

Jornadas de Automática

Plataforma robótica móvil reconfigurable para investigación y docencia

Garrido-Rey, Laura^{a,*}, Navarro-González, J. Daniel^a, Casans, Silvia^a, Sanchis, Enrique^a, Girbés-Juan, Vicent^a

^a Departament d'Enginyeria Electrònica, Universitat de València, ETSE-UV, Av. de la Universitat, s/n, 46100 Burjassot, España.

To cite this article: Garrido-Rey, L., Navarro-González, J. D., Casans, S., Sanchis, E., Girbés-Juan, V. 2025. Reconfigurable mobile robotic platform for research and teaching. *Jornadas de Automática*, 46. <https://doi.org/10.17979/ja-cea.2025.46.12235>

Resumen

En este trabajo se presenta el diseño y desarrollo de una plataforma robótica móvil terrestre modular orientada a investigación y docencia. Esta plataforma permite realizar tres configuraciones cinemáticas (diferencial, skid-steering y omnidireccional) mediante el intercambio de sus ruedas. La plataforma está organizada en tres niveles donde se integran sensores, actuadores y procesadores. El control de bajo nivel se gestiona mediante un microcontrolador ESP32 bajo FreeRTOS, mientras que una Raspberry Pi 5 ejecuta tareas de planificación y navegación utilizando ROS2 Jazzy. En el microcontrolador se han implementado los cálculos de la cinemática directa e inversa, el control PID para regular la velocidad de las ruedas y un Filtro de Kalman Extendido para la estimación del estado del robot (odometría). Además, se ha desarrollado un gemelo digital en ROS2 para simular el comportamiento de cada configuración. Finalmente, se ha validado una estrategia de control visual tipo *Leader-Follower* entre robots mediante la detección de marcadores ArUco y seguimiento visual basado en posición, permitiendo la colaboración entre sistemas multi-robot.

Palabras clave: Robots móviles autónomos, Información y fusión sensorial, Navegación cooperativa, Localización, Creación de mapa, Control de movimiento, Navegación robótica, Sistemas multivehículo.

Reconfigurable mobile robotic platform for research and teaching

Abstract

This work presents the design and development of a modular ground mobile robotic platform aimed at research and educational purposes. The platform supports three interchangeable kinematic configurations (differential, skid-steering, and omnidirectional) by swapping its wheels. It is structured into three layers that integrate sensors, actuators, and processors. Low-level control is handled by an ESP32 microcontroller running FreeRTOS, while high-level tasks such as planning and navigation are performed on a Raspberry Pi 5 using ROS2 Jazzy. The microcontroller implements forward and inverse kinematics computations, PID control for wheel speed regulation, and an Extended Kalman Filter for robot's state estimation (odometry). Additionally, a digital twin has been developed in ROS2 to simulate the behavior of each configuration. Finally, a visual Leader-Follower control strategy has been validated between robots using ArUco marker detection and position-based visual servoing, enabling collaboration within multi-robot systems.

Keywords: Autonomous mobile robots, Information and sensor fusion, Cooperative navigation, Localization, Map building, Motion control, Robot Navigation, Multi-vehicle systems.

1. Introducción

El uso de robots en diferentes sectores está aumentando con el paso de los años. La industria es uno de los sectores

que ha estado utilizando robots industriales para las cadenas de montaje, sobre todo en el sector automovilístico, y ha empezado a instaurar robots móviles terrestres autónomos para la logística (Sprunk *et al.*, 2017). En cambio, hay otros

sectores, como el de la agricultura, que todavía están investigando diferentes aplicaciones robóticas donde los robots puedan participar en las actividades agrícolas (Conejero *et al.*, 2025). Este crecimiento ha impulsado el desarrollo de plataformas robóticas versátiles que puedan adaptarse a diferentes entornos y necesidades.

La locomoción es un aspecto clave en el diseño de robots móviles. Existen múltiples configuraciones cinemáticas cuyas ventajas y limitaciones dependen del tipo de aplicación y entorno. Esta variedad también influye directamente en las estrategias de control, planificación y navegación que se implementan, incluyendo enfoques clásicos como el control PID o más avanzados como el control visual *Leader-Follower* (Chen *et al.*, 2022), ampliamente usado en escenarios colaborativos multi-robot.

En este contexto, este trabajo presenta el diseño y desarrollo de una plataforma robótica móvil terrestre, modular y orientada a la investigación y docencia. La plataforma permite intercambiar entre tres configuraciones cinemáticas distintas mediante el cambio de ruedas, e integra sensores, actuadores y procesamiento distribuido en varios niveles. Se ha implementado control de bajo nivel sobre un microcontrolador ESP32 con FreeRTOS, mientras que una Raspberry Pi 5 ejecuta ROS2 Jazzy para la planificación y navegación. Además, se ha desarrollado una simulación del robot en ROS2 para poder realizar pruebas, y se ha validado una estrategia de control visual tipo *Leader-Follower* mediante la detección de marcadores ArUco, permitiendo la colaboración en entornos multi-robot.

2. Estado del arte

Los robots móviles se pueden clasificar, dependiendo de su tipo de locomoción, en robots con ruedas, con orugas, con patas, híbridos, aéreos o submarinos (Siegwart *et al.*, 2011). Aunque el uso de cada uno de estos robots depende del tipo de aplicación que se quiera emplear, los más utilizados son los robots con ruedas. Estos robots se clasifican dependiendo del tipo de ruedas y de su configuración, teniendo así cinco tipos de robots diferentes.

Por un lado, están los robots diferenciales, los cuales presentan una de las configuraciones más comunes, ya que se trata de robots con dos ruedas motrices independientes y una o más ruedas pasivas que se utilizan como punto de apoyo para dar mayor estabilidad a la plataforma. Permiten realizar giros sobre su propio eje, pero no se pueden mover de forma instantánea en cualquier dirección y están limitados a superficies regulares, principalmente en entornos de interior. Por otro lado, los robots skid-steering tienen dos pares de ruedas motrices y giran por desplazamiento lateral, como los tanques y robots oruga, que son otro tipo de robot con configuración diferencial. Tienen buena tracción en terrenos irregulares, pero presentan un desgaste mecánico alto por la fricción.

Otro tipo de configuración es la de los robots triciclo, que incorporan una rueda directriz y ruedas de tracción que pueden estar posicionadas en la parte delantera o trasera del robot (Girbés *et al.*, 2010). Son adecuados para aplicaciones básicas o de bajo coste, pero están limitados en maniobrabilidad ya que no pueden girar sobre su propio eje.

Los robots tipo Ackermann son equivalentes al robot triciclo y su configuración es similar a la de los automóviles. Disponen de dos ruedas de dirección que suelen ir delante y dos ruedas motrices que habitualmente son las traseras, aunque existen diferentes combinaciones. Son buenos para realizar navegación en exteriores, pero no pueden hacer movimientos laterales ni girar sobre su propio eje, ya que su radio de curvatura está limitado.

Por último, los robots omnidireccionales son un tipo de robot holónomico, es decir, se pueden mover y rotar en cualquier dirección de forma simultánea. Aunque la configuración omnidireccional más habitual es con ruedas Mecanum o suecas, formadas por un conjunto de rodillos para incrementar su grado de movilidad, existen otras configuraciones con ruedas reorientables de tipo estándar. Los robots holónomos con ruedas Mecanum suelen incorporar tres o cuatro ruedas y presentan una alta maniobrabilidad en espacios reducidos, pero tienen menor tracción que otro tipo de ruedas.

En cuanto al control del movimiento de estos robots, la elección de la configuración cinemática tiene un efecto directo sobre la estrategia de control que se requiera para la navegación y ejecución de tareas. Estos métodos de control de movimiento se clasifican dependiendo del nivel de modelado y complejidad del entorno. En robots móviles que se desplazan a baja velocidad o que no tienen mucha masa e inercia es común utilizar el control cinemático, el cual se basa en ecuaciones de velocidad lineal y angular. En cambio, para vehículos autónomos con grandes masas e inercias es mejor utilizar un control dinámico donde se consideran las fuerzas y pares que afectan al movimiento del robot móvil. Para aplicaciones de navegación, se utilizan técnicas de seguimiento de trayectoria (Galati y Mantriota, 2023), seguimiento de camino (Girbés *et al.*, 2014) o controladores clásicos como PID (Proporcional-Integral-Derivativo), los cuales son ampliamente utilizados para la regulación de la velocidad de las ruedas. También, se emplean otros enfoques de control más avanzado como son el control predictivo basado en modelo (MPC) (Ebel y Eberhard, 2021), el basado en consenso (Alonso-Mora *et al.*, 2016), el basado en redes neuronales y en aprendizaje automático (Tsai *et al.*, 2017) y el basado en formación y seguimiento (*Leader-Follower*) (Chen *et al.*, 2022), entre otros.

En aplicaciones multi-robot se utiliza control visual, más conocido como *Visual Servoing* (VS), el cual es una técnica que permite controlar el movimiento de un robot utilizando retroalimentación visual en tiempo real. Su aplicación se emplea tanto en robots móviles como en robots manipuladores y la información se captura a través de una o más cámaras, traduciéndose en señales de control que guían al efecto final del manipulador o al robot móvil hacia una configuración deseada.

El VS ha evolucionado hacia dos enfoques principales: *Image-Based Visual Servoing* (IBVS) y *Position-Based Visual Servoing* (PBVS). El IBVS es un control basado en las coordenadas de características visuales en el plano de la imagen, lo que permite evitar una reconstrucción completa de la escena 3D y reduce la dependencia de modelos geométricos haciendo robusto frente a errores de calibración y condiciones de iluminación cambiantes (Zhang and Li, 2018),

(Chan and Srigrarom, 2023). Por otro lado, el PBVS utiliza la información visual para estimar explícitamente la pose 3D del objetivo con respecto a la cámara, lo que permite diseñar leyes de control directamente en el espacio de trabajo (Cartesiano). Esta característica lo convierte en un enfoque especialmente atractivo para tareas que requieren precisión espacial y trayectorias suaves, como el seguimiento *Leader-Follower* entre robots móviles (Wang, Liu and Li, 2014), (Al-Shafoon et al., 2018).

3. Arquitectura del robot

Se ha diseñado un robot móvil terrestre modular para su uso tanto en tareas de investigación como en docencia, por ejemplo, en prácticas de laboratorio o aprendizaje basado en proyectos en asignaturas de grado y máster universitario. Este robot móvil tiene una configuración tipo sándwich de tres capas en las cuales se disponen todos los elementos necesarios para su movimiento y control.

En la Figura 1 se muestra la plataforma robótica, cuyo diseño permite la posibilidad de tener tres configuraciones distintas: un robot omnidireccional añadiendo cuatro ruedas Mecanum, un robot skid-steering con cuatro ruedas estándar, o un robot diferencial con dos ruedas traseras estándar y una rueda loca (tipo Caster) o esférica en la parte delantera. Aunque no se ha implementado todavía, la configuración triciclo también es posible si se añade una pieza para insertar una rueda delantera de dirección con un servomotor.

3.1. Cinemática

En la Figura 2 se muestran los posibles modelos cinemáticos y las dimensiones de la plataforma móvil. Conociendo el radio de las ruedas ($r=0,04$ m las de tipo Mecanum y $r=0,0375$ m las de tipo estándar), la semi-distancia entre los ejes delantero y trasero ($a=0,1$ m) y la semi-distancia entre las ruedas de las partes izquierda y derecha ($b=0,1$ m), se puede calcular la cinemática directa e inversa para controlar el movimiento de cada tipo de robot.

El robot omnidireccional tiene cuatro ruedas directrices independientes cuyas velocidades angulares ($\omega_1, \omega_2, \omega_3$ y ω_4) se calculan a través de su cinemática inversa para seguir una trayectoria de referencia en el plano (v_x^r, v_y^r y ω_z^r). La velocidad de la plataforma móvil se representa mediante las variables v_x, v_y y ω_z . En las ecuaciones (1) y (2) se muestra la cinemática directa e inversa del robot omnidireccional, respectivamente.

$$\begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \frac{r}{4} & \frac{r}{4} & \frac{r}{4} & \frac{r}{4} \\ \frac{r}{4} & \frac{r}{4} & \frac{r}{4} & -\frac{r}{4} \\ -\frac{r}{4} & \frac{r}{4} & \frac{r}{4} & \frac{r}{4} \end{bmatrix} \cdot \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \cdot \begin{bmatrix} 1 & -1 & -(a+b) \\ 1 & 1 & (a+b) \\ 1 & 1 & -(a+b) \\ 1 & -1 & (a+b) \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} \quad (2)$$



Figura 1: Configuraciones de la plataforma robótica. De izquierda a derecha: robot omnidireccional, robot skid-steering y robot diferencial.

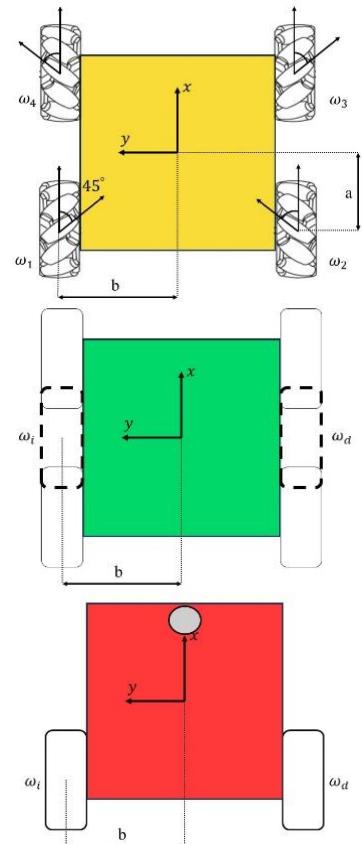


Figura 2: Diagramas de los diferentes modelos cinemáticos del robot móvil.

Con respecto al robot en configuración skid-steering, su cinemática se puede simplificar para que coincida con la cinemática del robot diferencial, aplicando la misma velocidad a las ruedas que están en el mismo lado (izquierdo/derecho). En esta configuración tan solo se tienen en cuenta dos ruedas motrices, cuyas velocidades se representan como ω_d para la rueda derecha y ω_i para la rueda izquierda. En cuanto a las dimensiones del robot, tan solo se tiene en cuenta la distancia que hay desde las ruedas al punto central del robot (b). En el caso del robot con configuración skid-steering, este punto está en el centro del robot mientras que, en el caso del robot diferencial, este punto está alineado con las ruedas motrices. Dado que se trata de una configuración no-holónómica, estos robots únicamente pueden desplazarse y rotar en la dirección de las ruedas motrices, dando lugar a una velocidad lineal v_x y una velocidad angular ω_z locales al sistema de referencia del robot. La cinemática directa de esta configuración está representada en (3) y la cinemática inversa en (4).

$$\begin{bmatrix} v_x \\ \omega_z \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2b} & -\frac{r}{2b} \end{bmatrix} \cdot \begin{bmatrix} \omega_d \\ \omega_i \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} \omega_d \\ \omega_i \end{bmatrix} = \frac{1}{r} \cdot \begin{bmatrix} 1 & b \\ 1 & -b \end{bmatrix} \cdot \begin{bmatrix} v_x \\ \omega_z \end{bmatrix} \quad (4)$$

3.2. Arquitectura hardware

En cuanto a la distribución de los elementos que conforman el robot, en la base inferior se encuentran una batería tipo LiPO, un convertidor DC-DC y motores de continua con caja reductora y encoder de tipo Hall, necesitando cuatro para la configuración omnidireccional o skid-steering y tan solo dos para la configuración diferencial. En la base intermedia están los drivers de potencia (se usan dos para la configuración omnidireccional o skid-steering y solo uno para la configuración diferencial); un sensor de medición inercial (IMU) modelo MPU9250 que incorpora acelerómetro, giroscopio y magnetómetro; un microcontrolador ESP32 model NodeMCU ESP-WROOM-32; y una Raspberry Pi 5 (RPi5) de 8 GB. En la base superior, se ha añadido un sensor LIDAR modelo S2 de RP lidar, mientras que en la parte frontal de la base intermedia se ha dispuesto un módulo que alberga una cámara de resolución 640x480 píxeles.

El ESP32 y la Raspberry Pi 5 están conectados mediante un cable USB y se comunican mediante un protocolo serie utilizando la librería MicroROS. El microcontrolador se encarga de realizar el control de bajo nivel (lectura de sensores, estimación de la odometría y control de velocidad de las ruedas), mientras que la Raspberry lleva a cabo todas aquellas tareas de más alto nivel relacionadas con la navegación y la planificación de trayectorias. La Figura 3 muestra la conexión entre los diferentes componentes.

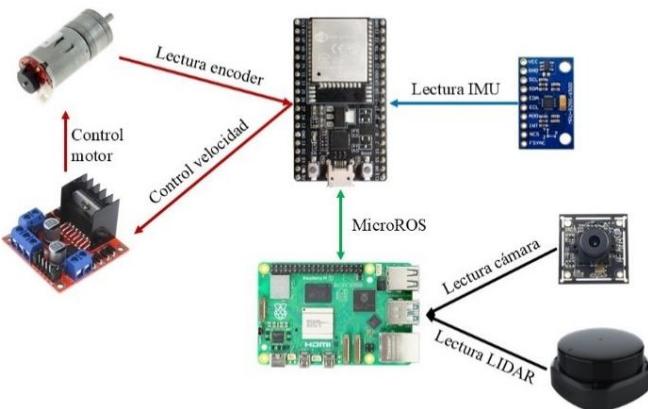


Figura 3: Esquema de la conexión entre los diferentes componentes electrónicos que conforman el robot y su función.

3.3. Arquitectura software

El ESP32 es un microcontrolador con doble núcleo, que ofrece la posibilidad de ejecutar tareas de forma paralela y de mejorar el rendimiento global del sistema de control, sobre todo en aplicaciones donde se requiere procesar datos en tiempo real. En el núcleo principal se procesa toda la información que proviene de los motores y se realizan los

cálculos necesarios para el movimiento del robot, mientras que en el segundo núcleo se procesa la información que proviene de la IMU.

Como gestor de las diferentes tareas se utiliza FreeRTOS (Free Real Time Operating System), un sistema operativo en tiempo real de código abierto que facilita las comunicaciones entre tareas de sistemas embebidos. El ESP32 alberga la cinemática directa y la cinemática inversa según el tipo de robot que se va a utilizar y calcula la velocidad a la que deben ir las ruedas dependiendo de la velocidad de referencia necesaria para alcanzar la posición objetivo. Se ha diseñado un control PID que, junto con la lectura de los encoders de los motores, permite controlar la velocidad de las ruedas en todo momento. Además, se ha implementado un filtro de Kalman extendido (EKF) que permite estimar la posición, la orientación y las velocidades lineal y angular del robot gracias a la fusión sensorial de la IMU y los encoders.

Por otro lado, la Raspberry Pi se encarga de procesar la lectura de la cámara y el sensor LIDAR, de recoger la información que viene del ESP32 y de crear mapas del entorno y planificar trayectorias para que el robot alcance la posición requerida. También es posible realizar una teleoperación del robot mediante un dispositivo de control remoto. Todas estas acciones se realizan a través de ROS (Robot Operating System), en este caso la distribución ROS2 Jazzy compatible con Ubuntu 24.04, siendo esta la última versión lanzada para la RPi5. Como ya se ha comentado, la comunicación entre el ESP32 y la RPi5 se realiza a través de MicroROS para poder integrar el microcontrolador en el ecosistema de ROS. De esta forma, la información del ESP32 se recibe a través de nodos y topics de ROS2, facilitando así su adquisición desde la RPi5.

En las Figuras 4 y 5 se muestra un esquema de cómo se ha dispuesto la información necesaria para la navegación autónoma del vehículo mediante la plataforma ROS2. Por un lado, la RPi5 recoge las velocidades lineal y angular a las que debe ir el robot para alcanzar la posición deseada y las almacena en el mensaje tipo *Twist* del topic */cmd_vel* (ver Figura 4) para que luego el ESP32 pueda realizar el control de la velocidad de las ruedas implementando la cinemática inversa del robot. Por otro lado, el ESP32 recoge medidas de los encoders y de la IMU, calcula la orientación y la cinemática directa del robot, implementa un filtro EKF para estimar la posición y orientación del robot (ver Figura 5). Toda esta información la almacena en el mensaje tipo *Odometry* del topic */odom* para que lo pueda utilizar la Raspberry Pi. De esta forma, se tiene un control de la posición y la velocidad a la que va el robot teniendo en cuenta a qué velocidad debería ir y en qué dirección para poder navegar de forma autónoma y alcanzar la posición deseada.

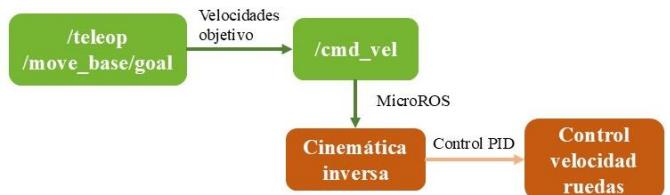


Figura 4: Esquema de la disposición de la información dentro de los topics de ROS (RPi5 hacia ESP32). En verde se muestra la parte de la RPi5 y en naranja la del ESP32.



Figura 5: Esquema de la disposición de la información dentro de los topics de ROS (ESP32 hacia RPi5). En verde se muestra la parte de la RPi5 y en naranja la del ESP32.

3.4. Simulación del sistema físico

Para poder simular los comportamientos del robot, se ha generado una simulación del robot físico en ROS2. Para ello, se ha desarrollado un modelo URDF para cada configuración de robot. De esta forma se puede simular una navegación autónoma del robot antes de llevarla a cabo en un entorno real. Además, para poder simular un comportamiento más próximo a la realidad se han tenido en cuenta las incertidumbres de todos los componentes que conforman el robot. En la Figura 6 se muestra la simulación del robot físico en cada una de las configuraciones nombradas.

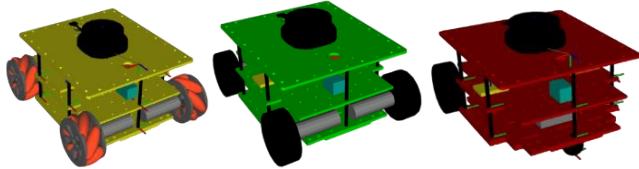


Figura 6: Simulación del robot dependiendo de su configuración. De derecha a izquierda: robot omnidireccional, robot skid-steering y robot diferencial.

En cuanto a la navegación autónoma, se ha utilizado el paquete *slam_toolbox* para realizar el mapeo del entorno, mientras que para la planificación y navegación del robot se ha implementado el framework *Nav2*. En la Figura 7 se muestra la simulación del robot omnidireccional navegando de forma autónoma por un mapa previamente creado a la par que el robot físico navega por el entorno.

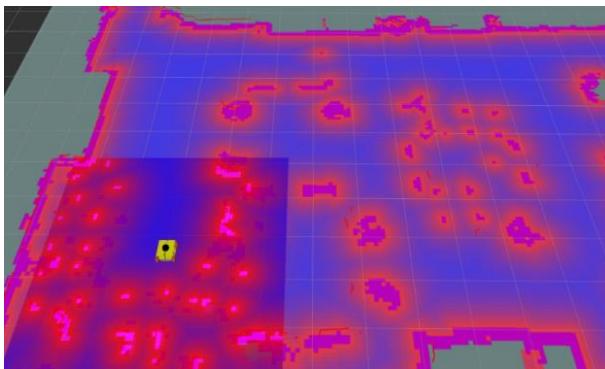


Figura 7: Robot móvil navegando por un entorno en el visor de ROS2 (RViz).

4. Leader-Follower

Para poder llevar a cabo tanto docencia como investigación en diferentes aplicaciones de los robots, se han diseñado nueve robots idénticos, permitiendo así tener una flota de robots móviles terrestres configurable. Esto permite poder realizar pruebas de *Leader-Follower* para poder determinar qué tipo de control es el adecuado dependiendo de la necesidad que se tenga.

La primera prueba que se ha hecho ha sido un control por realimentación visual (Visual Servoing) donde un robot móvil diferencial actúa como líder y otro robot omnidireccional asume el rol de seguidor. Tal y como se muestra en la Figura 8, el líder incorpora un código ArUco en su parte trasera y el seguidor detecta el marcador mediante visión artificial para obtener su posición y orientación con respecto al líder. La detección del código ArUco se hace a través de la librería OpenCV y el lenguaje de programación Python (ver Figura 9). De esta forma, el robot líder planifica su trayectoria, mientras que el robot seguidor lo sigue a través de un control visual, dejando una distancia de separación previamente especificada.

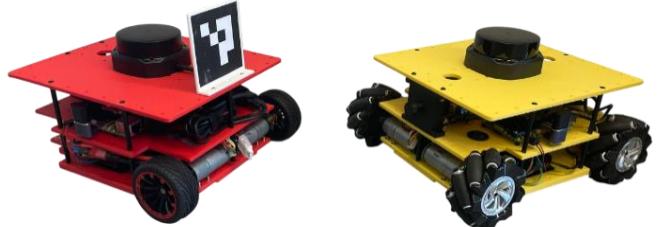


Figura 8: Robots en aplicación *Leader-Follower* mediante VS con detección de marcador ArUco.

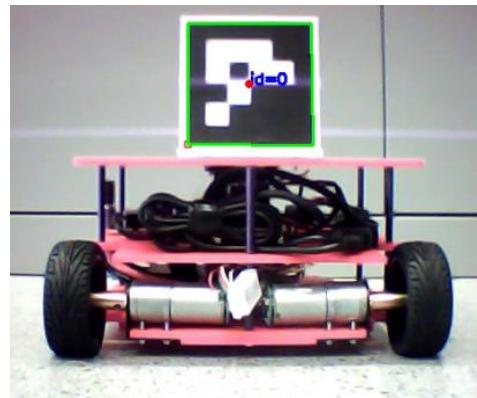


Figura 9: Detección del marcador ArUco a través de la librería OpenCV y el lenguaje de programación Python desde el robot seguidor.

5. Conclusiones

En este trabajo se ha desarrollado una plataforma robótica móvil terrestre modular orientada a docencia e investigación, capaz de adoptar distintas configuraciones cinemáticas: diferencial, skid-steering y omnidireccional. La arquitectura distribuida de la plataforma está organizada en tres niveles funcionales permitiendo realizar tareas de bajo nivel en un microcontrolador ESP32 bajo FreeRTOS y ejecutar algoritmos avanzados de navegación en una Raspberry Pi 5 utilizando ROS2 Jazzy.

Se ha implementado un control de velocidad mediante PID, así como un filtro de Kalman extendido para la estimación del estado. Además, se ha desarrollado una simulación digital del robot físico en ROS2 que permite realizar pruebas con las distintas configuraciones cinemáticas de la plataforma. La funcionalidad multi-robot ha sido validada mediante una estrategia de control visual *Leader-Follower* basada en la detección de marcadores ArUco y seguimiento posicional.

Asimismo, se ha diseñado, fabricado y puesto a punto una flota de robots móviles compuesta por ocho plataformas idénticas, lo que facilita tanto el desarrollo de actividades prácticas en entornos docentes como la investigación y desarrollo de algoritmos de control en aplicaciones de cooperación multi-robot.

6. Líneas futuras

La primera mejora que se plantea es adaptar la plataforma móvil para poder añadir la configuración triciclo. Se diseñarán las piezas necesarias para introducir una rueda delantera de dirección y se usarán las ruedas traseras para la tracción de los robots.

Como continuación de este trabajo, se propone investigar y hacer un estudio comparativo de los algoritmos de planificación y control más adecuados para llevar a cabo la estrategia *Leader-Follower*. Se planteará el uso de robots que tengan la misma configuración y también arquitecturas multiagente donde las configuraciones sean diferentes. Los algoritmos de control más prometedores son los siguientes: control basado en consenso, para lograr coordinación distribuida y robusta frente a fallos de comunicación o percepción; control predictivo basado en modelo (MPC), que permite anticipar obstáculos y comportamientos no lineales en trayectorias más complejas, así como añadir restricciones; técnicas de VS híbrido (IBVS+PBVS) con buena precisión de posición y robustez frente a ruido visual.

El objetivo es poder llegar a controlar una flota robótica destinada al transporte de objetos tanto de forma individual como conjunta en entornos reales con obstáculos, diferentes condiciones de iluminación, y superficies no planas.

Agradecimientos

Esta investigación ha sido financiada por MCIN/AEI/10.13039/501100011033, Agencia Estatal de Investigación (Gobierno de España), a través del proyecto PID2023-146747NA-C22, y por la Unión Europea a través del programa Europa Digital (Grant Agreement nº101083002).

Referencias

- Alonso-Mora, J., Montijano, E., Schwager, M., & Rus, D. (2016) 'Distributed multi-robot formation control among obstacles: A geometric and optimization approach with consensus', *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5356–5363. DOI: 10.1109/ICRA.2016.7487747.
- Al-Shanoon, A., Tan, A. H., Lang, H., & Wang, Y. (2018) 'Mobile robot regulation with position based visual servoing', *IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications*, pp. 1-6. DOI: 10.1109/CIVEMSA.2018.8439978.
- Chan, W.K. & Srirarom, S. (2023) 'Image-based Visual-Servoing for Air-to-Air Drone Tracking & Following with Model Predictive Control', *Proceedings of 2023 SICE International Symposium on Control Systems*, pp. 1-7. DOI: 10.23919/SICEISCS57194.2023.10079208.
- Chen, T., Sun, Y., Niu, X., Lan, Y., Fang, W., & Liu, P. (2022) 'Formation control for second-order nonlinear multi-agent systems with external disturbances via adaptive method', *Proceedings - 2022 Chinese Automation Congress, CAC 2022*, 2022-January, pp. 5616–5620. DOI: 10.1109/CAC57257.2022.10055393.
- Conejero, M. N., Montes, H., Bengoechea-Guevara, J. M., Garrido-Rey, L., Andújar, D., & Ribeiro, A. (2025) 'A collaborative robotic fleet for yield mapping and manual fruit harvesting assistance', *Computers and Electronics in Agriculture*, 235, 110351. DOI: 10.1016/J.COMPAG.2025.110351.
- Ebel, H. & Eberhard, P. (2021) 'Non-Prehensile Cooperative Object Transportation with Omnidirectional Mobile Robots: Organization, Control, Simulation, and Experimentation', *2021 International Symposium on Multi-Robot and Multi-Agent Systems, MRS 2021*, pp. 1–10. DOI: 10.1109/MRS50823.2021.9620541.
- Galati, R. & Mantriota, G. (2023) 'Path Following for an Omnidirectional Robot Using a Non-Linear Model Predictive Controller for Intelligent Warehouses', *Robotics*, 12(3), p. 78. DOI: 10.3390/ROBOTICS12030078.
- Giurbés, V., Armesto, L., & Tornero, J. (2010, June). Pisala project. intelligent sensorization for line tracking with artificial vision. *41st International Symposium on Robotics*, pp. 1-6. VDE.
- Giurbés, V., Armesto, L., & Tornero, J. (2014). 'Path following hybrid control for vehicle stability applied to industrial forklifts'. *Robotics and Autonomous Systems*, 62(6), 910-922.
- Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). *Introduction to autonomous mobile robots*. MIT press. ISBN: 9780262295321
- Sprung, C., Lau, B., Pfaff, P., & Burgard, W. (2017) 'An accurate and efficient navigation system for omnidirectional robots in industrial environments', *Autonomous Robots*, 41(2), pp. 473-493. DOI: 10.1007/s10514-016-9557-1.
- Tsai, C. C., Wu, H. L., Tai, F. C., & Chen, Y. S. (2017) 'Distributed consensus formation control with collision and obstacle avoidance for uncertain networked omnidirectional multi-robot systems using fuzzy wavelet neural networks', *Int. Journal of Fuzzy Systems*, 19(5), pp. 1375–1391. DOI: 10.1007/s40815-016-0239-0.
- Wang, K., Liu, Y. & Li, L. (2014) 'Visual servoing trajectory tracking of nonholonomic mobile robots without direct position measurement', *IEEE Trans. on Robotics*, 30(4), pp. 1026–1035. DOI: 10.1109/TRO.2014.2317891.
- Zhang, Y. & Li, S. (2018) 'A Neural Controller for Image-Based Visual Servoing of Manipulators with Physical Constraints', *IEEE Transactions on Neural Networks and Learning Systems*, 29(11), pp. 5419–5429. DOI: 10.1109/TNNLS.2018.2802650.