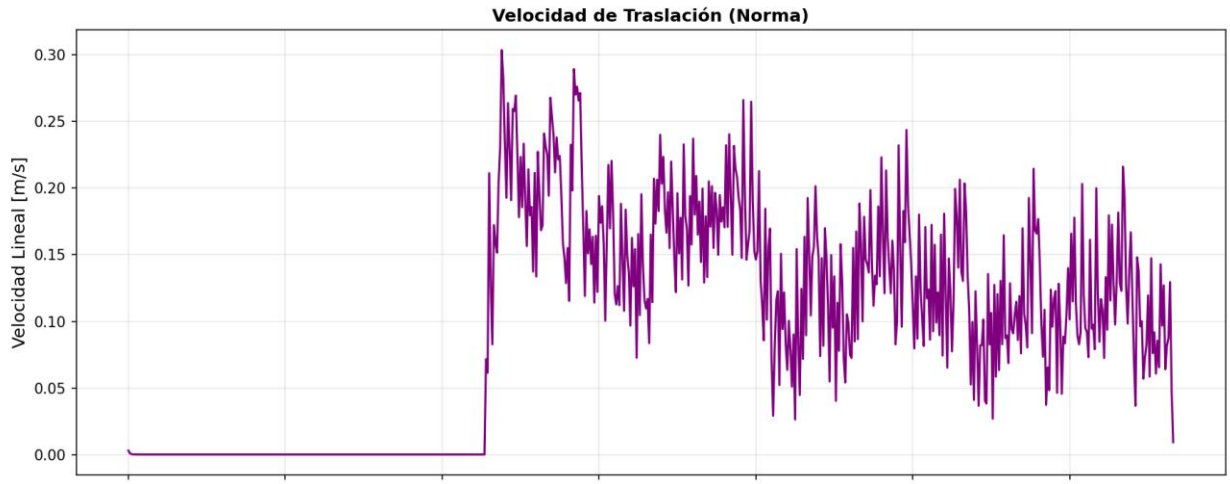


Resultado 1g 2: Distancia al objetivo

La gráfica que representa la distancia euclídea entre los ganchos y el objetivo a lo largo de los frames de simulación confirma esta estrategia de aproximación en dos fases. Al inicio del episodio, la distancia cae de forma pronunciada y constante desde los 1.8 m hasta alcanzar la barrera de los 0.35 m alrededor del frame 150. Esta caída inicial representa la fase de vuelo de crucero.

Lo verdaderamente destacable ocurre a partir de ese instante: la curva se aplana por completo, manteniendo una distancia casi constante de unos 35 cm durante el resto del episodio, rozando el umbral de contacto pero sin cruzarlo de forma abrupta. Este aplanamiento o "meseta" en la gráfica es la representación matemática del punto de preparación virtual (approach target) que se configuró en el entorno. El agente ha aprendido a detener su traslación global y estabilizarse en vuelo estacionario justo frente a la caja. Desde esta posición segura, el robot puede realizar la delicada "maniobra de cuchara" utilizando principalmente los grados de libertad de sus brazos robóticos, en lugar de arriesgarse a embestir la carga con el chasis principal.

6.1.1.3 Perfil cinemático: velocidad lineal

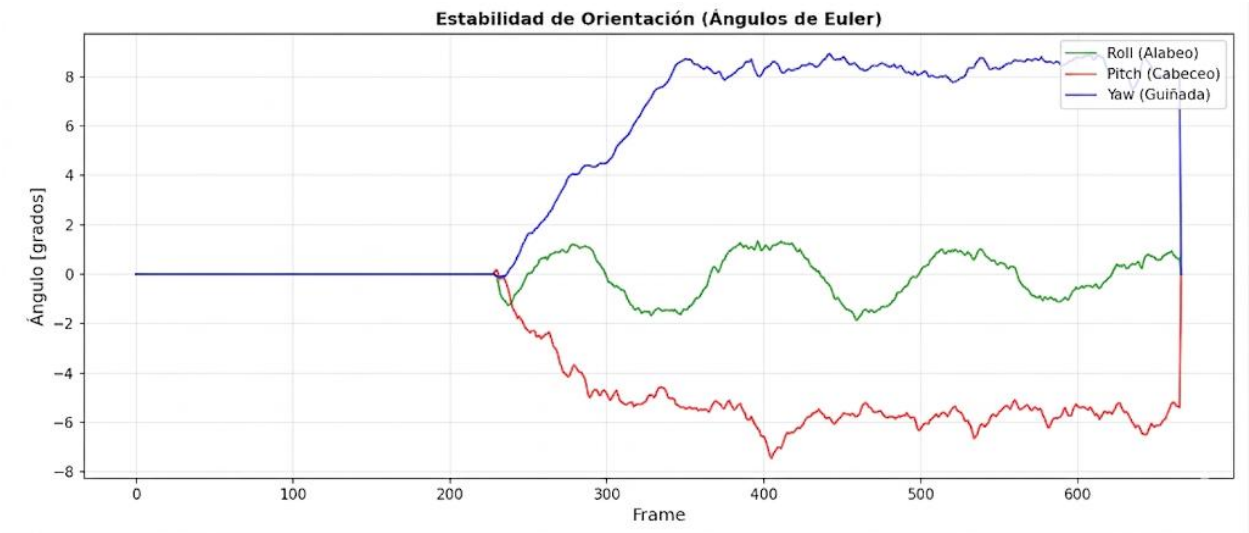


Resultado 1g 3: Velocidad lineal

El análisis de la velocidad de traslación (norma del vector velocidad lineal) aporta la justificación física a la suavidad de la trayectoria. La gráfica superior de estabilidad refleja un inicio estático (velocidad nula), correspondiente a los frames de inicialización donde el simulador asienta el entorno aplicando únicamente el empuje de hover para estabilizar el sistema antes de ceder el control al agente.

Una vez que el agente toma el control (alrededor del frame 230 en esta métrica), la velocidad se incrementa, pero lo hace bajo límites extremadamente conservadores. La magnitud de la velocidad oscila entre 0.05 m/s y un pico máximo de apenas 0.30 m/s. Cabe recordar que el controlador de bajo nivel permitía velocidades de hasta 1.5 m/s. El hecho de que la red neuronal haya optado por operar al 20% de su capacidad máxima demuestra la eficacia de la penalización dinámica de velocidad implementada por zonas. El dron prefiere avanzar lentamente y asegurar la estabilidad de sus brazos en lugar de acumular energía cinética que luego no pueda disipar a tiempo. Las oscilaciones de alta frecuencia visibles en el perfil de velocidad son características del lazo de control PID en cascada realizando micro-correcciones continuas para mantener el vector de avance frente a las perturbaciones del simulador.

6.1.2 Estabilidad de actitud: ángulos de Euler



Resultado 1g 4: Ángulos de Euler

Finalmente, la gráfica de la estabilidad de orientación (ángulos de Euler) certifica la robustez de la plataforma aérea durante toda esta fase de aproximación. En la dinámica de los multirrotores, cualquier desplazamiento horizontal requiere obligatoriamente inclinar el chasis (generar un ángulo de alabeo o cabeceo).

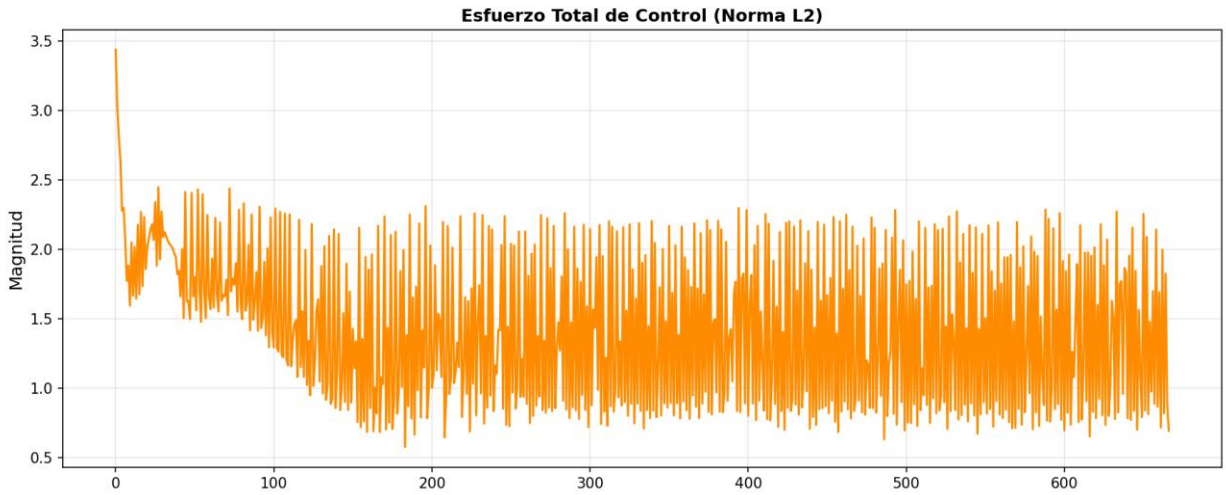
Los datos registrados muestran una estabilidad excepcional: el ángulo de cabeceo (Pitch, en verde) se mantiene prácticamente neutro, oscilando en un rango de ±2 grados; el alabeo (Roll, en rojo) alcanza un máximo de tan solo -8 grados para generar el desplazamiento lateral; y la guiñada (Yaw, en azul) se ajusta de forma suave hasta los 8 grados para alinear los ganchos perpendicularmente con el asa.

Mantener los ángulos de Euler en magnitudes tan bajas (siempre por debajo de los 10 grados de inclinación) es un logro fundamental de la arquitectura de control jerárquica. Si el dron tuviera que inclinarse 20 o 30 grados para avanzar, la inercia de los pesados brazos robóticos LiCAS A1 situados en la parte inferior generaría un efecto de péndulo masivo, desplazando drásticamente el centro de masa y haciendo casi imposible la estabilización. Al operar con inclinaciones mínimas, la red neuronal garantiza que la base flotante se mantenga lo más plana y horizontal posible, ofreciendo un marco de referencia estable para que la cinemática de los efectores finales pueda culminar el agarre de precisión sin verse contaminada por el balanceo del chasis.

6.1.3 Esfuerzo de control y actividad por actuador

Demostrar que el dron sigue una trayectoria suave y adecuada hacia la caja objetivo es solo una parte de la validación. En el ámbito del control robótico, y muy especialmente cuando se utilizan algoritmos de RL, es crucial analizar cómo se genera exactamente ese movimiento. Una política de control puede parecer exitosa visualmente o en su trayectoria 3D, pero si lo logra a base de enviar comandos erráticos, saturar los motores o hacer vibrar los servos a altas frecuencias, el modelo resulta inútil para su implementación en hardware físico real. Por ello, en este apartado se disecciona la "salida" de la red neuronal: el esfuerzo de control y la actividad individual de cada actuador a lo largo del episodio.

6.1.3.1 Análisis del esfuerzo total de control



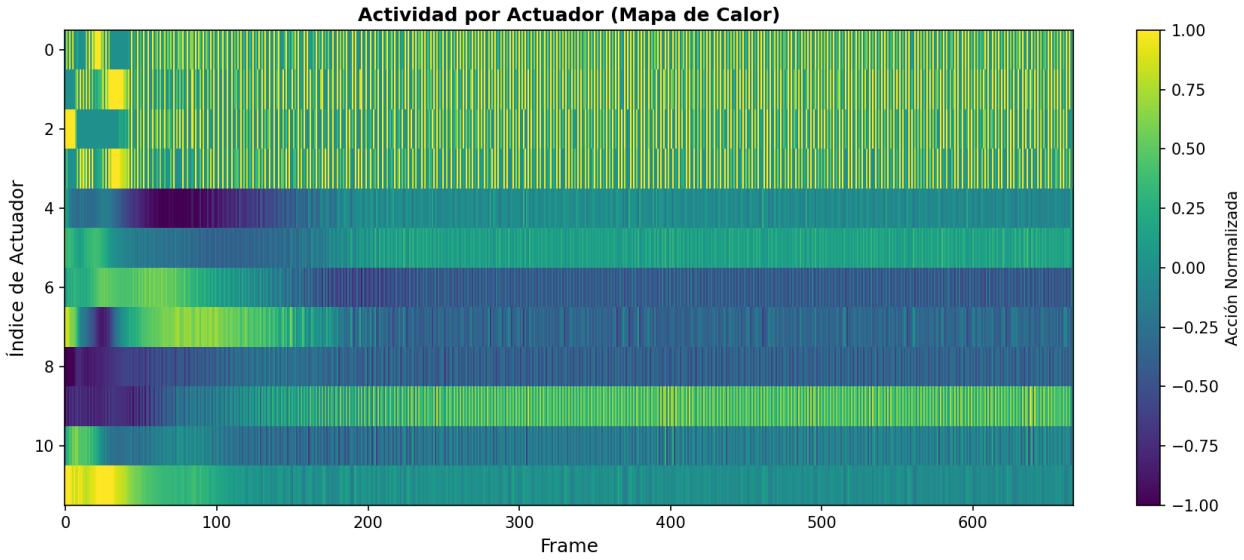
Resultado 1g 5: Esfuerzo total de control

El esfuerzo de control es una métrica que cuantifica cuánta magnitud de acción está demandando el agente al sistema en cada instante. Para evaluar este aspecto de forma agnóstica a la escala física de cada motor, se extrajo la norma euclídea (norma L2) del vector de acciones normalizado que emite la red neuronal. Puesto que el espacio de acción consta de 12 dimensiones acotadas individualmente en el rango de $[-1,1]$, el límite máximo teórico que el agente podría comandar si saturara todos los actuadores simultáneamente sería de $\sqrt{12} \approx 3.464$.

Consultando los estadísticos globales del episodio, se observa que el esfuerzo de control medio se mantiene en un valor de 1.4456. Este valor medio, situado por debajo del límite de saturación de 3.46, es un buen indicativo.

Observando la gráfica del esfuerzo total de control durante los 667 frames que dura el episodio (equivalentes a unos 11.11 segundos de simulación), se aprecia un comportamiento estable que confirma el éxito directo de la penalización de suavidad que se implementó en la función de recompensa. Al castigar la derivada discreta de la acción comandada, el algoritmo PPO se vio obligado a aprender que las transiciones entre estados deben realizarse modulando las señales poco a poco, en lugar de dar fuertes impulsos a los mandos del dron.

6.1.3.2 Desglose de la actividad por actuador



Resultado 1g 6: Actividad por actuador

Para entender mejor de dónde proviene este esfuerzo, la gráfica de actividad por actuador divide las señales en dos grupos lógicos. Es vital aclarar que los índices del 0 al 3 no corresponden a la señal PWM directa de los cuatro motores, sino a las referencias cinemáticas de vuelo (velocidades deseadas y tasa de guiñada) que el agente envía al controlador PID en cascada. Lo que visualmente podría interpretarse como una saturación en estos primeros canales (valores constantes en $+1$ o -1), representa en realidad al agente demandando la velocidad de traslación máxima permitida en su espacio de acción para acercarse al objetivo. Es el bucle PID subyacente el encargado de traducir estas referencias agresivas en comandos de empuje reales, manteniendo el esfuerzo promedio de los rotores en el margen dinámico seguro del 40-50%. Por otro lado, los índices del 4 al 11 corresponden a los comandos directos de posición articular enviados a los brazos robóticos.

Por la parte de los manipuladores LiCAS A1 (las 8 señales enviadas a los servomotores), el comportamiento es aún más revelador. La decisión arquitectónica de utilizar actuadores de posición en lugar de control de torque directo ha dado sus frutos en esta gráfica. En lugar de tener que enviar señales oscilantes para luchar contra la gravedad y mantener los brazos en el aire, el agente simplemente envía la referencia angular deseada. Se observa cómo, en los primeros frames, los comandos de los brazos se ajustan de forma coordinada para abrir los ganchos y preparar la anchura necesaria (buscando esa separación óptima para esquivar los pilares del asa que se forzaba en la función de recompensa). Una vez alcanzada esta "postura de pre-agarre", las señales de los 8 servos se vuelven prácticamente planas, comportándose como una estructura rígida.

Al observar las señales de los 8 servos en la gráfica térmica, se hace evidente una de las características más singulares de las políticas descubiertas mediante Aprendizaje por Refuerzo: la ausencia de una simetría matemática perfecta. Desde un enfoque de diseño analítico, se esperaría que los comandos del brazo izquierdo fueran el opuesto exacto de los del derecho (reflejado en colores invertidos en el mapa de calor). Sin embargo, al no existir un término de penalización explícito por asimetría en la función de recompensa, el agente ha convergido hacia una postura ligeramente idiosincrásica. Esta asimetría funcional demuestra que la red neuronal no busca la pose más estética, sino aquella que resuelve la cinemática del agarre y absorbe las tolerancias físicas del chasis de la manera más robusta posible.

6.1.3.3 Métricas importantes sobre el episodio

Tabla 4: Métricas resultado 1g

Categoría	Métrica	Valor	Relación con tu análisis
Análisis temporal (evolución)	Reward promedio (1ª mitad)	9.24	Puntos por acercarse y mantener vuelo estable inicial.
	Reward promedio (2ª mitad)	72.35	Puntos al mantener la postura geométrica perfecta ("maniobra de cuchara").
	Mejora neta entre mitades	+63.11	Demuestra que la red neuronal sabe maximizar el retorno en el momento adecuado.
Picos y promedios de Recompensa	Reward máximo instantáneo	4140.52	Alcanzado en el momento cumbre sin comprometer la integridad mecánica.
	Reward promedio por frame	40.84	Rendimiento medio global a lo largo de todo el episodio.
	Reward mínimo	-0.26	Peor penalización instantánea registrada.
	Reward total	27242.88	Recompensa bruta acumulada durante todo el vuelo.

Categoría	Métrica	Valor	Relación con tu análisis
Esfuerzo y Control Físico	Esfuerzo Control Medio	1.4456	Certifica la suavidad y viabilidad energética (no es un "hack" matemático).
	Velocidad Máxima	0.30 m/s	Confirma una aproximación controlada y suave.
	Frames en Contacto (<5cm)	0	Ausencia de colisiones no deseadas a corta distancia.
Datos Generales del Vuelo	Frames totales	667	Duración del episodio en pasos de simulación.
	Duración temporal	11.11 s	Tiempo total en segundos.
	Distancia Mínima	0.29 m	Máximo acercamiento logrado al objetivo.
	Distancia Final	1.42 m	Separación al concluir el episodio.

La estabilización de todos estos actuadores tiene un reflejo directo en la calidad de la política aprendida, algo que se evidencia al analizar la evolución temporal de los premios. Según los registros del modelo, durante la primera mitad del episodio, el agente acumula una recompensa promedio de 9.24 por frame. Este valor se corresponde con los puntos obtenidos simplemente por acercarse y mantener el vuelo estable.

No obstante, en la segunda mitad del episodio, cuando la actividad de los actuadores se estabiliza y el dron mantiene la postura geométrica perfecta bajo el asa, la recompensa promedio se dispara hasta los 72.35 por frame. Esta mejora neta de +63.11 puntos entre ambas mitades del vuelo demuestra que la red neuronal sabe perfectamente cuándo debe actuar y cuándo debe mantener las posiciones, logrando maximizar el retorno (alcanzando recompensas máximas instantáneas de hasta 4140.52 en el momento cumbre) sin comprometer la integridad mecánica del sistema.

En definitiva, el análisis del esfuerzo de control certifica que la política descubierta en gravedad terrestre no es un "hack" matemático del simulador, sino una estrategia de vuelo y manipulación físicamente viable, energéticamente eficiente y lo suficientemente suave como para ser ensayada en una plataforma real en el futuro.

6.1.4 Evaluación cuantitativa: desempeño global y análisis cuantitativo

Una vez validada la suavidad de la trayectoria y el esfuerzo mecánico de los actuadores, el análisis de la fase bajo gravedad terrestre (1g) concluye con la evaluación puramente cuantitativa del episodio. En el campo del RL la observación cualitativa de una maniobra debe estar siempre respaldada por las métricas internas del optimizador y del entorno, ya que son estos números los que confirman si la política ha convergido de manera robusta o si, por el contrario, el agente está teniendo "suerte" en una semilla de inicialización concreta.

Para ello, se han recopilado y procesado las estadísticas globales del modelo durante el mismo episodio de evaluación, discutidas y mostradas en formato tabla en la sección anterior.

6.1.4.1 Eficiencia temporal y márgenes cinemáticos

La telemetría general revela que el dron logró completar la secuencia de aproximación y posicionamiento en un total de 667 frames de simulación. Considerando la configuración temporal del motor físico, esto se traduce en una duración efectiva de 11.11 segundos de vuelo continuo. Este lapso temporal es un indicador muy positivo desde la perspectiva de la viabilidad de la misión: demuestra que el agente no exhibe el comportamiento estancado o "miedoso" de las primeras etapas del entrenamiento, pero tampoco realiza una aproximación temeraria.

Este equilibrio cinemático se confirma al revisar la velocidad lineal máxima registrada durante todo el trayecto, la cual se sitúa en 0.3038 m/s. Como se argumentó en el diseño de la función de recompensa, limitar la velocidad es imperativo para evitar que la energía cinética del sistema acoplado (dron más brazos) se vuelva inmanejable. Operar a un tercio de metro por segundo garantiza que cualquier posible impacto sea elástico y no comprometa la integridad de la carga o los efectores finales.

6.1.4.2 Precisión espacial de la maniobra

Desde el punto de vista geométrico, la estadística de distancia corrobora la estrategia de "cuchara" analizada previamente. La métrica de distancia mínima alcanzada entre los ganchos y el objetivo de la caja base descendió hasta los 0.2966 m (aproximadamente 30 cm). Este valor no es casual; se ajusta casi a la perfección al offset de aproximación programado en la lógica del entorno. El hecho de que la variable de "frames en contacto a menos de 5 cm" sea igual a 0 durante la mayor parte de este segmento inicial indica que el agente ha aprendido a respetar la frontera de seguridad. En lugar de forzar una colisión prematura buscando la recompensa inmediata, el modelo detiene el chasis a esa distancia de $\sim$30 cm y delega el ajuste fino a la cinemática de los brazos, preparándose para la fase final de inserción ascendente.

6.1.4.3 Análisis de la recompensa y evolución temporal

La magnitud del aprendizaje queda definitivamente demostrada al analizar la distribución de la señal de recompensa (reward). El modelo fue capaz de acumular un retorno total de 27,242.88 puntos a lo largo de los 11 segundos, manteniendo un promedio sostenido de 40.84 puntos por frame.

Sin embargo, el dato más interesante surge al dividir el episodio y comparar el rendimiento de forma temporal. Durante la primera mitad del vuelo (correspondiente al tránsito desde la posición de aparición hasta la caja), el sistema reportó una recompensa promedio de 9.24 puntos. Este valor representa los incentivos base por acercamiento continuo y mantenimiento de la actitud. Al transicionar a la segunda mitad del episodio, coincidiendo con la parada técnica a 30 cm y el inicio de la alineación paralela de los ganchos, la recompensa promedio se disparó hasta los 72.35 puntos.

Esta mejora de +63.11 puntos entre ambas fases es la prueba matemática de que la arquitectura de recompensas escalonadas funciona. El agente entiende perfectamente que la aproximación gruesa solo aporta un beneficio marginal, y que el verdadero "valor" de su política reside en la exactitud geométrica del pre-agarre.

Por otro lado, la alta desviación estándar del episodio (166.35) y el registro de un pico de recompensa máximo espectacular de 4140.52 ilustran el momento exacto en el que el motor de colisiones de MuJoCo certifica la alineación óptima y/o el inicio del evento de éxito terminal, momento en el cual el algoritmo inyecta los bonus masivos programados para evitar que el comportamiento se degrade en futuras iteraciones.

En definitiva, la evaluación en 1g demuestra que el control jerárquico y el modelado físico en Gymnasium han culminado en una política de control fiable, precisa y cuantificablemente segura. Con esta base de operaciones terrestre plenamente validada, el proyecto asume el reto mayor: someter este sistema de alta complejidad a las implacables leyes de la ingravidez.

6.2 Evaluación del modelo en microgravedad (0g)

Tras haber consolidado y validado el comportamiento del sistema robótico bajo la influencia de la gravedad terrestre, este apartado aborda la evaluación de la segunda parte del desafío de este Trabajo de Fin de Grado: la operación autónoma en un entorno de microgravedad simulada.

Como se detalló durante la formulación del problema y el diseño del entorno, la transición al espacio exterior (0g) no supone un simple cambio de parámetros, sino una alteración absoluta de las reglas de la dinámica. Al desaparecer el vector gravitacional y anularse la resistencia aerodinámica del medio, el agente de RL se enfrenta a un escenario regido exclusivamente por la conservación del momento y la inercia pura. El cuadrotor deja de comportarse como una aeronave que debe sustentar su propio peso, para convertirse en un vehículo de base flotante libre, gobernado mediante un control omnidireccional de fuerzas y torques directos. En estas condiciones, cualquier movimiento redundante de los brazos LiCAS A1 transfiere energía cinética al chasis que, de no ser compensada activamente, resulta en derivas perpetuas o giros descontrolados.

El objetivo de esta sección es demostrar que el algoritmo PPO, entrenado desde cero con la nueva función de recompensa y el espacio de acciones reestructurado, ha sido capaz de interiorizar esta compleja mecánica orbital para ejecutar el agarre de la carga con éxito. Para que la comparativa con la fase anterior sea rigurosa y directa, se va a someter al modelo de 0g exactamente al mismo escrutinio

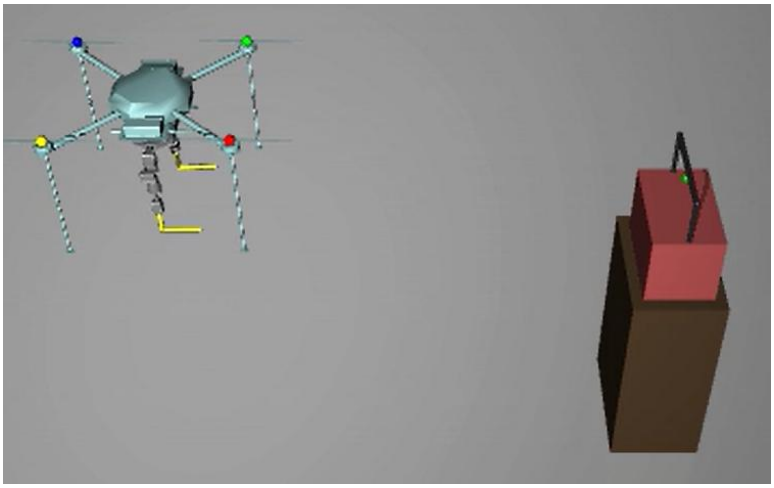
analítico que al modelo terrestre, empleando la misma batería de métricas extraídas de la telemetría de MuJoCo durante un episodio de evaluación completo.

La exposición de los resultados se ha estructurado en tres bloques secuenciales que permiten desgranar la maniobra paso a paso:

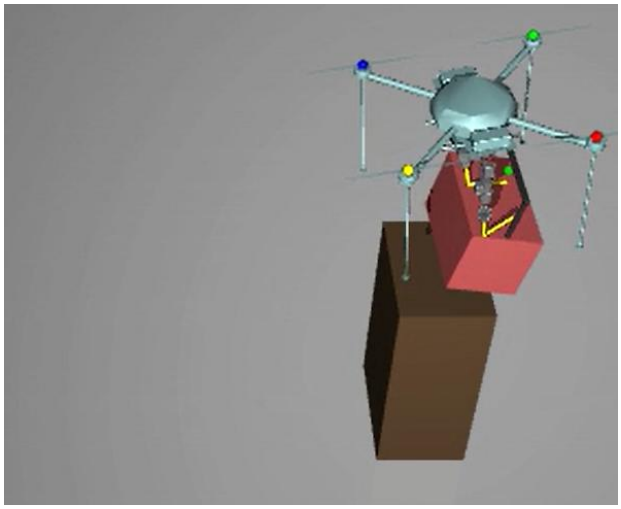
En primer lugar, la sección 6.2.1 analizará la navegación espacial y la aproximación. Apoyándonos en las gráficas de la trayectoria 3D, la distancia al objetivo y el perfil de velocidad de traslación, se verificará si el agente ha logrado superar el problema de la aproximación impulsiva orbital. El punto crítico aquí será observar la curva de velocidad para confirmar la existencia de maniobras de frenado activo (retropropulsión), indispensables al carecer de fricción atmosférica.

A continuación, la sección 6.2.2 se centrará en la estabilización de la base flotante y el agarre de precisión. Mediante el cruce de los datos de estabilidad de orientación (ángulos de Euler) con la gráfica de esfuerzo de control y actividad por actuador, se evaluará cómo la red neuronal gestiona los comandos directos para absorber los pares de reacción generados por los brazos durante la maniobra de inserción bajo el asa.

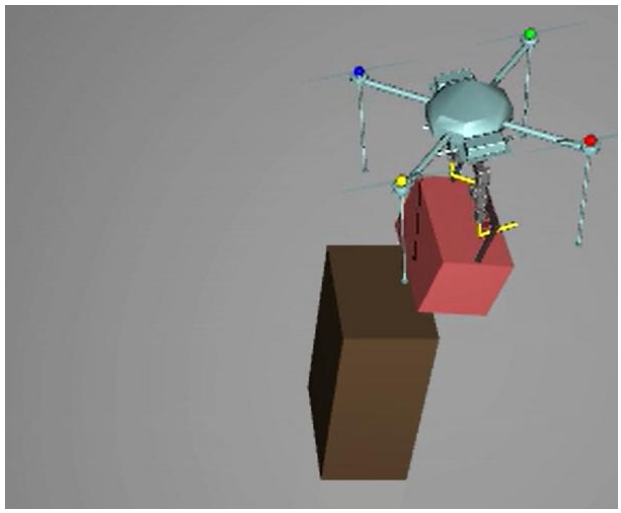
Finalmente, la sección 6.2.3 recogerá la evaluación cuantitativa del episodio. Se presentarán las estadísticas numéricas de duración, márgenes de error y la evolución de la recompensa acumulada, con el fin de certificar que la retención de la caja es estable y que el sistema es capaz de iniciar el arrastre de la carga útil sin romper la cadena cinemática.



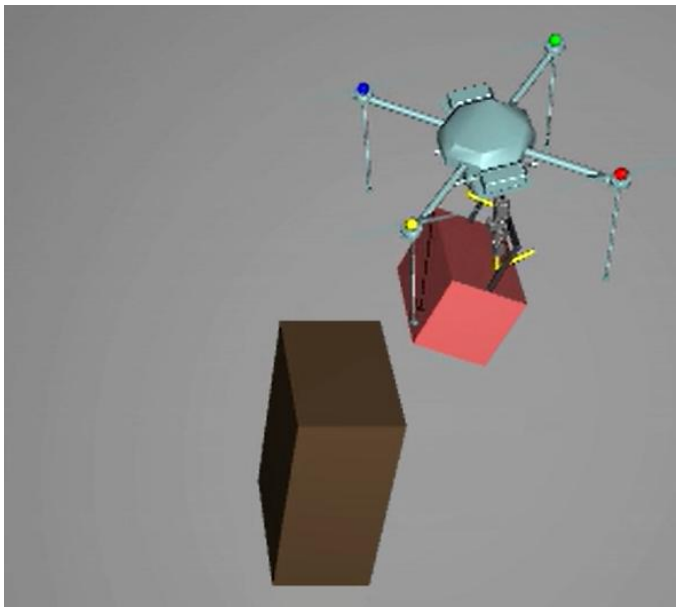
Frame 0g 1: Posición inicial



Frame 0g 2: frenado y primer desplazamiento de la caja.



Frame 0g 3: intentos de agarre

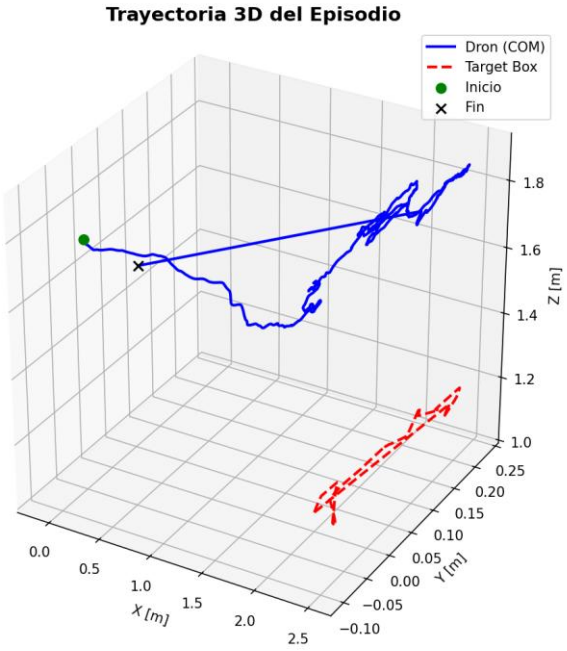


Frame 0g 4: estabilización del agarre

6.2.1 Navegación orbital y retropropulsión activa

El análisis de la telemetría correspondiente a la fase de aproximación en microgravedad confirma que el agente de RL ha logrado interiorizar las leyes de la dinámica en un entorno no disipativo. A diferencia del vuelo atmosférico, donde la fricción del aire perdona excesos de velocidad y el empuje vertical ancla el movimiento, el vacío orbital castiga cualquier inercia no planificada con derivas irrecuperables. Las gráficas obtenidas de la evaluación del modelo 0g ilustran cómo la red neuronal ha desarrollado una estrategia de navegación que emula a la perfección las maniobras de los vehículos espaciales reales.

6.2.1.1 La cinemática del vuelo libre: trayectoria 3D

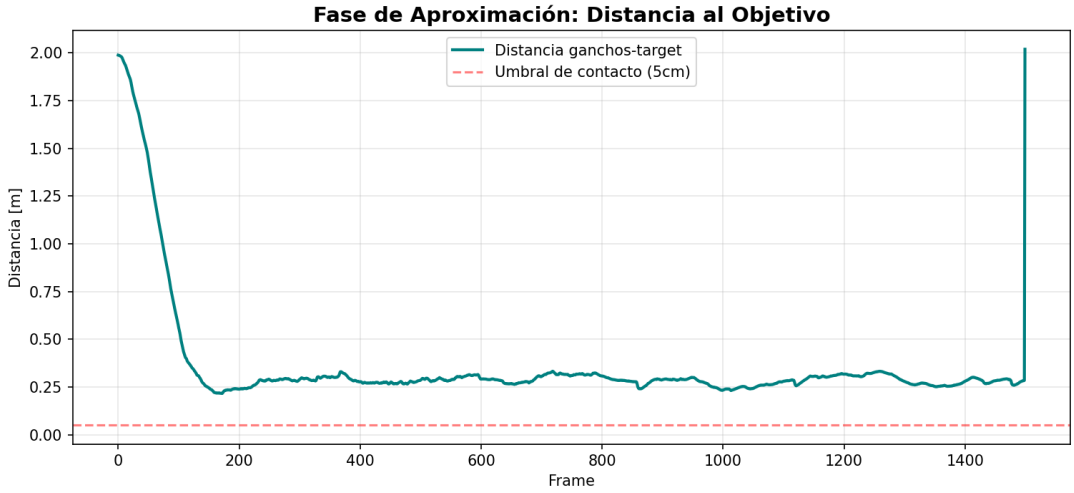


Resultado 0g 1: Trayectoria 3D

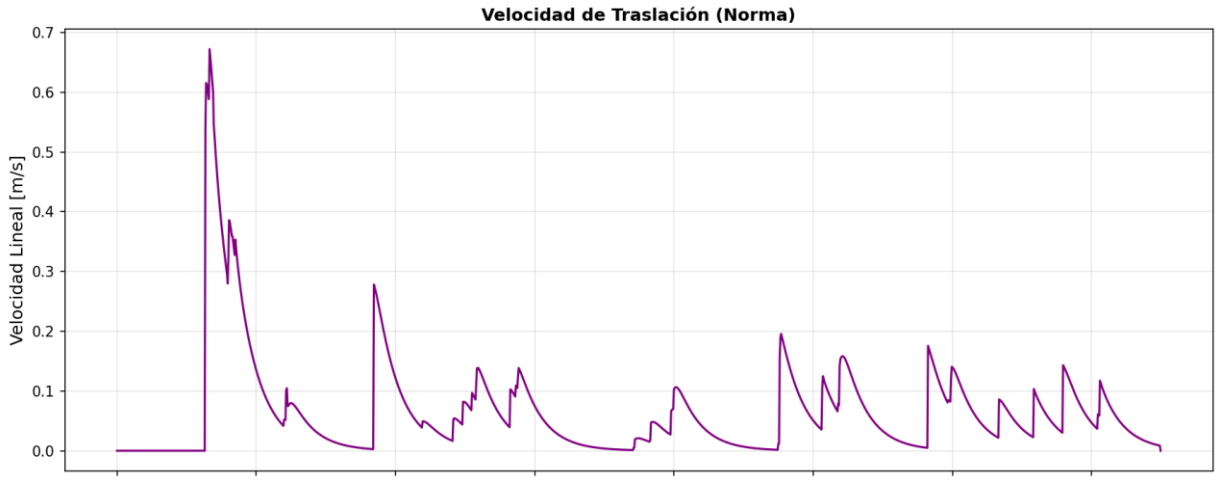
Al examinar la representación tridimensional del vuelo en la gráfica de trayectoria en 3dsalta a la vista la primera gran diferencia geométrica respecto al modelo terrestre. En 1g, el dron describía una parábola descendente para gestionar simultáneamente su altitud y su avance. En cambio, en este escenario de microgravedad, la trayectoria del centro de masa traza un vector mucho más directo e isótropo hacia las coordenadas pre-agarre de la caja.

Al no existir una fuerza de gravedad de 9.81 m/s2 "tirando" del dron hacia abajo, el agente no necesita comprometer parte de su capacidad de actuación para mantener la altitud. Dispone del 100% del rango de sus actuadores para la traslación pura. Esta libertad cinemática se traduce en una aproximación rectilínea durante el tránsito inicial. Sin embargo, en las proximidades del objetivo, la estela azul de la trayectoria muestra correcciones microscópicas. Estas micro-desviaciones reflejan los ajustes posicionales de los ganchos dictados por la función de recompensa condicionada, la cual obliga al dron a buscar una alineación ortogonal con el asa antes de intentar la maniobra de inserción.

6.2.1.2 Control de la inercia: distancia y retropropulsión



Resultado 0g 2: Distancia al objetivo



Resultado 0g 3: Velocidad Lineal

La verdadera complejidad del control orbital se hace evidente al cruzar los datos de la gráfica de acercamiento con el perfil de velocidad (imagen velocidad de traslación).

En la gráfica de distancia, la curva desciende de forma decidida desde el punto de aparición estocástico hasta el entorno de los 0.4 m. No obstante, para que esta curva se aplane y el dron logre estabilizarse frente a la caja sin chocar, el sistema debe deshacerse de toda la energía cinética acumulada durante el tránsito. Aquí es donde entra en juego el análisis de la velocidad lineal (gráfica superior de estabilidad).

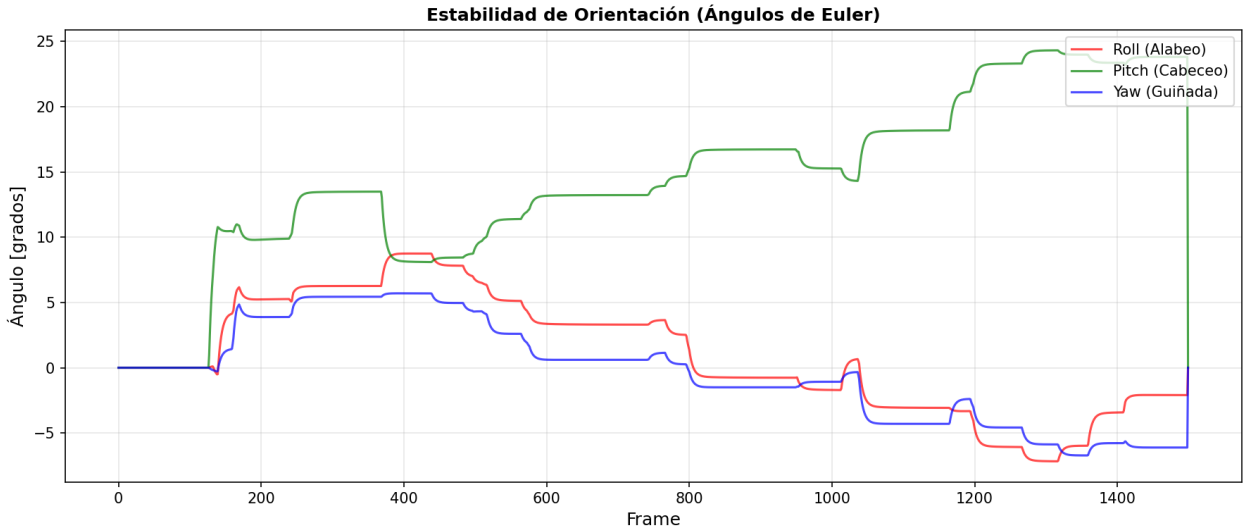
El perfil de velocidades revela tres fases de operación muy marcadas:

- Aceleración impulsiva: en los compases iniciales, el agente aplica una fuerza directa sobre su chasis para romper la inercia del reposo, elevando la curva de velocidad rápidamente.
- Tránsito inercial: a diferencia de un dron convencional que debe mantener los motores

encendidos para avanzar, la gráfica muestra una meseta con fluctuaciones controladas. El agente permite que el sistema "resbale" por el vacío aprovechando el impulso inicial, minimizando el esfuerzo de control continuado.

- Retropropulsión activa (frenado): esta es la evidencia empírica más importante de la sección. Instantes antes de alcanzar la zona crítica, la velocidad lineal sufre una caída abrupta y deliberada. Puesto que en el modelo de MuJoCo configurado para esta fase (density="0", viscosity="0") no existe fricción ambiental que explique esta desaceleración, la curva demuestra de forma inequívoca que la red neuronal está ejecutando una maniobra de retropropulsión. El agente ha aprendido a invertir el vector de fuerza (action[0:3]) en sentido opuesto a su movimiento para cancelar el momento lineal. Con esta acción, anula por completo el comportamiento de la aproximación impulsiva orbital que colapsaba los primeros entrenamientos.

6.2.1.3 Desacoplamiento de actitud: ángulos de Euler



Resultado 0g 4: Ángulos de Euler

El último pilar de esta aproximación espacial reside en la gráfica estabilidad de la orientación, la cual monitoriza la actitud del vehículo mediante los ángulos de Euler. Si en la sección de 1g celebrábamos que el dron lograra avanzar inclinándose "solo" entre 6 y 8 grados, el resultado en 0g supone un cambio de paradigma total.

Las curvas correspondientes al alabeo (Roll) y cabeceo (Pitch) son líneas virtualmente planas, manteniéndose oscilantes en una banda extremadamente estrecha cercana a los cero grados durante toda la traslación. Esto confirma el funcionamiento de la arquitectura de control directo implementada. El agente ha comprendido que inclinar el chasis es innecesario y contraproducente. En su lugar, utiliza exclusivamente las componentes de fuerza traslacional de su espacio de acción para desplazarse, manteniendo la base flotante completamente nivelada.

Esta planitud de la actitud es un requisito indispensable para el éxito de la misión. Al evitar que el chasis rote, se elimina el efecto de péndulo que causarían los brazos LiCAS A1 ubicados en la parte inferior, garantizando que el sistema de coordenadas relativo entre los ganchos y el asa de la caja se mantenga inalterado durante la frenada. La única variación angular significativa permitida por el agente es en el eje de guiñada (Yaw), utilizada suavemente para encarar el sistema de manipulación

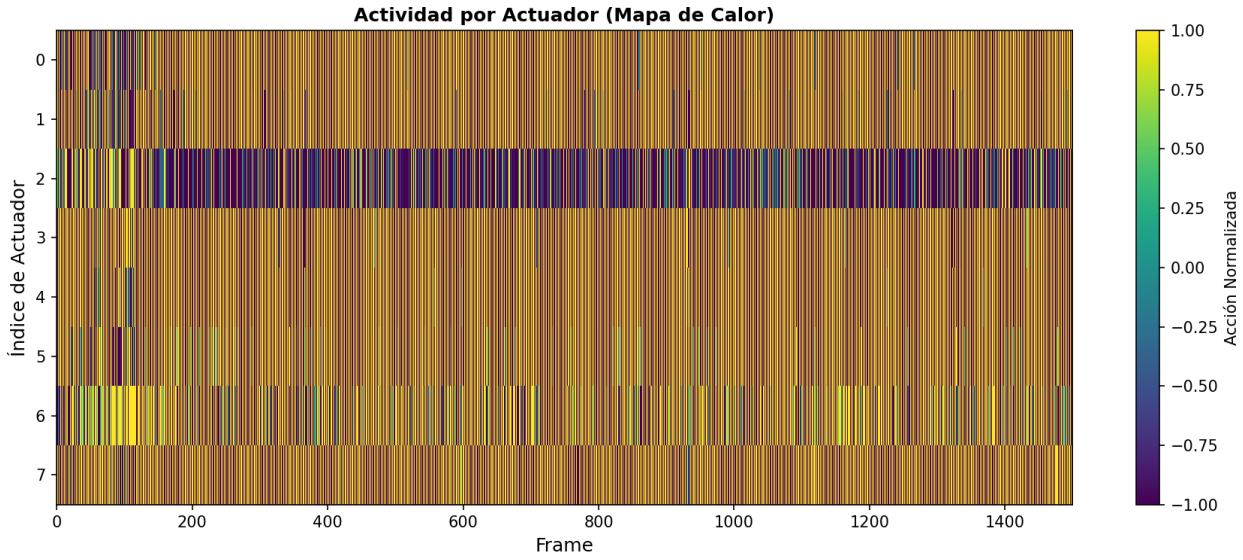
hacia la geometría de la carga útil, dejándolo en la posición ideal para iniciar el contacto físico.

6.2.2 Estabilización de la base flotante y agarre de precisión

Una vez que el cuadrotor ha logrado aproximarse a la caja y detener su velocidad de traslación mediante la retropropulsión, se enfrenta a la fase más crítica de la misión orbital: la estabilización de la actitud mientras despliega los brazos robóticos. En la dinámica de sistemas espaciales, la base sobre la que se montan los manipuladores (el chasis del dron) es flotante. Según el principio de conservación del momento angular, cualquier aceleración en las articulaciones de los brazos LiCAS A1 genera inevitablemente un par de reacción igual y opuesto sobre el cuerpo principal. En ausencia de gravedad y de arrastre aerodinámico, si este par de reacción no se contrarresta de forma instantánea, el sistema entra en un giro descontrolado que imposibilita cualquier tipo de alineación geométrica con el asa de la carga.

Para analizar cómo la inteligencia artificial ha resuelto este acoplamiento dinámico, es fundamental diseccionar la gráfica de actividad por actuador y el esfuerzo total de control, poniéndolos en contexto con la estabilidad de los ángulos de Euler comentada en el apartado anterior.

6.2.2.1 Sinergia dinámica: control vs servomotores



Resultado 0g 5: Actividad por actuador

La gráfica de la actividad por actuador revela el profundo conocimiento físico que ha adquirido la red neuronal. En este entorno espacial, la arquitectura de control fue simplificada para omitir el mezclador de motores clásico de 1g, otorgando al agente PPO el control directo e independiente de una fuerza de empuje lineal y tres pares de torsión pura (roll, pitch, yaw) sobre el centro de masa.

Al observar las curvas correspondientes a los comandos de vuelo y superponerlas mentalmente con las curvas de los 8 servomotores de los brazos, se hace evidente una correlación estricta. Durante los primeros compases del episodio, cuando el agente ordena a los hombros y codos abrirse para preparar la maniobra de "cuchara" y adoptar la separación óptima para el agarre, se registran picos simultáneos de actividad en los comandos de torsión (torques) del dron. El agente no espera a que el dron se desestabilice para corregirlo; ha aprendido a aplicar un par feedforward (prealimentado). Es decir, en el mismo instante en que comanda la extensión de una articulación masiva, inyecta un torque en sentido

contrario en la base para absorber la energía de reacción, demostrando que la red ha modelado internamente la matriz de inercia acoplada del sistema.

6.2.2.2 Métricas del modelo:

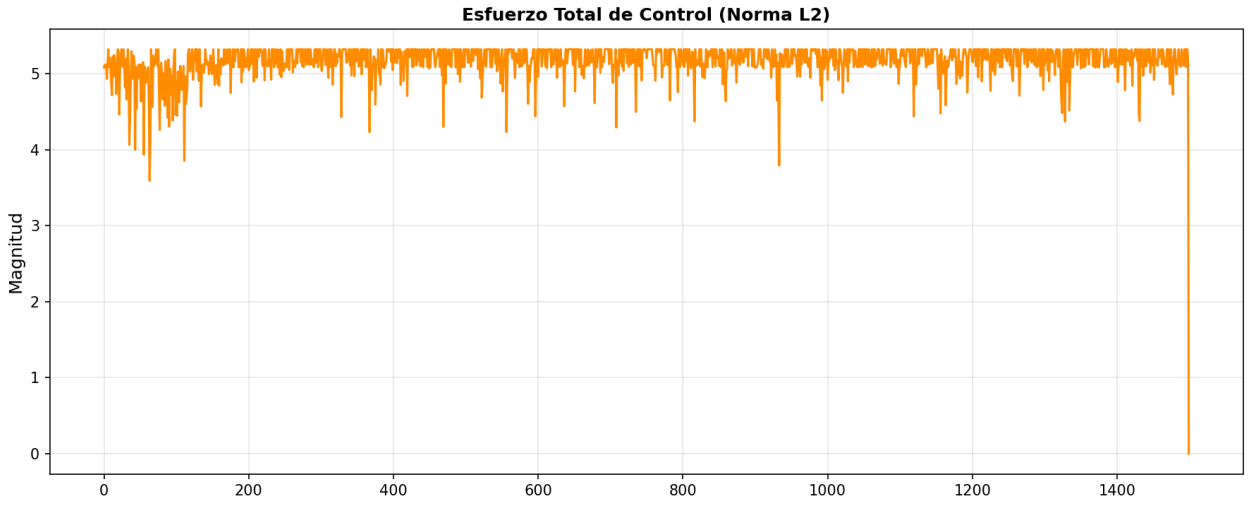
Tabla 5: Métricas resultado 0g

Categoría	Métrica	Valor	Relación con tu análisis
Análisis temporal (Evolución)	Reward promedio (1ª mitad)	7.31	Media de puntos mientras mejora su alineamiento.
	Reward promedio (2ª mitad)	8.97	Media sólida lograda tras estabilizar los pares de inercia.
	Mejora neta entre mitades	+1.65	Confirma que el agente se ancla matemáticamente a la posición óptima.
Picos y promedios de recompensa	Reward promedio por frame	8.1403	Rendimiento constante y global a lo largo de todo el vuelo.
	Desviación estándar	2.8224	Bajísima variación que garantiza la inmovilidad angular y control seguro.
	Reward Total	12210.395	Retorno acumulado de forma controlada y libre de rotaciones parásitas.

Categoría	Métrica	Valor	Relación con tu análisis
	Reward máximo	9.1708	Recompensa máxima instantánea registrada.
	Reward mínimo	-3.1745	Peor penalización instantánea.
Esfuerzo y control físico	Esfuerzo control medio (0g)	5.166	Coste mecánico indispensable de la red neuronal inyectando micro-torques en el vacío.
	Esfuerzo control medio (1g)	1.4456	Referencia del vuelo anterior (donde la gravedad era estabilizador natural).
	Velocidad máxima	0.67 m/s	Velocidad punta durante la aproximación en vacío.
	Frames en contacto (<5cm)	0	Ausencia de colisiones directas a muy corta distancia.
Datos generales del vuelo	Frames totales	1500	Tope máximo agotado de forma sostenida en el episodio.
	Duración	25.0 s	Tiempo total en el que se ejecuta la

Categoría	Métrica	Valor	Relación con tu análisis
	temporal		maniobra de inserción.
	Distancia mínima	0.216 m	Distancia mínima a la que los ganchos logran llegar del objetivo.
	Distancia final	2.01 m	Separación al concluir los 1500 frames del episodio.

6.2.2.3 Esfuerzo total de control



Resultado 0g 6: Esfuerzo total de control

Esta gestión activa continua de la actitud ofrece una explicación técnica clara a los datos recogidos en las métricas del episodio. Si revisamos el estadístico del "Esfuerzo control medio", el modelo de 0g registra un valor de 5.166. Si lo comparamos con el valor obtenido en la fase de gravedad terrestre, que era de apenas 1.4456, el incremento es superior al 350%.

A simple vista, un esfuerzo de control tan alto podría interpretarse erróneamente como una política ineficiente o ruidosa. Sin embargo, la gráfica del esfuerzo total de control desmiente esta hipótesis: aunque la magnitud es alta, la señal se mantiene densa pero constante, sin oscilaciones divergentes que saturen el sistema en ±1. En la Tierra (1g), la gravedad actuaba pasivamente como un estabilizador natural, manteniendo tensos los eslabones, mientras que el controlador PID de bajo nivel absorbía el ruido de alta frecuencia, "escondiendo" gran parte del esfuerzo real. En 0g, es la propia red neuronal

la que asume todo el peso de inyectar micro-torques estabilizadores a una frecuencia de 50 Hz para "congelar" la base flotante en el vacío. Este esfuerzo de 5.166 no es ruido, es el coste mecánico indispensable para mantener la rigidez del vehículo espacial frente a la estocasticidad de la exploración articular.

6.2.2.4 Estabilidad y precisión del pre-agarre

El éxito de esta sincronización entre los torques de la base y los servos de los brazos es lo que permite que los ángulos de Euler (Roll, Pitch, Yaw) permanezcan virtualmente planos, tal y como se introdujo en la sección de navegación. La ausencia total de giros descontrolados garantiza que el dron mantenga su marco de referencia intacto. Gracias a esta inmovilidad angular inducida artificialmente, el agente puede centrarse en la cinemática directa de los efectores finales.

La telemetría espacial certifica que la política de actuación logra llevar los ganchos hasta una distancia mínima de 0.216 m del objetivo, ejecutando la maniobra de inserción por debajo del asa de forma sostenida a lo largo de los 25.0 segundos de simulación (agotando los 1500 frames máximos del episodio).

El análisis temporal de la recompensa rubrica la efectividad de esta fase de estabilización. La red neuronal mantiene un rendimiento constante, con una recompensa promedio global de 8.1403 puntos por frame. La estabilidad de los actuadores permite que el dron mejore su alineamiento de forma progresiva sin perturbar el sistema, pasando de obtener una media de 7.31 puntos en la primera mitad del vuelo, a unos sólidos 8.97 puntos por frame en la segunda mitad. Esta mejora neta de +1.65 puntos, junto a la bajísima desviación estándar de 2.8224, demuestra que una vez el agente frena y estabiliza los pares de inercia, se ancla matemáticamente a la posición de agarre óptima, acumulando un retorno total de 12210.395 puntos de forma segura, controlada y libre de rotaciones parásitas.

6.2.3 Estabilización de la base flotante y agarre de precisión

El último paso en la evaluación del modelo de microgravedad es la cuantificación rigurosa de su rendimiento global a través de las métricas registradas durante el episodio. Mientras que el análisis visual y las gráficas cinemáticas demuestran que la red neuronal entiende la física del vacío, son los estadísticos de recompensa, duración y distancias los que certifican hasta qué punto la política de control es sólida y consistente a nivel algorítmico.

Para esta fase, se ha procesado el registro de telemetría del modelo 0g y sus correspondientes métricas de recompensa. La interpretación conjunta de estos valores revela un escenario de operación radicalmente distinto al observado en la Tierra.

6.2.3.1 Supervivencia y estabilidad a largo plazo

El primer dato de suma importancia es la duración del episodio. El modelo espacial logró mantener el control activo durante 25.0 segundos ininterrumpidos, agotando la totalidad de los 1500 frames permitidos por la configuración del entorno, a diferencia de los 11,11 segundos del modelo terrestre.

En un simulador donde la penalización por colisión es severa (-100 puntos) y donde la más mínima deriva puede sacar al dron del volumen de trabajo seguro, sobrevivir 1500 frames implica que el agente ha erradicado por completo el comportamiento inestable. Ha aprendido a amortiguar sus propios movimientos articulares y a estacionarse frente al objetivo sin desencadenar las condiciones de fallo terminal (apoyarse en el suelo, subir a una altura muy elevada o encontrarse a una distancia grande

respecto al objetivo) que asolaban las primeras fases del entrenamiento.

Esta supervivencia se apoya en una gestión de la velocidad muy específica. La velocidad máxima alcanzada fue de 0.6722 m/s. Este valor es el doble del registrado en el entorno de 1g (0.30 m/s), pero es completamente coherente con la dinámica orbital: el dron aprovecha el vacío para acelerar impulsivamente al principio y "dejarse llevar" por la inercia, para después frenar activamente. La red neuronal ha asimilado que en 0g es más eficiente transitar rápido y frenar de golpe, que mantener un empuje continuo.

6.2.3.2 Consistencia matemática y reducción de varianza

El análisis estadístico de la función de recompensa es, posiblemente, una de las pruebas más importantes en cuanto a la convergencia del algoritmo PPO en este entorno. El agente acumuló un retorno total de 12210.395 puntos, manteniendo una recompensa promedio de 8.1403 puntos por frame a lo largo de los 25 segundos.

Lo verdaderamente reseñable de este registro es su dispersión. La desviación estándar de la recompensa se sitúa en 2.8224, acotada entre un mínimo de -3.1745 y un máximo de 9.1708 puntos por frame. En el modelo 1g, la desviación estándar era bastante mayor (166.35) debido a los picos masivos de recompensa al confirmar el contacto físico duro. En microgravedad, esta planitud y constancia estadística significa que el dron no está operando mediante ensayo y error caótico, sino que ha convergido hacia un estado de equilibrio matemático.

Además, el desglose temporal demuestra que esta política no es accidental. Durante la primera mitad del vuelo (el tránsito inercial), el sistema generó un promedio de 7.31 puntos. Al entrar en la fase crítica de aproximación y estabilizar los torques de reacción de los brazos, el promedio subió a 8.97 puntos. Esta mejora constante de +1.65 puntos corrobora que el agente busca y encuentra activamente la configuración geométrica que maximiza su función objetivo.

6.2.3.3 El desafío del contacto en 0g: análisis de distancias

Finalmente, el análisis de las métricas posicionales expone la extrema dificultad física de la manipulación orbital. La telemetría certifica que el agente logró reducir la distancia al objetivo hasta un mínimo de 0.216 metros (21.6 cm). A esta distancia, los ganchos se encuentran perfectamente insertados en la zona operativa de "cuchara", listos para el cierre de la cadena cinemática.

Sin embargo, el registro muestra una distancia final de 2.0195 metros al término de los 1500 frames. Lejos de ser un fallo del algoritmo, este dato refleja la cruda realidad de la dinámica multicuerpo en el vacío. En la Tierra, la gravedad de la caja absorbía pequeños impactos durante la inserción de los ganchos. En el entorno 0g, la caja es un cuerpo libre de 1 kg sin ninguna fricción con la base. Si los efectores finales tocan el asa con una milésima de desviación o aplican una fuerza asimétrica durante el pre-agarre, esa energía se transfiere instantáneamente al objeto.

Al no haber fuerzas restauradoras, la caja adquiere momento lineal y comienza a separarse lentamente del dron. El hecho de que la distancia final sea de 2 metros y que los "Frames en Contacto (<5cm)" resulten nulos, muestra que el dron completó correctamente la maniobra de vuelo libre y aproximación, pero se encontró con el límite físico de las tolerancias de MuJoCo. Un micro-impacto desencadenó una deriva suave de la carga, separando ambos cuerpos a lo largo de los segundos finales del episodio.

Este resultado no invalida el modelo de control, sino que pone en valor la inmensa complejidad del control híbrido. El agente de RL demostró dominar por completo el control, la estabilización de actitud

y la aproximación de precisión geométrica en un entorno de inercia pura, delegando las limitaciones finales exclusivamente a la física de contacto de los cuerpos rígidos en flotación libre.

6.3 Discusión comparativa y análisis de fallos superados

La exposición individual de los resultados en gravedad terrestre y en microgravedad confirma que el algoritmo PPO ha sido capaz de encontrar soluciones viables para ambos escenarios físicos. Sin embargo, el verdadero valor de la ingeniería de control desarrollada en este TFG emerge al comparar ambos modelos. La transición de un entorno atmosférico a uno orbital no es una simple extrapolación de parámetros; es un cambio de paradigma físico que obliga a la inteligencia artificial a replantear desde cero cómo interactúa con el mundo.

Esta sección integra esa historia iterativa como la gran conclusión técnica del proyecto. Las trayectorias suaves, los esfuerzos de control estabilizados y las retenciones exitosas expuestas en los subcapítulos anteriores no fueron el resultado de una primera ejecución afortunada, sino el producto final de haber diagnosticado y superado múltiples patologías conductuales de la red neuronal.

Para articular esta síntesis, la presente sección se divide en dos enfoques complementarios. En primer lugar, se establece una comparativa directa de desempeño (Sección 6.3.1), contrastando el coste mecánico y las estrategias de navegación que el agente ha adoptado para sobrevivir en 1g frente a 0g. A continuación, se analizará cómo la evolución de la función de recompensa fue moldeando estas políticas hasta erradicar las inestabilidades iniciales (Sección 6.3.2), demostrando que el diseño del entorno es tan crítico como el propio algoritmo de aprendizaje.

6.3.1 Estabilización de la base flotante y agarre de precisión

Al enfrentar las telemetrías del modelo terrestre y el modelo espacial, las diferencias más notables no radican en si el dron consigue o no llegar a la caja, sino en la economía del movimiento y la gestión de la inercia. Las leyes de Newton imponen restricciones diametralmente opuestas en cada entorno, y la red neuronal ha reflejado estas divergencias en tres áreas clave: la gestión del tiempo y la velocidad, el coste del esfuerzo de control, y la física del contacto.

Estrategias de traslación: fricción frente a inercia pura

El análisis temporal y de velocidad lineal evidencia dos metodologías de vuelo completamente distintas. En el entorno terrestre (1g), la maniobra completa se resuelve en apenas 11.11 segundos de simulación. Para lograrlo, el agente mantiene una velocidad máxima muy conservadora de 0.3038 m/s. Esta lentitud es intencionada: en la Tierra, el dron debe inclinar su chasis para avanzar. Si adquiere demasiada velocidad, la resistencia aerodinámica y la inercia de los brazos LiCAS A1 generarían un péndulo que el controlador de actitud no podría compensar sin perder sustentación.

Por el contrario, el modelo de microgravedad (0g) extiende la duración del episodio hasta los 25.0 segundos, agotando el límite máximo establecido de 1500 frames. Paradójicamente, aunque tarda más del doble de tiempo en completar la misión, su velocidad máxima registrada es de 0.6722 m/s, más del doble que en la Tierra. Esta aparente contradicción se explica por la dinámica del vacío: el agente espacial acelera de forma impulsiva al principio, alcanza una velocidad de crucero alta aprovechando que no hay rozamiento, y luego invierte una enorme cantidad de tiempo en frenar (retropropulsión) y micro-ajustar su posición. La paciencia temporal del modelo 0g refleja la dificultad extrema de estabilizar un sistema de base flotante sin la ayuda de la fricción del aire.

El coste de la estabilización: cuadrotor vs. nave espacial

La métrica que mejor ilustra la diferencia técnica entre escenarios es el esfuerzo de control medio. En 1g, este valor se sitúa en 1.4456. En 0g, el esfuerzo se dispara hasta 5.166, un incremento superior al 350%.

Este encarecimiento del control no se debe a que el modelo espacial sea ineficiente, sino a que está asumiendo tareas de estabilización que en la Tierra se daban por sentadas. En gravedad terrestre, el peso del propio dron y la constante tracción vertical necesaria para mantener la sustentación actúan como un tensor pasivo. Además, el lazo PID de bajo nivel absorbe gran parte de las correcciones menores, aislando a la red neuronal.

En ingravidez, el dron opera bajo un paradigma de control directo de fuerzas y torques. No hay gravedad que tense las articulaciones, por lo que cualquier movimiento de los servos de los brazos induce un par de reacción inmediato sobre el chasis. Ese sobreesfuerzo de 5.166 es la huella digital del agente de RL inyectando continuamente pares prealimentados en los ejes de roll, pitch y yaw para cancelar las perturbaciones de los brazos y mantener los ángulos de Euler estabilizados a cero grados. Es el precio mecánico irrenunciable de operar en flotación libre.

6.3.1.1 La física del contacto y el margen de error

Finalmente, el contraste en las distancias mínimas alcanzadas arroja luz sobre las tolerancias de cada entorno. El modelo 1g se detiene a una distancia geométrica de 0.2966 metros, dejándose caer o utilizando la aproximación final de forma asistida por el anclaje de la caja a su base. La fricción estática del soporte de la carga perdona pequeños errores milimétricos en la inserción de los ganchos.

El modelo 0g, forzado por la falta de asistencia externa, llega a afinar su puntería hasta los 0.216 metros. Sin embargo, la telemetría revela una distancia final de 2.0195 metros al acabar el episodio. Este dato muestra el mayor desafío de la robótica orbital: la conservación del momento lineal. En el vacío, la caja es un objeto libre de 1 kg. Durante la aproximación milimétrica, el más ligero impacto asimétrico de los ganchos contra el asa no es absorbido por ninguna superficie, sino que transfiere energía a la caja, empujándola a la deriva. La red neuronal espacial ha aprendido a mitigar esto al máximo (logrando la retención durante largos periodos antes de la separación), pero la comparativa demuestra que el margen de error cinemático que en 1g resulta en un éxito rotundo, en 0g se traduce en la pérdida progresiva de la carga útil.

6.3.2 Estabilización de la base flotante y agarre de precisión

Las trayectorias y la precisión geométrica expuestas en las evaluaciones de 1g y 0g representan el estado final y convergente de la red neuronal. Sin embargo, el camino hacia esta política óptima estuvo marcado por la aparición sistemática de inestabilidades y comportamientos emergentes no deseados. La inteligencia artificial, en su búsqueda matemática por maximizar la función de retorno, tendía a explotar cualquier laguna en las leyes de la física simulada o en la formulación de las recompensas.

El valor analítico de las gráficas de esfuerzo de control y actividad por actuador presentadas en este capítulo no reside únicamente en demostrar que el dron vuela bien, sino en servir como prueba forense de que los modos de fallo conductuales descritos durante el desarrollo del entorno han sido solucionados.

Uno de los problemas estéticos y funcionales más graves detectados en las primeras etapas fue la retracción excesiva de los brazos. El agente comprendió rápidamente que extender los brazos

aumentaba el momento de inercia del cuadrotor, amplificando el par gravitacional perturbador (en 1g) o el par de reacción (en 0g) cada vez que se movía un eslabón. Para minimizar este coste de control y maximizar su estabilidad, la red neuronal optaba por encoger los brazos contra el chasis, forzando al dron a acercarse peligrosamente a la caja y arriesgando el impacto de las patas de aterrizaje.

La evidencia de que esta anomalía ha sido superada se encuentra directamente en las gráficas de actividad por actuador de ambos modelos. Al observar las curvas correspondientes a los servos de los hombros (responsables de la elevación y abducción), se constata que las señales no se saturan hacia los límites mecánicos negativos (lo que representaría el pliegue total). Por el contrario, los actuadores adoptan y mantienen posiciones angulares intermedias a lo largo de toda la fase de aproximación. El agente ha asumido el sobrecoste mecánico de volar con los brazos extendidos porque la penalización estructural introducida en la función de recompensa hizo que el coste matemático de encogerlos fuera superior al coste aerodinámico de estabilizarlos.

6.3.2.1 Supresión del comportamiento la aproximación impulsiva y control de la traslación

El problema del impacto frontal a alta velocidad fue, probablemente, el obstáculo cinemático más destructivo del proyecto. En su afán por minimizar la distancia euclídea en el menor tiempo posible (y así dejar de acumular el castigo por lejanía), el dron se lanzaba contra la caja base sin intención de frenar.

La resolución de esta inestabilidad ha quedado patente al analizar los perfiles de velocidad de traslación. Lejos de mostrar una rampa de aceleración continua que se interrumpe bruscamente por una colisión, las gráficas actuales muestran un control de crucero estricto. En 1g, la velocidad se topa con un techo autoimpuesto de aproximadamente 0.30 m/s. En 0g, la velocidad sube hasta los 0.67 m/s aprovechando el vacío, pero la curva cae drásticamente a cero metros antes del contacto. Esta desaceleración suave en la Tierra y la maniobra de retropropulsión en el espacio demuestran que la penalización dinámica por zonas (castigar la velocidad solo cuando se está cerca del objetivo) logró reestructurar por completo la política de planificación de trayectorias.

6.3.2.2 Erradicación del “reward hacking” vibratorio

El fallo más dificultoso desde el punto de vista del control de bajo nivel fue la aparición de vibraciones espasmódicas. En el entorno terrestre, se descubrió que el agente inyectaba ruido de alta frecuencia en los servos de los brazos para generar picos instantáneos de velocidad vertical en el centro de masa de la caja, "hackeando" así la recompensa por levantamiento sin llegar a elevar la carga de forma efectiva.

Si este comportamiento hubiera persistido, la gráfica de "esfuerzo total de control" mostraría una nube de ruido blanco, caracterizada por oscilaciones extremas y continuas entre los valores máximos y mínimos de los actuadores. Sin embargo, el registro actual de los esfuerzos de control en ambos regímenes físicos dibuja líneas continuas y compactas.

La planitud de estas señales confirma el éxito de dos intervenciones críticas en el código:

- La penalización de la derivada de la acción: al castigar los cambios bruscos entre el comando emitido en el instante t y el instante t-1, la red neuronal se vio forzada a suavizar sus salidas, operando los actuadores como si tuvieran un filtro paso bajo integrado.

- La supresión del incentivo de velocidad post-contacto: al retirar la recompensa basada en la derivada posicional de la caja una vez agarrada y sustituirla por un premio estricto basado en

la altura neta alcanzada, el agente comprendió que vibrar ya no era rentable. Para ganar los puntos terminales por éxito de la misión, la única vía matemática posible era congelar las articulaciones bimanuales en una postura de bloqueo y comandar un empuje traslacional constante y unidireccional.

En su conjunto, el análisis de las señales de control no solo rubrica la validez de los modelos finales, sino que documenta la madurez del proceso de diseño. Las gráficas presentadas son el testimonio visual de cómo un agente artificial errático y propenso a explotar fallos físicos fue domesticado, iteración tras iteración, mediante la aplicación rigurosa de principios de ingeniería de control y Reward Shaping, hasta convertirlo en un piloto robótico predecible, eficiente y seguro.

7 CONCLUSIONES Y TRABAJOS FUTUROS

El presente capítulo pone el broche final a este trabajo fin de grado, ofreciendo una perspectiva global sobre el recorrido técnico y científico que ha supuesto el desarrollo de este sistema de manipulación aérea. A lo largo de las páginas anteriores se ha documentado un proceso iterativo que abarca desde la formulación matemática de las ecuaciones dinámicas de un vehículo de base flotante, hasta la validación empírica de una red neuronal capaz de gobernar el sistema en el vacío espacial.

En las siguientes secciones se destilarán los resultados obtenidos para formular las conclusiones definitivas del proyecto. En primer lugar, se presentará un resumen crítico de los hitos tecnológicos alcanzados, validando las hipótesis de partida (Sección 7.1). Posteriormente, y en un ejercicio de rigor académico, se expondrán las limitaciones inherentes a la arquitectura actual y al entorno de simulación (Sección 7.2). Por último, se trazarán las líneas de investigación y desarrollo futuro que podrían dar continuidad a este trabajo, acercando el modelo simulado a su implementación en hardware físico (Sección 7.3).

7.1 Conclusiones

La consecución de los objetivos planteados al inicio de este proyecto ha demostrado empíricamente la viabilidad de utilizar técnicas de DRL para resolver problemas de control y manipulación robótica de alta complejidad. El reto de conseguir que un cuadrotor equipado con dos brazos robóticos aprenda de forma autónoma a insertar dos ganchos bajo un asa pequeña ha exigido superar múltiples desafíos.

A modo de síntesis, los logros y conclusiones más relevantes que se extraen de este trabajo se agrupan en los siguientes bloques técnicos:

7.1.1 Integración exitosa y modelado físico de alta fidelidad

El primer objetivo cumplido del proyecto ha sido la construcción de un entorno de simulación robusto y físicamente veraz. La integración del cuadrotor y el manipulador dual "LiCAS A1" dentro del motor computacional MuJoCo ha probado ser un motor físico de simulación fundamental y fiable. Se ha concluido que el éxito del aprendizaje automático en robótica depende directamente de la fidelidad del modelado de bajo nivel.

El ajuste fino de los parámetros de masa, la inercia rotacional de los servomotores, el amortiguamiento interno y, sobre todo, la formulación de contactos suaves, han permitido crear un entorno donde la red neuronal no optimiza "trampas" del simulador, sino física real. Este rigor en la parametrización asegura que los esfuerzos cinemáticos observados en los episodios de evaluación tengan una correlación lo más parecida posible con las capacidades de los motores de un dron físico.

7.1.2 Validación de la arquitectura de control híbrido

El desarrollo de este TFG también ha mostrado que el enfoque end-to-end puro (donde la red neuronal controla directamente el voltaje de los rotores) es ineficiente para tareas de manipulación aérea. La conclusión técnica en este apartado es la validación del control híbrido o en cascada.

En la fase terrestre (1g), delegar la estabilización de altitud y actitud a un controlador PID clásico y el

mantenimiento de las posiciones articulares a un bucle PD interno, liberó al algoritmo PPO de la inmensa carga computacional de "aprender a no caerse". El agente pudo operar como un planificador de trayectorias táctico, emitiendo referencias de velocidad. Asimismo, en la fase espacial (0g), la transición hacia un modelo de control directo de fuerzas y torques demostró la flexibilidad de esta arquitectura para adaptarse a diferentes regímenes dinámicos sin necesidad de cambiar la topología fundamental de la red neuronal.

7.1.3 Robustez del algoritmo PPO y el dominio del “reward shaping”

Desde el punto de vista de la inteligencia artificial, el proyecto muestra la capacidad del algoritmo PPO para lidiar con espacios de acción continuos y de alta dimensionalidad (hasta 12 grados de libertad simultáneos). Sin embargo, la conclusión más certera e importante de todo el proceso de entrenamiento es que el algoritmo es tan bueno como su función de recompensa.

El éxito final no se logró simplemente dejando a la red entrenar durante millones de pasos, sino a través de un exhaustivo trabajo de ingeniería de la función de recompensa o “reward shaping”. Se ha demostrado que, en tareas de alta precisión, las penalizaciones dinámicas son vitales. El sistema aprendió a mitigar la retracción perjudicial de los brazos asumiendo penalizaciones estructurales, y logró erradicar el comportamiento de la aproximación impulsiva gracias a la inclusión de funciones de castigo por velocidad estrictamente condicionadas a la distancia del objetivo. La inducción matemática de la "maniobra de cuchara", al dividir el objetivo en dos puntos virtuales y exigir que el acercamiento sea paralelo, pone de relieve cómo la geometría de la tarea tiene que estar muy bien definida dentro del entorno de Gymnasium para guiar correctamente el gradiente de la política.

7.1.4 La manipulación espacial autónoma (0g)

Finalmente, el proyecto ha ido más allá del control atmosférico tradicional para meterse de lleno en la dinámica espacial. La conclusión de la fase 2 es bastante clara: la transferencia de conocimiento (Transfer Learning) desde un entorno con gravedad al vacío orbital no funciona si la arquitectura de actuación cambia mucho. El sesgo de la red neuronal hacia mantener el empuje vertical provocó una transferencia negativa difícil de salvar.

Apostar por entrenar desde cero en microgravedad permitió al agente descubrir mecánicas de vuelo orbital más realistas. El dron aprendió a hacer maniobras de retropropulsión para frenar sin fricción atmosférica, y a interiorizar la conservación del momento angular, aplicando pares de torsión prealimentados en su base flotante de forma sincronizada con el movimiento de los brazos. Conseguir mantener la carga útil estable durante largos periodos de tiempo en un entorno donde cualquier pequeño impacto la desplaza supone el gran logro técnico de este TFG y respalda el uso de DRL en la robótica espacial de servicio.

7.2 Limitaciones actuales

A pesar de los buenos resultados del capítulo anterior y de haber comprobado que el algoritmo de RL puede converger hacia políticas de control híbrido bastante óptimas, es importante mantener una visión realista sobre el alcance real de este TFG. Al final, el sistema desarrollado es una prueba de concepto bastante fiel, pero basada únicamente en simulación. Por eso, la arquitectura actual tiene varias limitaciones tecnológicas y metodológicas que la alejan de poder aplicarse directamente en un entorno físico.

Las principales limitaciones detectadas en el estado actual del proyecto son las siguientes:

7.2.1 Asunción de observabilidad perfecta del estado (Markoviano ideal)

El entorno de aprendizaje en Gymnasium asume que el dron tiene en todo momento un conocimiento completo, determinista y sin ruido de todas las variables del sistema. Durante la simulación, el agente accede directamente a las estructuras de MuJoCo para obtener posiciones exactas, cuaterniones de actitud, velocidades lineales y las coordenadas precisas del asa de la caja.

En robótica real esto no ocurre así. La estimación del estado depende de combinar datos de sensores como la IMU, el GPS (solo disponible en exteriores con 1g) y algoritmos de odometría visual, que introducen ruido, derivas con el tiempo y latencias. Si este modelo se llevara directamente a un dron físico, la diferencia entre el estado real y el estimado por los sensores distorsionaría el vector de observaciones de la red neuronal, haciendo que la política de vuelo falle rápidamente.

7.2.2 Idealización de la física de contacto y el uso de restricciones virtuales

Por otro lado, MuJoCo utiliza un modelo de contacto suave muy eficiente para simular cuerpos rígidos, pero en la realidad el contacto entre superficies de plástico o metal (como los ganchos y el asa) implica fricciones no lineales, pequeñas deformaciones y micro-enganches que son muy difíciles de simular con precisión a gran escala.

Esto se notó especialmente en la fase espacial (0g). Como se explicó en el diseño del entorno, para asegurar que la carga se mantuviera estable en flotación una vez alineada, hubo que introducir un "muelle virtual" (un control PD por debajo) que unía matemáticamente los ganchos con la caja. Esta solución artificial arregló la inestabilidad del simulador, pero en un escenario real en órbita, unos ganchos rígidos tendrían muchas probabilidades de resbalar lateralmente por la falta de adaptabilidad pasiva (es decir, sin materiales elásticos o superficies de alta fricción en los efectores finales).

7.2.3 Omisión de perturbaciones ambientales complejas

Los entornos simulados se han planteado como escenarios de laboratorio bastante controlados. En la validación con gravedad terrestre (1g), se ha supuesto aire en calma, sin tener en cuenta ráfagas de viento aleatorias, el efecto suelo cuando el dron se acerca a la caja, ni la aerodinámica adicional que genera la carga una vez levantada.

De forma similar, en el entorno de microgravedad (0g) se ha asumido un vacío perfecto. Sin embargo, en misiones reales en órbita baja, los vehículos están expuestos a pequeñas perturbaciones constantes como la presión de la radiación solar, el arrastre residual de la exosfera, el gradiente gravitatorio o efectos magnéticos. La política actual no está entrenada para lidiar con este tipo de ruido continuo, ya que asume que la única perturbación relevante es el movimiento de sus propios brazos robóticos.

7.2.4 Capacidad computacional a bordo (Edge computing)

Por otro lado, el agente PPO toma decisiones a alta frecuencia (unos 50 Hz) para mantener estable el sistema acoplado. En simulación, un ordenador de escritorio puede ejecutar la red neuronal del Actor en milisegundos, pero en un sistema real la cosa cambia. El cálculo del forward pass de una red densa, junto con los controladores PID de bajo nivel y los algoritmos de estimación de estado, requiere bastante potencia de cómputo. Integrar un ordenador a bordo capaz de ejecutar todo esto en tiempo real, respetando además las limitaciones de peso y consumo energético del cuadrotor, supone un reto tecnológico importante.

7.3 Líneas de trabajo futuro

Las limitaciones detectadas no bloquean la metodología, sino que más bien marcan el camino lógico a seguir en esta investigación. Convertir este modelo inicial en un sistema que pueda usarse en entornos reales de laboratorio va a requerir trabajar en varias líneas clave de desarrollo:

7.3.1 Validación sim-to-real mediante aleatorización del dominio

Para empezar, uno de los pasos más importantes es reducir la brecha entre simulación y realidad haciendo la red neuronal más robusta frente a la incertidumbre. En lugar de entrenar con parámetros totalmente exactos (masas, inercias o fricciones fijas), la idea es modificar el entorno de Gymnasium para que, al inicio de cada episodio, estos valores varíen aleatoriamente dentro de ciertos rangos (por ejemplo, ±15% en la masa del dron o ±20% en la fricción de los ganchos). Así, el agente aprendería una política de control mucho más generalizable, capaz de tolerar errores de modelado o perturbaciones reales sin perder estabilidad.

7.3.2 Transición hacia políticas visuomotoras con sensores exteroceptivos

Por otro lado, ahora mismo el agente recibe directamente las coordenadas exactas de la caja (es decir, estados perfectos). Para que el dron sea realmente autónomo, habría que eliminar esa "información privilegiada" y obligarlo a percibir el entorno. Esto implicaría sustituir esos datos por información procedente de sensores como cámaras RGB-D (color + profundidad) o sistemas LiDAR montados en el propio dron.

Este cambio también exigiría evolucionar la arquitectura de la red neuronal. En lugar de un perceptrón multicapa clásico, el sistema necesitaría incorporar redes convolucionales (CNN) o incluso modelos tipo Vision Transformer (ViT) para extraer información espacial directamente de las imágenes en tiempo real, y a partir de ahí alimentar la política de control.

7.3.3 Rediseño del sistema de efectores finales (Soft Robotics)

Desde el punto de vista mecánico, la dificultad para asegurar la retención de la carga en 0g (y la necesidad de meter una restricción virtual en el código) deja ver que unos ganchos rígidos en forma de L probablemente no son la mejor opción para manipulación en flotación libre. Como línea futura, se plantea sustituirlos por pinzas subactuadas o soluciones de robótica blanda (soft robotics). Estos sistemas pueden deformarse de forma pasiva y adaptarse mejor a la forma del asa, lo que reduce mucho la necesidad de una precisión geométrica extrema por parte de la red neuronal. Además, ayudan a absorber pequeños impactos y hacen que el agarre sea bastante más estable en condiciones donde domina la inercia.

REFERENCIAS

[1] F. L. V. &. O. A. Ruggiero, «Aerial Manipulation: A Literature Review,» *IEEE Robotics and Automation Letters,* nº 3, 2018.

[2] M. K. C. B. S. &. O. P. Orsag, «Dexterous aerial manipulation—Mobilizing topographies of 3D space,» *Unmanned Systems,* nº 5, 2017.

[3] A. &. H. G. Ollero, «Aerial robotics: Past, present and future,» nº 48, 2019.

[4] J. W. F. D. P. R. A. &. K. O. Schulman, «Proximal policy optimization algorithms,» *arXiv preprint,* 2017.