CEA Comité Español de Automática
XLVII JORNADAS DE AUTOMÁTICA CÓRDOBA 2026
UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Navegación autónoma en invernaderos mediterráneos mediante aprendizaje por refuerzo en ROS 2

Fernando Cañadas-Aránega[a,*], Juan D. Gil[a], José C. Moreno[a]

[a] *Universidad de Almería, Centro Mixto CIESOL, ceia3, Ctra.Sacramento s/n, 04120, Almería,España*

To cite this article: Cañadas-Aránega, F., Gil, J.D., Moreno, J.C. 2026. Autonomous Navigation in Mediterranean Greenhouses Using ROS 2-based Reinforcement Learning. Jornadas de Automática, 47. https://doi.org/10.17979/ja-cea.2026.47.13719

Resumen

La automatización en la agricultura protegida es clave para optimizar la productividad. Sin embargo, la navegación autónoma de robots en invernaderos mediterráneos plantea severos desafíos debido a restricciones geométricas, pasillos estrechos e invasión de la masa foliar, donde los algoritmos tradicionales resultan insuficientes. Para abordar este problema, este trabajo propone un sistema de navegación autónoma basado en ROS 2 que integra un agente de aprendizaje por refuerzo basado en el algoritmo PPO — por sus siglas en inglés, *Proximal Policy Optimization* — en el robot móvil diferencial AgriCobIoT I. La principal aportación radica en el diseño y adaptación de una función de recompensa específicamente modelada para mitigar las severas condiciones del entorno agrícola y guiar el aprendizaje en escenarios restrictivos. Los resultados demuestran que la estrategia propuesta permite el entrenamiento y validación de agentes de aprendizaje por refuerzo antes de su transferencia al entorno real.

Palabras clave: Robótica en Agricultura, Robótica Móvil, Aprendizaje automático, PPO, Inteligencia Artificial

Autonomous Navigation in Mediterranean Greenhouses Using ROS 2-based Reinforcement Learning

Abstract

Automation in protected agriculture is key for optimising productivity. However, the autonomous navigation of robots in Mediterranean greenhouses poses severe challenges due to geometric constraints, narrow aisles and dense foliage, where traditional algorithms prove insufficient. To address this problem, this work proposes an autonomous navigation system based on ROS 2 that integrates a reinforcement learning agent based on the Proximal Policy Optimization (PPO) algorithm on the AgriCobIoT I differential mobile robot. The main contribution lies in the design and adaptation of a reward function specifically modelled to mitigate the severe conditions of the agricultural environment and guide learning in restrictive scenarios. The results demonstrate that the proposed strategy enables the training and validation of reinforcement learning agents before their transfer to the real environment.

Keywords: Robotics in Agriculture, Mobile Robotics, Reinforcement Learning, PPO, Artificial Intelligence

1. Introducción

Durante las últimas décadas, la agricultura protegida, particularmente en la región mediterránea, se ha consolidado como una estrategia fundamental para maximizar la productividad alimentaria bajo estrictos criterios de optimización de recursos hídricos y energéticos (Kim et al., 2025). Para mantener la competitividad de estos sistemas de producción intensiva, el sector está experimentando una transición hacia la automatización integral, delegando tareas repetitivas y de alta demanda logística en plataformas robóticas móviles (Hernández et al., 2025). Sin embargo, la introducción de robots móviles en estos escenarios plantea desafíos operativos sustanciales debido a las condiciones estructurales intrínsecas de las explotacio-

*Autor para correspondencia: fernando.ca@ual.es

nes agrícolas, las cuales difieren drásticamente de los entornos industriales controlados y diáfanos (Cañadas-Aránega et al., 2026c). El principal obstáculo para la navegación autónoma radica en la naturaleza geométrica y dinámica del propio invernadero mediterráneo. Con el fin de maximizar la superficie de cultivo útil, estas instalaciones se diseñan con pasillos de circulación extremadamente estrechos que reducen al mínimo el margen de maniobra física de los vehículos (Sánchez-Molina et al., 2024). A esta severa restricción espacial se suma un entorno altamente cambiante e incierto: la masa foliar de las plantas crece continuamente, invadiendo las vías de tránsito y alterando las lecturas de los sensores, mientras que los operarios humanos interactúan constantemente en las mismas líneas de trabajo como obstáculos dinámicos. Bajo estas condiciones, los algoritmos de navegación tradicionales basados en mapas estáticos o reglas heurísticas rígidas resultan insuficientes (Cañadas-Aránega et al., 2026b), al ser incapaces de adaptarse a las perturbaciones del entorno en tiempo real.

Para abordar esta falta de flexibilidad, el uso del Aprendizaje por Refuerzo (RL, por sus siglas en inglés *Reinforcement Learning*) en la robótica agrícola ha evolucionado significativamente en los últimos años, transitando desde enfoques discretos basados en redes Q profundas (DQN, por su siglas en inglés *Deep Q-Networks*) para tareas de recolección y guiado básico hacia algoritmos de gradiente de política capaces de gestionar los espacios de acción continuos que exigen las plataformas reales (Lai et al., 2025; Wang et al., 2025). Entre estos últimos, arquitecturas deterministas como *Deep Deterministic Policy Gradient* (DDPG) y estocásticas como el *Soft Actor-Critic* (SAC) han sido ampliamente adoptadas para tareas de control automático (Gil et al., 2026). Sin embargo, aunque estos métodos ofrecen una elevada eficiencia muestral, suelen presentar inestabilidades durante el entrenamiento y requerir una compleja sintonización de hiperparámetros en entornos geométricamente confinados. En este contexto, algoritmos basados en gradientes de política más estables, como *Proximal Policy Optimization* (PPO), han ganado interés en aplicaciones de agricultura de precisión (Schulman et al., 2017). PPO destaca gracias a su mecanismo de recorte de su función objetivo, el cual restringe el tamaño de las actualizaciones de la política y contribuye a evitar degradaciones bruscas en el rendimiento del agente durante el entrenamiento. Esta característica, sumada a su robustez matemática y estabilidad ante la incertidumbre del entorno, convierte a PPO en una solución adecuada para garantizar la convergencia segura en maniobras críticas de alta precisión, tales como el viraje completo de 180 grados en las estrechas cabeceras de los pasillos en invernaderos mediterráneos (Gao et al., 2026).

En este artículo se propone e implementa un sistema de navegación autónoma basado en ROS 2 (Macenski et al., 2022), en el que se emplea el algoritmo PPO para aprender la política de control del robot móvil AgriCobIoT I (AGI) en entornos agrícolas complejos, como los invernaderos mediterráneos. La principal aportación del artículo radica en el diseño y adaptación de una función de recompensa específicamente concebida para mitigar las severas condiciones del entorno agrícola y guiar el aprendizaje en escenarios altamente restrictivos. Para ello, el agente se entrena dentro de un modelo de simulación 2D construido a partir de datos reales obtenidos en Cañadas-Aránega et al. (2026a). Dicho entorno reproduce con precisión las restricciones dimensionales del invernadero y la ocupación real del follaje, permitiendo que el robot AGI aprenda de forma segura a recorrer pasillos estrechos y ejecutar la maniobra crítica de giro en cabecera sin colisionar con las plantas ni con los elementos estructurales. De este modo, se establece una base metodológica sólida que minimiza el riesgo antes de la transferencia del agente al entorno real.

El resto del artículo se estructura de la siguiente manera: La sección 2 presenta la descripción del modelo del sistema y el entorno de entrenamiento. La configuración adaptada a este problema se presenta en la sección 3. La sección 4 analiza los experimentos realizados. Por último, la sección 5 se dedica a presentar algunas conclusiones.

2. Modelado del sistema y configuración del entorno

En esta sección se presenta el modelo de robot móvil utilizado y el entorno de simulación para realizar el RL.

2.1. *Modelo cinemático del AgriCobIoT I (AGI)*

Para realizar la simulación se utiliza el robot móvil AgriCobIoT I (AGI) (López-Gázquez et al., 2023), basado en la plataforma Husky de Clearpath™, un vehículo terrestre diseñado para operar en terrenos irregulares. En la ecuación (1) se presenta el modelo cinemático del AGI con (x, y) como las coordenadas de posición y θ denotando su orientación.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{1}$$

donde v representa la velocidad lineal instantánea del robot y ω es su velocidad angular. El espacio de acciones del algoritmo PPO modificará directamente este vector de control $\mathbf{u} = [v, \omega]^T$ en cada paso de tiempo de la simulación, estando ambos comandos acotados por los límites físicos y dinámicos de la plataforma para asegurar trayectorias suaves.

2.2. *Entorno de simulación del invernadero*

El entorno de aprendizaje se despliega sobre un plano 2D simulado en Rviz2 (Macenski et al., 2022), ver Fig. 1, cuyas dimensiones geométricas y disposición espacial han sido obtenidas a partir de datos empíricos de un invernadero mediterráneo real (Cañadas-Aránega et al., 2024). El mapa resultante define dos hileras de cultivos dispuestas en pasillos paralelos simétricos de 12.5 metros de longitud y 1.5 metros de anchura, conectados entre sí por una zona habilitada para el giro al inicio y al final de los pasillos.

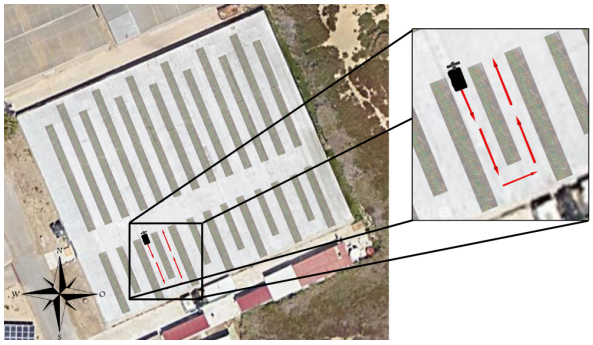


Figura 1: Plano 2D simulado del invernadero real

3. Configuración del agente de RL basado en PPO

Para aplicar PPO al problema de navegación autónoma, el sistema debe formularse previamente como un Proceso de Decisión de Markov (MDP, por sus siglas en inglés, *Markov Decision Process*). Esta formulación permite definir de manera estructurada los elementos fundamentales del RL: el espacio de estados, el espacio de acciones y la función de recompensa. En esta sección se detalla dicha formulación y se describe cómo estos elementos se integran en el entorno de simulación empleado para el entrenamiento del agente.

3.1. Formulación del MDP

El guiado autónomo del robot AGI dentro del invernadero mediterráneo se modela formalmente como un MDP definido por la tupla $\langle \mathcal{S}, \mathcal{A}, R, P, \gamma \rangle$, donde $P(\mathcal{S}_{t+1} \mid \mathcal{S}_t, \mathcal{A}_t)$ representa la función de transición de estado que rige la dinámica estocástica del entorno y γ es el parámetro de descuento de la MDP (ver Fig. 2).

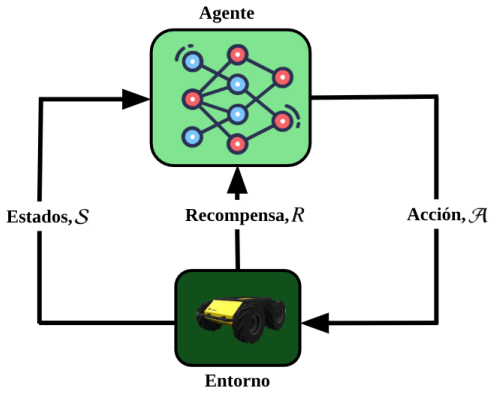


Figura 2: Marco de aprendizaje por refuerzo

3.1.1. Espacio de Estados

Para cada instante de tiempo discreto t, el vector de estados $s_t \in \mathbb{R}^{12}$ combina información perceptiva y de navegación:

$$s_t = [\, \ell_1, \ldots, \ell_9, \tilde{d}_t, \sin\alpha_t, \cos\alpha_t \,], \tag{2}$$

donde $\ell_k \in [0, d_{\max}]$ es la distancia mínima del LiDAR en el k-ésimo sector angular (9 sectores de 40° cada uno, $d_{\max}$=5 m); $\tilde{d}_t$ es la distancia euclídea normalizada entre el estado del robot y el objetivo; y $(\sin\alpha_t, \cos\alpha_t)$ codifica el ángulo relativo al objetivo $\alpha_t = \text{atan2}(\sin(\phi_t - \theta_t), \cos(\phi_t - \theta_t))$ (siendo $\phi_t -$ la orientación hacia el objetivo) evitando la discontinuidad en $\pm\pi$.

3.1.2. Espacio de Acciones Discretas

El espacio de acciones discretas se define mediante una configuración discreta de pares (v, ω), diseñada para representar los tres comportamientos básicos requeridos por el robot en este escenario, y descritos más adelante. La selección de dichas acciones no es arbitraria, sino que responde directamente a las restricciones geométricas y maniobras viables dentro del entorno agrícola. Así, para un instante de tiempo discreto t, el conjunto de acciones $a_t \in \mathbb{R}^2$ posibles viene dado por:

$$a_t \in \{(v_{\max}, 0),\ (0, \pm\omega_{\max}),\ (v_{\max}, \pm 0.4\,\omega_{\max})\}, \tag{3}$$

con $v_{\max}$=0.2 m/s y $\omega_{\max}$=$\pi/4$ rad/s, y donde cada acción se ejecuta durante un ciclo LiDAR ($\Delta t \approx 100$ ms). La definición de las acciones son:

- *Acción* $(v_{\max}, 0)$: representa el robot cuando avanza recto por el pasillo (acción que debe realizar la mayor parte del tiempo).
- *Acción* $(0, \pm\omega_{\max})$: representa el giro cerrado con avance nulo, diseñado específicamente para negociar el cambio de pasillo sin colisionar.
- *Acción* $(v_{\max}, \pm 0.4\,\omega_{\max})$: representa la corrección suave de rumbo manteniendo velocidad máxima, para ajustar la trayectoria dentro del pasillo recto sin perder velocidad.

3.1.3. Función de Recompensa

La trayectoria del cambio de pasillo impone que la distancia euclídea al objetivo no sea un indicador fiable de progreso: un episodio en el que el robot retrocede sobre el pasillo ya recorrido sin cambiar al siguiente puede parecer más cercano a la meta que uno en el que el robot acaba de superar la curva. Para resolver esta ambigüedad se introduce la variable binaria $c_t \in \{0, 1\}$, que indica si ha pasado la curva. Una vez activado, c_t permanece a 1 el resto del episodio. La función de recompensa adopta la estructura:

$$R_t = \begin{cases} R_{\text{col}} & \text{colisión}, \\ R_{\text{goal}} & d_t < d_{\text{end}}, \\ R_{\text{time}} & n_t \geq N_{\max}, \\ R_{\text{step}} & \text{en caso contrario}, \end{cases} \tag{4}$$

donde d_{end}=0.8 m representa el valor del diámetro válido previo a llegar al objetivo, n_t representa el numero de pasos ejecutados hasta el instante t y $N_{\max}$=1300 pasos máximos permitidos, con:

$$R_{\text{col}} = -10, \quad R_{\text{goal}} = 300 + 0.2\,(N_{\max} - n_t), \quad R_{\text{time}} = -10. \tag{5}$$

Los valores han sido escogidos por prueba y error buscando la optimización de la convergencia del PPO. En pasos no terminales la recompensa se descompone en cuatro términos:

$$R_{\text{step}} = \underbrace{R_{\text{base}}}_{\text{avance}} + \underbrace{R_{\text{rot}}}_{\text{giro excesivo}} + \underbrace{R_{\text{plants}}}_{\text{plantas}} + \underbrace{R_{\text{nav}}}_{\text{navegación}}, \tag{6}$$

donde el término de avance recompensa el movimiento lineal:

$$R_{\text{base}} = \begin{cases} +0.5 & v > 0, \\ -0.3 & v = 0; \end{cases} \tag{7}$$

la penalización por giros consecutivos n_{rot} con $v = 0$ m/s (giros sobre si mismo) desincentiva las oscilaciones en el mismo punto:

$$R_{\text{rot}} = \begin{cases} -0.3\,n_{\text{rot}} & v = 0,\ n_{\text{rot}} > 3, \\ 0 & \text{en caso contrario}; \end{cases} \tag{8}$$

donde se permiten 3 giros para los casos en los que el robot necesite orientarse. La distancia a las plantas proporciona un gradiente antes de la colisión:

$$R_{\text{plants}} = \begin{cases} -5.0\left(1 - \dfrac{\ell_{\min}}{0.7}\right) & \ell_{\min} < 0.7\ \text{m}, \\ 0 & \text{en caso contrario}; \end{cases} \tag{9}$$

y el término de navegación determina la lógica del *waypoint* de curva:

$$R_{\text{nav}} = \begin{cases} +50.0 & c_{t-1}{=}0,\ \|(x_t, y_t){-}\mathbf{p}_c\| < r_c \text{ (curva superada)}, \\ -20.0 & c_t{=}1,\ \beta_t{=}1,\ \|(x_t, y_t){-}\mathbf{p}_c\| < r_c \text{ (retroceso)}, \\ 5.0\,\Delta d_t & c_t{=}1 \text{ (progreso al objetivo)}, \\ 0 & c_t{=}0 \text{ (antes de la curva)}, \end{cases} \tag{10}$$

donde $\mathbf{p}_c$ representa un punto de paso en medio de la curva al siguiente pasillo, (x_t, y_t) representa la posición del robot en el instante t, r_c =1.5 m es el radio de detección de la curva, $\Delta d_t = d_t - d_{t+1}$ el acercamiento al objetivo en el paso t y β_t indica que el robot se alejó de la curva hacia el objetivo tras superarla. El bonus de +50 se otorga una única vez; la penalización de retroceso −20 se aplica cuando el robot regresa al radio de la curva tras haberlo abandonado. La Tabla 1 resume todos los términos.

Tabla 1: Resumen de la función de recompensa.

Condición	**Valor**
Colisión	-10
Éxito ($d_t < 0.8$ m)	$300 + 0.2\,(N_{\max} - n_t)$
Timeout ($n_t \geq 1300$)	-10
Avance ($v > 0$)	$+0.5$
Sin avance ($v = 0$)	-0.3
Giro excesivo ($n_{\text{rot}} > 3$)	$-0.3\,n_{\text{rot}}$
Proximidad obstáculo ($\ell_{\min} < 0.7$ m)	$-5.0\,(1 - \ell_{\min}/0.7)$
Primera vez en curva	$+50.0$
Retroceso a curva	-20.0
Progreso al objetivo (c_t=1)	$+5.0\,\Delta d_t$

3.2. *Agente PPO*

El agente PPO empleado en este trabajo se basa en una arquitectura actor-crítico (Schulman et al., 2017), donde la red actor aproxima la política estocástica mediante la función $\pi_\theta(a_t|s_t)$, parametrizada por los pesos θ, mientras que la red *critic* estima la función de valor del estado $V_\phi(s_t)$ definida por los parámetros ϕ.

La entrada de ambas redes corresponde al estado definido en el MDP, compuesto por las observaciones del entorno percibidas por el robot AGI. Ambas redes comparten un extractor de características basado en un perceptrón multicapa con dos capas ocultas de 256 neuronas y funciones de activación *Rectified Linear Unit* (ReLU) (Cañadas-Aranega et al., 2026). La salida de la red actor consiste en una distribución categórica sobre el conjunto discreto de acciones definido en el MDP, mientras que la red critic produce una estimación escalar del valor esperado del estado.

3.3. *Configuración del entrenamiento del agente PPO*

PPO maximiza una función objetivo con recorte (*clipping*) (Schulman et al., 2017):

$$\mathcal{L}^{\text{CLIP}} = \mathbb{E}_t\Big[\min\Big(\rho_t \hat{A}_t,\ \text{clip}(\rho_t, 1{\pm}\epsilon)\,\hat{A}_t\Big)\Big], \tag{11}$$

donde $\rho_t = \pi_\theta(a_t|s_t)/\pi_{\theta_{\text{old}}}(a_t|s_t)$ representa la razón entre la política actual y la política previa. La ventaja $\hat{A}_t$ se estima mediante *Generalized Advantage Estimation* (GAE) (Schulman et al., 2015) como $\hat{A}_t = \sum_{k=0}^{\infty}(\gamma\lambda)^k\delta_{t+k}$, con γ como factor de descuento del MDP, λ como parámetro de suavizado de GAE y $\delta_t = R_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$ representa el error de diferencia temporal (siendo V_ϕ la estimación actual que hace la red crítica sobre cuánta recompensa a largo plazo va a conseguir el robot desde el estado actual y V^{goal} es el objetivo real (la recompensa verdadera que el robot acabó acumulando en el episodio utilizando la ecuación GAE donde opera γ)).

La función de pérdida total utilizada durante el entrenamiento se define como:

$$\mathcal{L} = -\mathcal{L}^{\text{CLIP}} + 0{,}5\,\mathbb{E}_t[(V_\phi - V^{\text{goal}})^2] - 0{,}05\,\mathcal{H}[\pi_\theta], \tag{12}$$

donde $\mathcal{H}[\pi_\theta]$ corresponde al término de entropía de la política, utilizado para favorecer la exploración durante el aprendizaje.

El entrenamiento se realiza de forma iterativa utilizando bloques de 51200 pasos (escalados según las demandas cinemáticas de los *rollouts* de T = 2048 pasos) hasta alcanzar una tasa de éxito superior al 80 % en 10 episodios consecutivos de evaluación. La Tabla 2 resume los hiperparámetros empleados.

Tabla 2: Hiperparámetros del agente PPO (resto de parámetros necesarios en Almeida et al. (2023))

Parámetro	**Símbolo**	**Valor**
Factor de descuento	γ	0.99
Parámetro de suavizado GAE	λ	0.95
Radio de recorte	ϵ	0.20
Tasa de aprendizaje	α	3×10^{-4}
Pasos por *rollout*	T	2 048
Pasos máx. episodio	$N_{\max}$	1 300

4. Resultados

El agente PPO se configuró en un entorno simulado cuyos obstáculos y trayectorias se derivaron de los datos recolectados en (Cañadas-Aránega et al., 2024). A partir de dicha información, se implementó un mapa SLAM 2D (ver Fig. 3) en Rviz2 (ejecutado a un factor de aceleración de 10× respecto al tiempo real) y se definieron tanto el modelo del robot como su cinemática de control diferencial. La fase de entrenamiento se llevó a cabo en un ordenador equipado con una GPU NVIDIA RTX 4060 (8 GB), una CPU Intel Core i7-13400 y 32 GB de memoria RAM DDR4 de Kingston (3200 MHz), tomando 45 minutos en total.

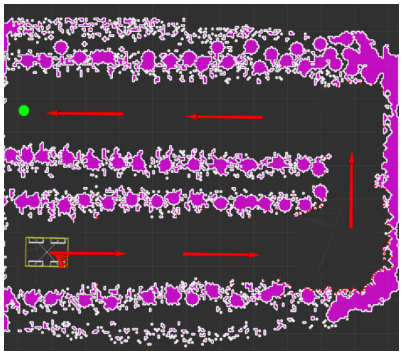


Figura 3: SLAM 2D de la zona de trabajo en el invernadero

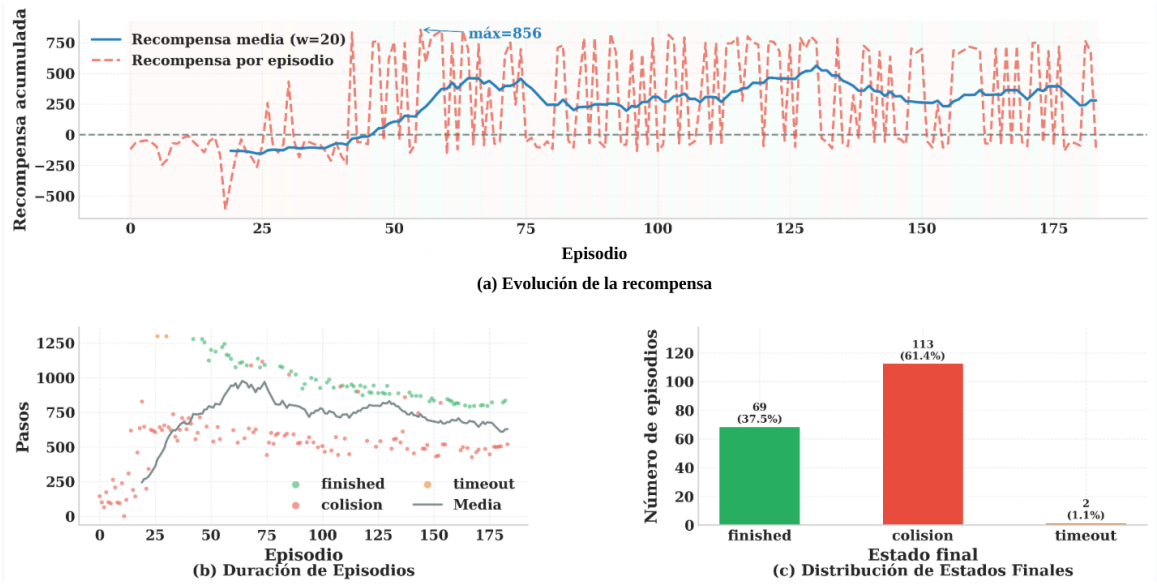


Figura 4: Rendimiento del entrenamiento

El entrenamiento requiere que el vehículo comience su marcha en un pasillo, ejecute una maniobra de giro al llegar al extremo y retorne por el pasillo contiguo. Dado que esta secuencia lineal fija induciría un sesgo en el agente hacia giros exclusivos a la izquierda, se introdujo una estrategia de inversión estocástica de la trayectoria. De este modo, cada vez que un episodio termina, el robot comienza en un punto aleatorio y ejecuta el recorrido con distintos giros, lo que garantiza un aprendizaje completo. En esta sección se presentan y analizan los resultados obtenidos tras el proceso de optimización.

4.1. *Rendimiento del entrenamiento*

El agente PPO se entrenó durante tres iteraciones de 55392 (51200 + 4192 de 10 test), 58028 (51200 + 6828 de 10 test) y 26875 pasos respectivamente (140295 pasos en total), registrando 164 episodios de entrenamiento y 20 episodios de evaluación. La Fig. 4.a muestra la evolución de la recompensa acumulada por episodio. Durante la primera iteración, el agente registró una tasa de colisión del 73.3 % y una recompensa media de 11.55, con los primeros episodios exitosos apareciendo a partir de los 20000 pasos acumulados (episodio 42, 20866 pasos totales) (Fig. 4.b). En la tercera iteración, la tasa de colisión se redujo al 43.7 % y la recompensa media fue de 334.4, evidenciando que el agente consolidó la estrategia de superar el *waypoint* de curva antes de dirigirse al objetivo (Fig. 4.c). La Tabla 3 resume las métricas por iteración.

Tabla 3: Métricas de entrenamiento por iteración.

Iter.	Ep.	Colisión (%)	$\bar{R}$ (media)	$\bar{R}$ (max)
1	75	73.3	11.55	75.12
2	68	52.9	292.9	702.3
3	41	43.7	334.4	856.9

La reducción de la tasa de colisión entre la iteración 1 y la iteración 3 (73.3 % → 43.7 %) coincide con la estabilización de la recompensa media en episodios exitosos en torno a 334.4 puntos, lo que indica que la política aprendió a navegar de forma consistente por el pasillo una vez superada la curva. En este *enlace* se muestra como ejemplo un vídeo correspondiente al episodio número 75 del entrenamiento, donde el robot acaba en un estado *finished*.

4.2. *Análisis de navegación en el invernadero*

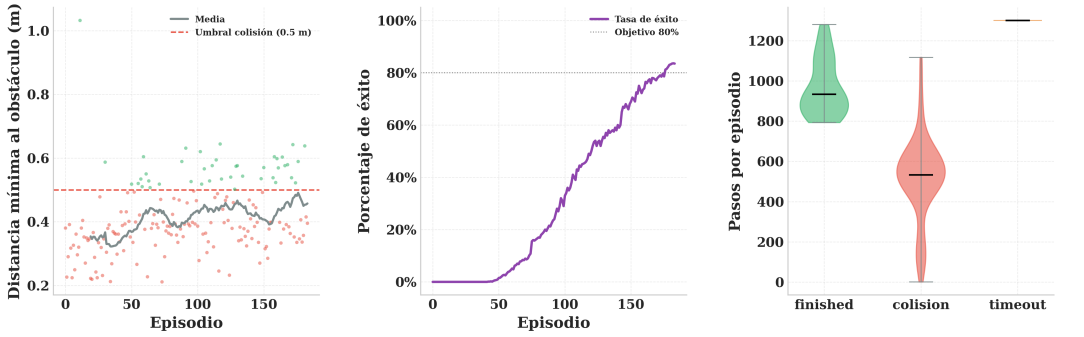
La distancia mínima media al obstáculo más cercano fue de 0.514 m en episodios exitosos y 0.411 m en el conjunto total de episodios de entrenamiento, ambas por encima del mínimo físico del AGI (0.335 m de semiancho). En este caso, la ecuación (9) fue efectiva para mantener al robot alejado de las plantas en los episodios que concluyeron con éxito: ningún episodio exitoso registró un valor medio de $\ell < 0.21$ m, por lo que se mantuvo, en la mayoría del trayecto, a una distancia segura de las plantas (Fig. 5.a).

4.2.1. *Tasa de éxito*

El algoritmo comenzó con una tasa de éxito baja ya que, en los primeros episodios, detectaba una colisión habiendo avanzado pocos pasos. En el episodio 50, AGI logró girar al siguiente pasillo, dotándole con una recompensa alta. En ese sentido, aprendió a pasar la curva crítica del invernadero. El robot logró alcanzar el objetivo bajo los criterios establecidos, donde más de 10 episodios lograron alcanzar una tasa de éxito mayor del 80 % en los 20 episodios anteriores, finalizando la simulación y obteniendo el modelo PPO entrenado (Fig. 5.b).

4.2.2. *Duración y suavidad de la trayectoria*

Los episodios exitosos requirieron una media de 966 pasos (≈ 97 s a Δt=100 ms), frente a los 675 pasos de media general, lo que indica que el agente siguió trayectorias largas pero completas. La duración media de los episodios exitosos mejoró progresivamente entre iteraciones: 1164 pasos en la primera iteración, 940 en la segunda iteración y 822 en la tercera iteración, evidenciando que la política fue encontrando rutas más cortas conforme avanzó el entrenamiento (Fig 5.c).



(a) Proximidad Mínima a Plantas (b) Tasa de Éxito (c) Distribución de Duración por Estado

Figura 5: Gráficas de navegación del PPO

5. Conclusiones y trabajo futuro

La implementación de una estrategia adaptada a entornos complejos, como son los invernaderos mediterráneos, resolvió eficazmente el riesgo de choque con el cultivo para navegar entre pasillos contiguos con ayuda del agente PPO. A su vez, el proceso de optimización logró disminuir la tasa de colisiones desde un 73.3 % inicial hasta un 43.7 % al cierre de la tercera iteración, obteniendo un indice de convergencia mayor del 80 % durante 10 episodios.

En términos de integridad operativa, el diseño de la función de recompensa basada en el paso de un *waypoint* en la curva demostró ser plenamente efectivo. En el 100 % de los episodios exitosos, los cuales representaron el 47.05 % del total global, el vehículo se mantuvo por encima del umbral de colisión física, promediando una distancia de seguridad de 0.514 m frente a los límites estructurales de las plantas. Esta evolución se tradujo además en una optimización temporal de la ruta, ya que a lo largo de los 140,295 pasos totales, la política localizó trayectorias más cortas y directas que redujeron la duración media de navegación de 1164 a solo 822 pasos por episodio. Finalmente, el uso de un factor de aceleración de 10× permitió alcanzar la convergencia en 45 minutos. Por último, el uso de datos previos permitió un entrenamiento offline acelerado a 10×, alcanzando la convergencia en 45 minutos y agilizando la prueba futura de nuevos agentes y arquitecturas.

Como trabajo futuro, se contempla la transferencia de la política optimizada al robot AGI real para evaluar la robustez del algoritmo frente al deslizamiento del terreno, las variaciones lumínicas sobre el LiDAR y la flexibilidad de la masa vegetal, entre otras.

Agradecimientos

Este trabajo ha sido financiado por el Proyecto LIFE ACCLIMATE, cofinanciado por la Unión Europea bajo el acuerdo de concesión del programa LIFE No. LIFE23-CCA-ES-LIFE-ACCLIMATE/101157315. Además, el autor Fernando Cañadas-Aránega es beneficiario de una beca FPI (PRE2022-102415) del Ministerio de Ciencia, Innovación y Universidades de España.

Referencias

Almeida, F., Leão, G., Sousa, A., 2023. An educational kit for simulated robot learning in ros 2. In: Iberian Robotics conference. Springer, pp. 513–525.

Cañadas-Aranega, F., Gil, J. D., Moreno, J. C., Blanco-Claraco, J. L., 2026. Marco metodológico para la navegación de robots en invernaderos mediante aprendizaje por refuerzo. Simposios del Comité Español de Automática (CEA) 2 (2).

Cañadas-Aránega, F., Mañas-Álvarez, F. J., Moreno, J. C., Blanco-Claraco, J. L., et al., 2026a. A ros2 benchmarking framework for hierarchical control strategies in mobile robots for mediterranean greenhouses. arXiv preprint arXiv:2602.15162.

Cañadas-Aránega, F., Moreno, J. C., Blanco-Claraco, J. L., 2026b. Greenseg: Ground segmentation algorithm for agricultural robots in mediterranean greenhouses using rgb-d point clouds. arXiv preprint arXiv:2605.25279.

Cañadas-Aránega, F., Wollherr, D., Guzmán, J. L., Moreno, J. C., Blanco, J. L., August 22-28 2026c. An adaptive control architecture for slope and terrain compensation in autonomous navigation in mediterranean greenhouses. In: Proceedings of the 23rd IFAC World Congress. Busan, South Korea, accepted for publication.

Cañadas-Aránega, F., Blanco-Claraco, J. L., Moreno, J. C., Rodriguez-Diaz, F., 2024. Multimodal mobile robotic dataset for a typical mediterranean greenhouse: The greenbot dataset. Sensors 24 (6).

Gao, Y., Jiang, Q., Wang, M., Dong, X., 2026. Advances in path-planning algorithms for agricultural robots. Journal of Field Robotics 43 (1), 89–145.

Gil, J. D., Chanona, E. A. D. R., Guzmán, J. L., Berenguel, M., 2026. Reinforcement learning meets bioprocess control through behavior cloning: Real-world deployment in an industrial photobioreactor. Engineering Applications of Artificial Intelligence 164, 113326.

Hernández, H. A., Mondragón, I. F., González, S. R., Pedraza, L. F., 2025. Reconfigurable agricultural robotics: Control strategies, communication, and applications. Computers and Electronics in Agriculture 234, 110161.

Kim, C. H., Silwal, A., Kantor, G., 2025. Autonomous robotic pepper harvesting: Imitation learning in unstructured agricultural environments. IEEE Robotics and Automation Letters.

Lai, M., Go, K., Li, Z., Kröger, T., Schaal, S., Allen, K., Scholz, J., 2025. Roboballet: Planning for multirobot reaching with graph neural networks and reinforcement learning. Science Robotics 10 (106), eads1204.

López-Gázquez, A., Mañas-Álvarez, F. J., Moreno, J. C., Cañadas-Aránega, F., Sánchez, J. A., October 22–27 2023. Navigation of a differential robot for transporting tasks in mediterranean greenhouses. In: Proceedings of the 2023 International Symposium on New Technologies for Sustainable Greenhouse Systems (Greensys). ISHS (International Society for Horticultural Science), Cancún, Mexico, pp. 1–8.

Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W., 2022. Robot operating system 2: Design, architecture, and uses in the wild. Science Robotics 7 (66), eabm6074.

Sánchez-Molina, J. A., Rodríguez, F., Moreno, J. C., Sánchez-Hermosilla, J., Giménez, A., 2024. Robotics in greenhouses. scoping review. Computers and Electronics in Agriculture 219, 108750.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P., 2015. Trust region policy optimization. In: International conference on machine learning. PMLR, pp. 1889–1897.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Wang, P., He, P., Ma, C., Niu, C., Gao, H., Wang, H., Muyeen, S., Zhou, D., 2025. A novel path planning approach for plant protection uav based on ddpg and ila optimization algorithm. Computers and Electronics in Agriculture 239, 111006.