



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

Título del Proyecto

MEMORIA PRESENTADA POR:

Carlos Mira López

Nicolás Miró Mira

Vittorio Alessandro Esposito Ceballos

MODELADO Y CONTROL DE ROBOTS

GRADO EN INFORMÁTICA INDUSTRIAL Y ROBÓTICA

Curso: 2025/2026

Resumen

Índice general

Resumen	I
Índice general	II
Índice de figuras	IV
Índice de tablas	V
1. Introducción, objetivos y estructura del documento	1
1.1. Introducción	1
1.2. Objetivo general y objetivos específicos	2
1.3. Estructura del documento	2
2. Antecedentes y estado del arte	4
2.1. Fundamentos teóricos relevantes	4
2.2. Estado del arte en robótica y control	4
2.3. Herramientas y entornos de simulación	5
2.4. Trabajos y resultados previos	5
2.5. Relación con el proyecto	5
3. Materiales y métodos	6
3.1. Modelado cinemático/dinámico del robot + tool (si aplica)	6
3.2. Metodología de control y aprendizaje por refuerzo (RL)	6
3.3. Entorno de simulación y herramientas utilizadas	8
3.4. Procedimiento de experimentación / entrenamiento	9
3.5. Programación	10
4. Resultados	11
4.1. Validación del modelado matemático	11
4.2. Entrenamientos en IsaacLab/IsaacSim	11
4.3. Comparación de controladores	11
4.4. Pruebas finales y validación del sistema	12
4.5. Discusión crítica	12
5. Conclusiones y trabajo futuro	13

5.1. Conclusiones	13
5.2. Trabajo futuro	13

Índice de figuras

Índice de tablas

1 Introducción, objetivos y estructura del documento

El presente proyecto aborda el desafío de la navegación autónoma en robótica móvil mediante el uso de percepción visual. Se centra en el modelado y control del robot TurtleBot3 dentro de un entorno de simulación de alta fidelidad (IsaacSim/IsaacLab). El objetivo fundamental es desarrollar estrategias de control que permitan al robot alcanzar objetivos definidos utilizando cámaras RGB/RGB-D, explorando tanto algoritmos de navegación clásicos como técnicas avanzadas de Aprendizaje por Refuerzo (Reinforcement Learning), evaluando su desempeño en escenarios controlados.

1.1 Introducción

La robótica móvil ha experimentado una transformación radical en la última década, pasando de ser una curiosidad académica a convertirse en un pilar fundamental de la Industria 4.0 y la logística moderna. Los robots móviles autónomos (AMR) son hoy esenciales en tareas de transporte de materiales, inspección de infraestructuras y robótica de servicios. Sin embargo, para que estos sistemas sean verdaderamente autónomos, deben poseer la capacidad de percibir su entorno, interpretarlo y tomar decisiones de navegación seguras y eficientes en tiempo real.

Tradicionalmente, la navegación se ha resuelto mediante mapas estáticos y sensores láser (LiDAR). No obstante, la tendencia actual se dirige hacia el uso de percepción visual (cámaras RGB y de profundidad) y técnicas de inteligencia artificial, que permiten obtener una información semántica del entorno mucho mejor y a un menor coste de hardware.

Este proyecto surge de la necesidad de comprender y dominar las técnicas modernas de control robótico. Específicamente, se justifica en la transición tecnológica actual que busca sustituir o complementar la programación clásica con algoritmos de aprendizaje. El uso de herramientas de simulación fotorrealista y física precisa, como NVIDIA IsaacSim e IsaacLab, permite entrenar y validar estos sistemas en entornos virtuales seguros antes de su despliegue físico, reduciendo riesgos y costes (concepto conocido como *Sim-to-Real*).

Desde una perspectiva docente y académica, este trabajo permite la adquisición de competencias transversales en modelado cinemático, visión por computador, integración de sensores y, fundamentalmente, en la aplicación de Deep Reinforcement Learning (Deep RL), una técnica que está redefiniendo el estado del arte en el control de robots dinámicos.

1.2 Objetivo general y objetivos específicos

El **objetivo principal** de este proyecto es diseñar, implementar y validar un sistema de control para la navegación autónoma del robot móvil TurtleBot3 en un entorno simulado, capaz de localizar y alcanzar objetivos visuales utilizando información sensorial proveniente de cámaras a bordo y algoritmos de inteligencia artificial.

Para alcanzar esta meta, se han definido los siguientes objetivos específicos:

1. **Modelado y configuración del entorno de simulación:** Importar y configurar el modelo cinemático y visual del TurtleBot3 en el entorno IsaacSim/IsaacLab, integrando sensores virtuales de visión (cámara RGB/RGB-D) y configurando escenarios con metas visuales específicas.
2. **Desarrollo del sistema de percepción y navegación:** Implementar los algoritmos necesarios para procesar la información visual del robot y traducirla en comandos de velocidad (lineal y angular) que permitan la navegación hacia el objetivo.
3. **Implementación de agentes de Aprendizaje por Refuerzo (RL):** Diseñar y entrenar un agente basado en RL (explorando arquitecturas con redes neuronales convolucionales - CNNs) capaz de aprender la política de navegación óptima mediante la interacción prueba-error con el entorno simulado.
4. **Validación y análisis comparativo:** Evaluar el rendimiento del sistema en términos de tasa de éxito, tiempo de llegada y suavidad de la trayectoria, comparando la robustez de las soluciones frente a cambios en la posición del objetivo o del entorno.
5. **Documentación y análisis crítico:** Analizar la viabilidad de la transferencia de estos algoritmos a robots de servicio reales y documentar el proceso técnico para futuras investigaciones.

1.3 Estructura del documento

La presente memoria se organiza en cinco capítulos que detallan el desarrollo del proyecto desde su concepción teórica hasta la validación de resultados:

- **Capítulo 1: Introducción, objetivos y estructura.** Introduce el contexto del problema, justifica la elección del TurtleBot3 y las herramientas de simulación, y define las metas del proyecto.

- **Capítulo 2: Antecedentes y estado del arte.** Revisa los fundamentos teóricos de la robótica móvil diferencial, la percepción visual y el Aprendizaje por Refuerzo. Se analizan trabajos previos y se describen las herramientas utilizadas (ROS2, IsaacSim, PyTorch).
- **Capítulo 3: Materiales y métodos.** Detalla la metodología empleada, incluyendo el modelado matemático del robot, la configuración de los sensores en simulación, la arquitectura de las redes neuronales utilizadas y el diseño de la función de recompensa para el entrenamiento del agente.
- **Capítulo 4: Resultados.** Presenta las gráficas de entrenamiento, las métricas de desempeño obtenidas en las pruebas de navegación y una comparativa de los distintos controladores implementados. Se valida el cumplimiento de los objetivos.
- **Capítulo 5: Conclusiones y trabajo futuro.** Sintetiza los hallazgos principales, discute las limitaciones encontradas durante el desarrollo y propone líneas de investigación para continuar el trabajo, como la implementación en hardware real.

2 Antecedentes y estado del arte

En este capítulo se debe contextualizar el proyecto dentro del conocimiento existente. No se trata de hacer un simple resumen, sino de demostrar que se entiende qué se ha hecho ya en el área, qué problemas siguen abiertos y qué aporta vuestro trabajo en ese contexto.

2.1 Fundamentos teóricos relevantes

Breve repaso de los conceptos esenciales de cinemática y dinámica de robots.

Introducción a los esquemas de control clásicos (PID, control por realimentación, control óptimo, etc.).

Principios básicos del aprendizaje por refuerzo (Markov Decision Processes, política, recompensa, exploración vs. explotación).

2.2 Estado del arte en robótica y control

Aplicaciones recientes de cinemática y dinámica en robots manipuladores y móviles.

Técnicas actuales de control: comparación entre enfoques tradicionales y métodos basados en inteligencia artificial.

Uso del aprendizaje por refuerzo en robótica: principales algoritmos (DQN, PPO, SAC, DDPG) y retos en su implementación.

2.3 Herramientas y entornos de simulación

Breve revisión de los simuladores más utilizados en robótica (Gazebo, MuJoCo, PyBullet, IsaacSim).

Justificación del uso de IsaacLab/IsaacSim para este proyecto.

2.4 Trabajos y resultados previos

Ejemplos de investigaciones o proyectos similares que hayan aplicado RL en robótica.

Limitaciones encontradas en esos trabajos y vacíos de investigación que motivan este proyecto.

2.5 Relación con el proyecto

Identificación de qué aspectos se tomarán como base para el trabajo (ej. modelado cinemático/dinámico tradicional).

Qué parte supone un reto o innovación (ej. implementación de RL en IsaacLab).

3 Materiales y métodos

La sección Materials and Methods (también llamada Methodology o Experimental Section, según la disciplina) es una parte esencial de artículos y memorias de ámbito académico/docente. Su objetivo principal es que otro estudiante, profesor, investigador, ingeniero, etc., pueda reproducir el trabajo siguiendo las descripciones dadas.

3.1 Modelado cinemático/dinámico del robot + tool (si aplica)

Descripción de los pasos seguidos para modelar el robot dentro del entorno de simulación. Si se ha realizado un modelado teórico, incorporar: Formulación de la cinemática directa e inversa mediante el uso de matrices de transformación homogénea, parámetros de Denavit–Hartenberg. Obtención de la matriz Jacobiana y análisis de singularidades.

Modelado dinámico (ecuaciones de Euler–Lagrange o Newton–Euler) con las hipótesis adoptadas (ej. robot rígido, sin rozamiento, etc.).

Representación clara de ecuaciones y, cuando sea útil, diagramas que apoyen la comprensión.

3.2 Metodología de control y aprendizaje por refuerzo (RL)

Explicación de los esquemas de control diseñados: control clásico de referencia, control basado en RL, comparación, etc.

3.2.1 Esquema de control

Control por LIDAR y cámara

Este control se basa en el guiado por el entorno basandose en la información recibida por el LIDAR, basandose en los datos frontales y laterales que recibimos. Con la cámara lo que hacemos es buscar el color que tiene el objetivo, en nuestro caso es el color rojo, por lo que se va analizando la imagen constantemente, ya que si detectamos un área mínima del

color del objetivo, extraemos cuanto estamos descentrados para poder centrarse y poder ir recto al objetivo.

Control por RL

Justificación de la elección de algoritmos de RL (ej.: PPO, SAC, DDPG), con breve descripción de su funcionamiento.

3.2.2 Algoritmo RL PPO

Definición de recompensas, estados y acciones empleados en el entorno de simulación.

3.2.3 Recompensas, estados y acciones

recompensas

Las recompensas que se le han asignado al robot se basan en si se ha descubierto area o no dentro de la imagen de la cámara. Se han definido zonas de distancia de LIDAR al robot para darle más recompensa o menos:

- Distancia frontal libre(FRONT_CLEAR) : cuando el lidar por delante detecta más de 10m
- Distancia frontal peligrosa(FRONT_DANGER) : Cuando la distancia frontal del LIDAR por delante baja de los 7m
- Distancia muy peligrosa : Es la distancia antes del choque donde antes de chocarse prefiero que vaya hacia atrás el robot, 5.5m

Primero se comprueban las recompensas por alcanzar el objetivo, o por colisión:

- Si se alcanza el objetivo : +30.0
- Si se ha colisionado : -8.0

Si no se ha alcanzado el área mínima de 1000px, las recompensas son las siguientes:

- Si delante es muy libre, es decir, por encima del umbral FRONT_CLEAR, y el robot tiene la acción de ir hacia delante rapido o lento le damos 0.3 por rápido, y 0.15 por lento
- Si delante esta por debajo del umbral FRONT_DANGER y el robot tiene la accion de ir hacia delante, le quitamos 0.6

- Si delante está por debajo del FRONT_CLEAR, y giramos hacia el lado que más depejado esta, le damos 0.4, con esto lo que hacemos es que si por delante vemos una distancia donde no cabe el robot, es innecesario entrar dentro de esa sala, es mejor girar

Estados

Acciones

- Moverse hacia delante rápido
- Moverse hacia delante lento
- Moverse fuerte hacia la derecha
- Moverse suave hacia la derecha
- Moverse fuerte hacia la izquierda
- Moverse suave hacia la izquierda
- Moverle lento hacia atrás

3.3 Entorno de simulación y herramientas utilizadas

Descripción de IsaacSim/IsaacLab: versiones empleadas, configuración inicial y librerías auxiliares. Recursos computacionales (hardware, GPU, sistema operativo).

Configuración de escenarios de entrenamiento (robot, entorno, sensores virtuales, condiciones de interacción).

3.3.1 Unity + Visual estudio

3.3.2 Visual estudio code

3.3.3 TensorFlow

TensorFlow Dashboard es una herramienta que a través de los logs que genera nuestro entrenamiento, va graficando cada 4024 pasos, que son 256 por robot, se grafican los resultados.

3.3.4 KERAS + gym

3.3.5 ROS 1

Hemos empleado ROS1 como pasarela de comunicación entre los robots y el máster, que sería nuestro propio ordenador. Lo que hacemos es que cada robot tiene unos tópicos propios, sobre los cuales envían y reciben la información.

3.3.6 Turtle bot

3.3.7 Entorno

3.3.8 Sensores virtuales

LIDAR

Hemos modelado un LIDAR que publica los valores en un topic de ros, es este colisionado es un código de Unity el cual genera una nube de puntos, y los publica todos a través del tópic

3.3.9 Condiciones de iteración

3.4 Procedimiento de experimentación / entrenamiento

Explicación del flujo de trabajo seguido: preparación de modelos, definición de hiperparámetros, duración de entrenamientos, validación de políticas aprendidas.

3.4.1 Flujo de trabajo

Se probó primero definiendo un escenario muy sencillo, donde si el robot giraba en el primer cruce ya alcanzaba el objetivo final. Esto se modificó y se hicieron 4 escenarios para el mismo entrenamiento, donde el objetivo estaba en diferentes puntos.

Estrategia de comparación: métricas definidas (tiempo de convergencia, estabilidad, error en el seguimiento, etc.).

3.4.2 Estrategia de comparacion

Se definió que cada 1000 pasos en cada robot, se realice una comparacion de modelo, pero se detendria el entrenamineto si la media de los ultimos 1000 episodios no mejoraba durante los 5 siguientes episodios.

Número de episodios, pruebas o repeticiones realizadas.

3.5 Programación

Diagramas de flujo o pseudocódigo de los programas implementados.

4 Resultados

En este capítulo se deben presentar y analizar los resultados obtenidos tras la implementación del proyecto. No se trata solo de mostrar datos, gráficos o tablas, sino de interpretarlos y relacionarlos con los objetivos planteados.

4.1 Validación del modelado matemático

Comparación entre los resultados teóricos (cinemática y dinámica calculadas) y los obtenidos en simulación.

Comprobación de trayectorias, posiciones finales y detección de posibles singularidades.

Discusión sobre la precisión y las limitaciones del modelo.

4.2 Entrenamientos en IsaacLab/IsaacSim

Presentación de las curvas de aprendizaje (recompensa acumulada, convergencia del algoritmo).

Evaluación de los tiempos de entrenamiento y de la estabilidad de la política aprendida.

Análisis de hiperparámetros: cómo afectan al rendimiento y a la velocidad de convergencia.

4.3 Comparación de controladores

Resultados de control clásico vs. control basado en RL.

Métricas de desempeño: error en el seguimiento de trayectorias, suavidad de los movimientos, robustez frente a perturbaciones.

Discusión sobre ventajas y limitaciones de cada enfoque.

4.4 Pruebas finales y validación del sistema

Ejemplos de simulaciones finales con el robot ejecutando las tareas propuestas.

Posibles animaciones, capturas de pantalla o gráficas que ilustren los comportamientos logrados.

Evaluación cualitativa (ej.: naturalidad de movimientos, respuesta a cambios en el entorno).

4.5 Discusión crítica

Relación de los resultados con los objetivos generales y específicos planteados en la introducción.

Identificación de fortalezas, limitaciones y posibles mejoras del trabajo.

5 Conclusiones y trabajo futuro

En este capítulo tendrás la oportunidad de realizar las conclusiones de tu proyecto, volviendo a señalar los aspectos más importantes que se puede ver en la memoria, los logros conseguidos, etc. Además, deberás exponer aquellos aspectos en los que piensas que se podría trabajar de cara a ofrecer una mejor solución que la propuesta y/o ampliar la solución.

5.1 Conclusiones

5.2 Trabajo futuro