

UNIVERSIDAD SANTO TOMÁS

Implementación de Rutinas en ROS2 para la aplicación de algoritmos multiagentes en el movimiento de Robots tipo *Soccer Small Size*.

Realizado por

Daniel Castiblanco Rey
John Felipe Vega Otálora



Grupo de Investigación GED (Grupo de Estudio y Desarrollo en Robótica)
Facultad de Ingeniería Electrónica
División de Ingenierías

2024

Implementación de Rutinas en ROS2 para la aplicación de algoritmos multiagentes en el movimiento de Robots tipo *Soccer Small Size*.

Realizado por

Daniel Castiblanco Rey
John Felipe Vega Otálora

Proyecto de Grado presentado en cumplimiento del requisito parcial
para optar por el grado de Ingeniería Electrónica

Dirigido por

Sindy Paola Amaya

Codirigido por

Armando Mateus Rojas
David Alejandro Martínez Vásquez

Grupo de Investigación GED (Grupo de Estudio y Desarrollo en Robótica)
Facultad de Ingeniería Electrónica
División de Ingenierías

2024

Autoridades de la Universidad

RECTOR GENERAL

R.P. FRAY ÁLVARO JOSÉ ARANGO RESTREPO, O.P.

VICERRECTOR ADMINISTRATIVO Y FINANCIERO GENERAL

R.P. FRAY HERNÁN YESID RIVERA ROBERTO, O.P.

VICERRECTOR ACADÉMICO GENERAL

R.P. FRAY MAURICIO ANTONIO CORTÉS GALLEGO, O.P.

SECRETARIO GENERAL

Dra. INGRID LORENA CAMPOS VARGAS

DECANO DIVISIÓN DE INGENIERÍAS

R.P. FRAY JAVIER ANTONIO HINCAPIÉ ARDILA, O.P.

SECRETARIA DE DIVISIÓN

E.C. LUZ PATRICIA ROCHA CAICEDO

DECANO FACULTAD DE INGENIERÍA ELECTRÓNICA

PHD. CARLOS ANDRÉS TORRES PINZÓN

Nota de aceptación

Firma del autor

Firma del autor

Firma del jurado

Firma del jurado

Advertencia

La Universidad Santo Tomás no se hace responsable de las opiniones y conceptos expresados en el trabajo de grado, solo velará por qué no se publique nada contrario al dogma ni a la moral católica y porque el trabajo no tenga ataques personales y únicamente se vea el anhelo de buscar la verdad científica.

Capítulo III –Art. 46 del Reglamento de la Universidad Santo Tomás.

Dedicatoria

Dedico esta tesis a Dios, por ser mi guía y fortaleza en cada paso de mi vida. Su amor y bendiciones me han dado el valor y la determinación para alcanzar mis metas. A mi abuelita Maria del Rosario que en paz descanse, gracias a ella me dio la fuerza para continuar en los momentos en que me sentía abrumado y por ella fue que seguí. Siempre estarás en mis recuerdos y desde el cielo en donde te encuentras te dedico este logro que me ayudaste a obtener. A mis padres y mi hermano, quienes con su amor, sacrificio y apoyo incondicional han sido la base de todo lo que soy y he logrado. Gracias por creer en mí y por darme las herramientas para seguir adelante. A mis amigos, por su amistad, risas y momentos compartidos. Gracias por estar ahí en los buenos y malos momentos, y por hacer de este viaje algo más llevadero.

Daniel Castiblanco Rey

A mis padres, por ser mi guía constante, por su amor incondicional y por enseñarme el valor del esfuerzo y la perseverancia. A mi abuela, por su sabiduría y ternura, por ser un ejemplo de fortaleza y por llenar mi vida de historias y enseñanzas que atesoro profundamente. A la memoria de mi abuelo, quien en paz descanse, por su legado de sabiduría, integridad y amor que perdura en cada uno de mis logros. Aunque ya no esté físicamente con nosotros, su espíritu y sus enseñanzas siguen guiando mis pasos. Este logro es también suyo, y lo llevo en mi corazón siempre. A mi pareja, por ser mi apoyo inquebrantable, por su amor, paciencia y por creer siempre en mí. Gracias por estar a mi lado en cada paso de este camino. A cada uno de ellos dedico este trabajo como muestra de gratitud por su apoyo, amor y compañía.

John Felipe Vega Otálora

Agradecimientos

La realización de esta tesis no habría sido posible sin el apoyo y la colaboración de muchas personas. En primer lugar, quiero expresar mi más profundo agradecimiento a Dios, por haberme dado la fortaleza, sabiduría y salud necesarias para llevar a cabo este trabajo. Sin su guía y bendiciones, este logro no habría sido posible.

Quiero también agradecer a mi directora Sindy Paola Amaya y a mis codirectores Armando Mateus y David Martínez por sus constantes guías, asesoramientos y paciencia durante todo el proceso de investigación. Sus valiosas sugerencias y apoyo incondicional han sido fundamentales para la realización de este trabajo.

A la Universidad Santo Tomás, quiero expresar mi gratitud por brindarme los recursos necesarios para llevar a cabo esta investigación. Así mismo, agradezco al grupo GED, por facilitarme el acceso a las herramientas y espacios necesarios para realizar las pruebas y análisis.

Un agradecimiento especial a mis amigos y grupo de trabajo, por su amistad, apoyo moral y por compartir conmigo sus conocimientos y experiencias. Sus palabras de aliento y compañía hicieron más llevadero este camino.

Y por último, pero no menos importante a mi familia, mis más sinceros agradecimientos por su amor, comprensión y apoyo incondicional a lo largo de estos años de estudio. Su confianza en mí y sus constantes ánimos han sido una fuente inagotable de motivación.

Daniel Castiblanco Rey

Quiero expresar mi más profundo agradecimiento a mi familia, cuya fortaleza y amor han sido mi inspiración constante. Su apoyo inquebrantable y su confianza en mí han sido pilares fundamentales en este recorrido. No hay palabras suficientes para agradecerles por creer en mí y por ser mi refugio en cada paso de este camino.

A mi pareja, gracias por tu comprensión y paciencia. Tu aliento constante y tu fe en mis capacidades me han dado la fuerza necesaria para seguir adelante, incluso en los momentos más desafiantes. Eres mi mayor apoyo y motivación.

A mis directores de tesis, agradezco su inestimable orientación y sabiduría. Su pasión por la investigación y su dedicación han sido una fuente de inspiración para mí. Gracias por su compromiso incansable, por sus consejos y por guiarme con paciencia y rigor a lo largo de este proyecto. Su influencia ha sido crucial en la consecución de este logro.

A mis compañeros de carrera, con quienes he compartido innumerables horas de estudio, risas y desafíos. Gracias por su amistad, por estar siempre dispuestos a colaborar y por los momentos de aprendizaje conjunto. Su apoyo y camaradería han hecho de este viaje una experiencia enriquecedora y memorable.

John Felipe Vega Otálora

Índice general

Lista de Figuras	IX
Lista de Tablas	X
Glosario	XI
1. Resumen	1
2. Abstract	3
3. Introducción	5
3.1. Planteamiento del Problema	6
3.2. Objetivos	7
3.2.1. Objetivo General	7
3.2.2. Objetivos Específicos	7
3.3. Justificación	8
3.4. Impacto Social	9
4. Estado del Arte	11
4.1. <i>Small Size Robot Team</i>	11
4.2. Inteligencia Artificial	12
4.3. Visión Artificial	13
4.4. Sistemas Multiagentes	13
4.5. Sistema Operativo de Robots	14
5. Marco Teórico	16
5.1. <i>STOx's Soccer Small Size</i>	16
5.1.1. Estructura Mecánica y Electrónica	16
5.1.2. Modelos Omnidireccionales	18
5.1.2.1. Matriz de Fuerzas y Aceleración	19
5.1.2.2. Matriz de Velocidad	20
5.1.3. Comunicación mediante Radio Frecuencia	21
5.2. Sistemas Multiagente	22
5.2.1. Consenso	22
5.2.1.1. Juegos Potenciales	23

5.2.1.2. Aprendizaje por Refuerzo	24
5.3. Visión Artificial	24
5.3.1. Segmentación	25
6. Diseño Metodológico	27
6.1. Fases y Metodología	27
6.2. Fase 1: Inspección y revisión bibliográfica	27
6.2.1. Adecuación de las plataformas robóticas	27
6.3. Fase 2: Actualización a ROS2	28
6.4. Fase 3: Implementación de la estrategia multiagente en las plataformas robóticas	28
6.5. Fase 4: Validación de las técnicas implementadas en el sistema.	28
7. Desarrollo Conceptual	29
7.1. Fase 1	29
7.1.1. Selección de Plataformas	29
7.1.2. Localización Mediante Sistema Bitcraze	30
7.1.3. Detección por Profundidad	31
7.1.4. Detección por Color	33
7.2. Fase 2	34
7.2.1. Nodo Publicador	34
7.2.2. Nodo Subscriptor	35
8. Implementación y Resultados	37
8.1. Fase 1	37
8.1.1. Programación de rutinas de control	37
8.2. Fase 2	39
8.2.1. Integración a ROS2	39
8.3. Fase 3	42
8.3.1. Aplicación de la Técnica Multiagente	42
8.4. Fase 4	46
8.4.1. Validación en Escenarios de Prueba	46
9. Conclusiones y Trabajos Futuros	52
9.1. Conclusiones	52
9.2. Trabajos Futuros	55
Bibliografía	57

Lista de Figuras

1.	Robot <i>Small Size</i>	17
2.	Representación en 3D de los <i>Small Size STOX's</i> , tomado de [37]	17
3.	Diagrama de la placa de los <i>STOX's</i> , tomado de [38]	18
4.	Diagrama de la disposición de las ruedas en robot tipo <i>Soccer Small Size</i> , tomado de [7]	19
5.	Diagrama de las ruedas en los robot <i>STOX's</i> , tomado de [38]	19
6.	Servidor RF de los <i>STOX's</i>	22
7.	Placa de los <i>STOX's</i>	30
8.	<i>Lighthouse Positioning System</i> , tomado de [76]	31
9.	Imagen de Profundidad tomada con la cámara <i>Intel Realsense D435</i>	31
10.	Proceso de pruebas en profundidad.	32
11.	Detección de contornos.	34
12.	Detección de contornos estables.	34
13.	Conexión Publicador-Suscriptor.	35
14.	Diagrama de la composición del sistema.	36
15.	Flujo grama de rutina.	38
16.	Robots en posición inicial.	38
17.	Robots en posición final.	39
18.	Grupo de robots con Identificador.	40
19.	Publicación Coordenadas e Identificadores.	40
20.	Suscripción Coordenadas e Identificadores.	41
21.	Diagrama de conexión en ROS.	42
22.	Simulación del entorno y los agentes.	43
23.	Diagrama de flujo Algoritmo Multiagente.	45
24.	Diagrama de flujo nodo publicador.	47
25.	Detección con <i>RealSense</i>	47
26.	Detección con cámara web.	48
27.	Diagrama de flujo nodo Suscriptor.	49
28.	Máscara de los contornos en posición final.	50
29.	Mensajes de paro.	50
30.	Comparación de tiempos entre cámaras.	51

Lista de Tablas

1. Comparación de tiempos entre cámaras RealSense y Genérica 51

Glosario

- **CNN:** Convolutional Neural Networks.
- **DBSCAN:** Density-Based Spatial Clustering of Applications with Noise.
- **GED:** Grupo de Estudio y Desarrollo en Robótica.
- **HSV:** Hue, Saturation, Value.
- **IA:** Inteligencia Artificial.
- **MAS:** Multi-Agent Systems.
- **OMR:** Omnidirectional Mobile Robots.
- **RF:** Radio Frecuencia.
- **RGB:** Red, Green, Blue
- **ROS:** Robot Operating System.
- **SSL:** Soccer Small Size League.

Capítulo 1

Resumen

El presente trabajo busca abordar la implementación de rutinas propias de sistemas multiagentes, mediante el sistema operativo de robots en su segunda versión (ROS2) para el equipo STOX's; ROS2 facilita la implementación y desarrollo de sistemas robóticos. Esta es una propuesta innovadora ante los diferentes desafíos a los cuales se enfrenta la robótica, como lo pueden ser la navegación autónoma y la colaboración eficiente entre robots. Por ello, con el respaldo de ROS2, se busca mejorar la coordinación y el rendimiento del grupo de robots en escenarios de prueba, contribuyendo así al avance de este recurso investigativo. Como aportes de este trabajo se tienen el desarrollo de los comportamientos multiagentes para los robots del equipo STOX's y un mecanismo de visión artificial que permite la localización de cada robot-agente desde una perspectiva de cámara lateral en lugar de la perspectiva superior normalmente utilizada.

El equipo STOX's del grupo de robótica de la Universidad Santo Tomás, desarrolló el grupo de robots para la liga *Soccer Small Size League* (SSL) con el cual se participó en la competencia de *RoboCup* durante los años 2011 a 2017 obteniendo muy buenos resultados. Debido al bajo uso del grupo de robots durante un amplio margen de tiempo (desde el 2017 al 2023), se presentan diferentes desafíos a la hora de desarrollar el proyecto. Esto se debe a que varias plataformas robóticas pueden no estar operativas o llegar a requerir mantenimiento. Por lo tanto, se optó por manejar un pequeño número de robots funcionales, los cuales son ideales para afrontar el desarrollo del proyecto debido a su versatilidad y precisión. Esta elección, permitió garantizar la viabilidad del proyecto, aprovechando los recursos disponibles y la validez en la implementación de los algoritmos propuestos.

En un principio, se llevó a cabo una revisión bibliográfica para obtener modelos de algoritmos multiagentes, así como herramientas que permitiesen la detección del grupo de robots en tiempo real y de manera estable. Posteriormente se buscó implementar diferentes técnicas para la detección y localización del grupo de robots de manera que fuese posible probar su desplazamiento en un entorno delimitado y realizar pruebas de forma satisfactoria.

Por último, se trasladaron a ROS2 las funcionalidades correspondientes al movimiento de los robots, así como la detección y localización de estos, de igual manera, se ajustaron los modelos multiagentes previamente probados en simulación, para la comprobación de su funcionamiento en el entorno dado a través de los robots, realizando así el proceso de validación de la propuesta sobre un escenario físico. Como resultado, se logra que los robots realicen movimientos coordinados con base en los algoritmos multiagentes implementados.

Capítulo 2

Abstract

The present work aims to address the implementation of routines inherent to multi-agent systems through the Robot Operating System in its second version (ROS2) for the STOX's team; ROS2 facilitates the implementation and development of robotic systems. This is an innovative proposal in response to the various challenges faced by robotics, such as autonomous navigation and efficient collaboration between robots. Therefore, with the support of ROS2, the aim is to improve the coordination and performance of the group of robots in test scenarios, thus contributing to the advancement of this research resource. Contributions of this work include the development of multi-agent behaviors for the STOX's team robots and a computer vision mechanism that allows the localization of each robot-agent from a side camera perspective instead of the normally used top-down perspective.

The STOX's team, from the Robotics Group of the Universidad Santo Tomás, developed a group of robots for the Soccer Small Size League (SSL), participating in the RoboCup competition from 2011 to 2017 with very good results. Due to the low usage of the group of robots over a long period of time (from 2017 to 2023), various challenges arise in developing the project. This is because several robotic platforms may be non-operational or may require maintenance. Therefore, it was decided to handle a small number of functional robots, which are ideal for tackling the development of the project due to their versatility and precision. This choice allowed for guaranteeing the project's feasibility by taking advantage of the available resources and ensuring the validity in the implementation of the proposed algorithms.

Initially, a literature review was conducted to obtain models of multi-agent algorithms, as well as tools that would allow real-time and stable detection of the group of robots. Subsequently, different techniques were implemented for the detection and localization of the group of robots,

making it possible to test their movement in a delimited environment and to carry out tests satisfactorily.

Finally, the functionalities corresponding to the movement of the robots, as well as their detection and localization, were transferred to ROS2. Likewise, the multi-agent models previously tested in simulation were adjusted to verify their functionality in the given environment through the robots, thus carrying out the validation process of the proposal on a physical scenario. As a result, the robots achieved coordinated movements based on the implemented multi-agent algorithms.

Capítulo 3

Introducción

Según la investigación presentada en [1], la robótica móvil ha visto un continuo avance en el último siglo, llegando a explorar desde autómatas bioinspirados hasta el desarrollo de vehículos autónomos, así como la exploración espacial. Dicho progreso ha visto un gran impacto gracias a la mejora en dispositivos electrónicos, como procesadores, sensores, cámaras, baterías, entre otros dispositivos aplicados en robots. Por otra parte, según [2], es posible decir que los robots móviles, son robots que pueden desplazarse de un lugar a otro de manera autónoma, esto quiere decir que tienen la capacidad de moverse libremente dentro de un espacio de trabajo predefinido para realizar tareas determinadas o alcanzar algún objetivo deseado. La robótica en general y la robótica móvil en particular seguirán su evolución durante los próximos años, impulsadas por avances continuos en áreas como la inteligencia artificial, aprendizaje automático y la integración de demás tecnologías [3].

En años recientes, se han podido llegar a evidenciar avances significativos en sistemas multiagentes, más específicamente en el aprendizaje por refuerzo (RL), registrando el éxito en la resolución de diversos problemas de toma de decisiones secuenciales en el aprendizaje automático [4]. De esta manera, los algoritmos de aprendizaje por refuerzo multiagentes (MARL) se ocupan de sistemas que constan de varios agentes que interactúan dentro de un entorno común. Cada agente toma una decisión en cada paso de tiempo y de esta manera trabajan con los demás agentes para lograr un objetivo predeterminado [5].

Para el cumplimiento de los objetivos planteados en el presente trabajo, se propone en un inicio hacer una revisión de estudios previos de modelos que incluyan MARL, así como herramientas que permitan realizar la detección y localización de los robots en un área delimitada durante su movimiento.

La comprobación de los resultados obtenidos se dará de diferentes maneras, en un primer escenario, se busca dar un proceso de simulación de la técnica multiagente para posteriormente implementarla en el grupo de robots y de esta manera verificar su funcionamiento en el entorno de pruebas.

El presente documento tiene como objetivo la implementación de rutinas en ROS2 para la aplicación de algoritmos multiagentes en el movimiento de un grupo específico de robots, estructurado de manera clara y coherente. Comienza estableciendo una pregunta problemática que fundamenta y orienta el desarrollo del proyecto. Luego, se dan a conocer los objetivos planteados, delineando las metas y tareas necesarias para su realización. La justificación destaca la importancia del proyecto, vinculándose con la pregunta problémica. Para la sección del estado del arte y el marco teórico, se dan a conocer las herramientas, conceptos y definiciones necesarias a la hora de la realización de las tareas. Luego se presenta la metodología dada para la realización del proyecto, donde se expone el paso a paso para el cumplimiento de los objetivos. Además, se presenta el desarrollo que se realizó para la obtención de resultados y la discutiendo sus implicaciones. Finalmente, se exponen las conclusiones basadas en los resultados obtenidos y se sugieren áreas de investigación futura.

3.1. Planteamiento del Problema

Según la federación internacional de robótica (IFR), para febrero de 2023 el stock de robots operativos a nivel internacional, alcanzó la cifra de 3.5 millones de unidades [6]. La creciente cantidad de plataformas demuestra una tendencia global en el área de la robótica, que se aplica en diversos campos, desde la academia hasta el hogar y la industria. En el ámbito de la robótica móvil, se observa un amplio margen de investigación, especialmente en robots autónomos con ruedas, que presentan algunos inconvenientes en su funcionamiento en entornos dinámicos.

Desde su creación, el fútbol robótico ha adquirido relevancia, debido a las características que ofrece, inclusive vistas en problemas reales y complejos, incluyendo el manejo de tecnologías como agentes autónomos, sistemas multiagente, machine learning, visión artificial, entre otras, proporcionado así una descripción para la agrupación de dichas tecnologías capaces de manejar un grupo de robots [7].

Durante el año 2010, la Universidad Santo Tomás desarrolló el equipo de robots *STox's* de la liga *Robo Cup Small Size* con el fin de participar en competencias de robótica. El equipo destacó en diferentes participaciones, siendo la última de ellas en la *Small Size Robot Soccer League* en Japón

en 2017 [8]. Sin embargo, actualmente su uso en el ámbito académico e investigativo se encuentra en pausa. Esto se debe en parte a la exploración en otras áreas de estudio y a que se cuenta con nuevas plataformas robóticas mayormente utilizadas. Por lo tanto, es crucial modernizar estas plataformas con tecnologías de software para actualizar el equipo *STox's Soccer Small Size* de manera que se pueda emplear en nuevas tendencias como las estrategias multiagente.

Hoy por hoy, existen múltiples técnicas que permiten el aprendizaje de los robots autónomos con ruedas en entornos dinámicos, controlados o en variados escenarios de prueba; dichas técnicas son implementadas en diferentes sistemas, uno de ellos es la segunda versión del sistema operativo de robots (ROS2) el cual ha evolucionado respecto a ROS, sistema pionero que estableció una base estándar para la comunicación y utilización de servicios distribuidos entre robots [9]. Por lo tanto, es necesario la realización y evaluación de estudios y tecnologías a través de algoritmos que permitan cumplir con las diferentes exigencias para el grupo robots. Además, es importante considerar las limitaciones de los recursos disponibles. En este sentido, se hace necesario verificar la funcionalidad y limitaciones de los robots previamente diseñados evaluando cada enfoque técnico a partir de las ventajas y desventajas.

De acuerdo con problemática planteada, se plantea la siguiente pregunta de investigación:

¿Como implementar una estrategia multiagente para el conjunto de robots móviles tipo *Soccer Small Size* con los que cuenta la facultad de Ingeniería Electrónica de la Universidad Santo Tomás?

3.2. Objetivos

3.2.1. Objetivo General

Implementar una técnica multiagente para reorganización posicional, a través de la integración de ROS2, para las plataformas robóticas tipo *Soccer Small Size* de la Facultad de Ingeniería Electrónica de la Universidad Santo Tomás.

3.2.2. Objetivos Específicos

- Programar rutinas de control para las plataformas robóticas tipo *Soccer Small Size* de la Universidad Santo Tomás para la aplicación de una técnica multiagente.

- Integrar las rutinas de control a través del entorno de trabajo de ROS2 mediante la definición de nodos, servicios y tópicos a utilizar.
- Aplicar un modelo multiagente en las plataformas tipo Soccer Small Size, basado en la arquitectura de ROS2.
- Validar el funcionamiento del algoritmo multiagente implementado en el equipo de robots *Small Size*, en maniobras y escenarios de prueba.

3.3. Justificación

De acuerdo al planteamiento del problema, es posible evidenciar la necesidad de aportar al desarrollo y actualización del equipo de robots *STox's Soccer Small Size* en software, facilitando su comportamiento en un escenario de prueba, lo cual permite nuevamente el manejo de estas plataformas y así retomar dicho recurso investigativo. El desarrollo y manejo de robots móviles autónomos (por sus siglas en inglés AMR) es esencial además de importante, para lograr un entorno de trabajo flexible, esto debido a que los AMR pueden moverse libremente por una instalación industrial, así como en demás entornos y campos; por ello son cada vez más necesarios para los procesos y tareas comunes las 24 horas del día, los 7 días de la semana [10]. La organización y planificación de rutas es una de las tareas más importantes, así como complejas en la navegación AMR, esto ya que exige que el robot identifique la mejor ruta en función de los criterios de rendimiento deseados, como el margen de seguridad, el tiempo más corto y el consumo de energía [11]. Por ello, la planificación y el control en los ARM son conceptos de suma relevancia, su avanzado hardware y software de control permiten operaciones autónomas en diferentes entornos, actuando de esta manera en operaciones logísticas como fabricación y almacenamiento en plantas industriales, terminales e incluso hospitales [12].

Actualmente, se han propuesto diferentes plataformas de software que presentan características modulares y adaptables que facilitan la construcción de sistemas de robots. La aparición de ROS constituyó un hito importante en tanto este sistema integra funcionalidades que facilitan enormemente el trabajo cooperativo entre robots; sin embargo, su utilización evidencia así mismo algunas falencias y oportunidades de mejora. Por lo mismo, se dio a conocer ROS2, resolviendo muchos de los problemas en la construcción de sistemas robóticos confiables [13]. Por consiguiente, se busca la actualización del grupo de robots *Small Size* a las tecnologías vistas a día de hoy, capaces de movilizarse en tiempo real a través de implementación de un algoritmo multiagente.

Para llevar a cabo el presente trabajo, es esencial considerar diversos puntos, incluyendo el contexto actual del semillero de robótica. Es importante observar que tanto las áreas de investigación, así como el manejo de equipos varían significativamente con el tiempo. Por lo tanto, el uso de las plataformas robóticas se ha visto en pausa en los últimos años, lo que ha afectado su funcionamiento. Factores como el estado de las baterías o del hardware limitan su capacidad de operación y disminuyen la cantidad de robots disponibles, dificultando así las pruebas en entornos dinámicos, controlados o demás escenarios de pruebas. Es fundamental abordar estas limitaciones para mejorar el desempeño de los robots y obtener resultados adecuados en el desarrollo de este proyecto, así como para investigaciones dadas a futuro.

Teniendo en cuenta el contexto y el ambiente local, el impacto visto por este proyecto en la Universidad Santo Tomás, y más específicamente en el grupo de investigación GED, proporcionará el impulso y desarrollo en diferentes líneas de investigación dirigidas hacia los robots autónomos con ruedas, así como las habilidades STEM (Ciencia, Tecnología, Ingeniería y Matemáticas). Una vez realizada la actualización, se prevé que nuevos estudiantes puedan realizar implementaciones sobre las plataformas robóticas, permitiendo avanzar aún más en este campo a futuro.

3.4. Impacto Social

El desarrollo de este proyecto, pretende tener un impacto social positivo en el grupo de investigación GED de la Facultad de Ingeniería Electrónica de la Universidad Santo Tomás, mejorando así varios aspectos de la investigación en robótica como son la visión e inteligencia artificial, inteligencia computacional y también en sistemas multiagente, enriqueciendo estas áreas en un nivel teórico práctico, a través de la recopilación de información, además de las diferentes pruebas y estrategias propuestas para el conjunto de robots.

Considerando ahora el programa de Ingeniería Electrónica de la Universidad Santo Tomás, se espera que el proyecto contribuya positivamente a las diversas áreas académicas del plan de estudios, especialmente aquellas que pertenecen al campo de la Ingeniería Electrónica aplicada.

De igual manera, las demandas en inspección de infraestructuras con robots autónomos han aumentado notablemente y se espera que el mercado crezca en consecuencia. Una de las virtudes, es que los robots autónomos logran navegar por el entorno e interactuar con los humanos en tareas que pueden ser consideradas peligrosas y complejas para el mismo [14]. Por consiguiente, el sector agrícola, es uno de los campos de aplicación para tener en cuenta a la hora

de manejar robots autónomos, ya que la producción agrícola se encuentra bajo una presión constante tanto para aumentar la productividad como para equilibrar la sostenibilidad a largo plazo. En este sentido, debido a la disminución de la oferta de mano de obra calificada, los enfoques agrícolas tradicionales no son suficientes para alcanzar la máxima productividad, sin embargo, los robots y vehículos agrícolas autónomos tienen el potencial de mejorar la eficiencia de la producción agrícola y reducir el consumo de recursos [15].

Además, también es posible mencionar al sector de la salud como uno de los campos con mayor impacto en los robots autónomos. Por ejemplo, los sistemas robóticos, reducen significativamente el riesgo de transmisión de enfermedades infecciosas a los trabajadores de atención médica de primera línea, al permitir clasificar, evaluar, monitorear y tratar a los pacientes desde una distancia segura [16].

Capítulo 4

Estado del Arte

4.1. *Small Size Robot Team*

Los partidos de fútbol entre robots, constituyen diferentes actividades de experimentación y testeo para el desarrollo de robots autónomos que llegan a cooperar entre si con el fin de realizar una tarea en específico [7]. El fútbol de robots se presenta como un entorno multiagente en un ambiente hostil, donde los robots operan de modo que buscan ‘marcar goles’ a través de la cooperación entre los robots del equipo y enfrentando las estrategias de los robots del equipo rival. En este ámbito, se presentan diferentes roles que corresponden a la combinación de los comportamientos de los robots junto a su ubicación en el campo de juego. Los comportamientos dados por cada plataforma corresponden a los movimientos que realizan los robots con el fin de realizar su rol, como ir por el balón, bloquear un oponente o realizar un pase [17].

La idea de utilizar el fútbol como banco de pruebas para aplicaciones de IA y robótica tiene su origen en el artículo de Alan Mackworth, 1992. Inmediatamente después, un grupo de investigadores empezó a dar forma a la estructura de la *RoboCup*, cuya primera competición oficial se celebró en 1997 [18]. En 2001, la introducción de la categoría *Small Size* en el *RoboCup*, con robots de menos de 15 cm de diámetro, marcó un hito al facilitar la participación de equipos con recursos limitados y fomentar la innovación tecnológica. Para 2005, estos equipos comenzaron a mostrar avances significativos en la coordinación y el comportamiento autónomo, aunque persistían desafíos en la percepción del entorno y la toma de decisiones en tiempo real [19].

El fútbol de robots en la categoría *Small Size* ha evolucionado significativamente con diversas innovaciones que han transformado la competencia con el paso del tiempo. En 2015, se introdujo el juego en campos de tamaño doble, lo que representó un desafío adicional en términos de

estrategia y coordinación. Con el transcurso de los años, los equipos de *Small Size* continuaron incorporando tecnologías y algoritmos de vanguardia, reflejando un progreso constante y sostenido [20]. En 2023, esta evolución persistió, con mejoras continuas que han elevado el nivel de autonomía e inteligencia de los robots, consolidando así el papel del fútbol robótico como un campo de investigación y desarrollo clave en la robótica y la inteligencia artificial.

4.2. Inteligencia Artificial

La inteligencia artificial (IA) se refiere a la capacidad de las máquinas para simular la inteligencia humana, abarcando procesos como el aprendizaje, el razonamiento, la resolución de problemas y la toma de decisiones [21]. A lo largo de las décadas, la IA ha experimentado una evolución notable, caracterizada por periodos de estancamiento y resurgimiento. El resurgimiento de la investigación en IA ha sido impulsado por avances en la capacidad de computación, la creación de nuevos algoritmos y la disponibilidad masiva de datos. Este renacimiento se ha materializado a través del aprendizaje automático y las redes neuronales artificiales, que se han convertido en herramientas esenciales para el análisis de datos y la toma de decisiones [22].

Entre 2006 y 2010, la explosión del *Big Data* marcó un hito significativo al facilitar el desarrollo de técnicas de análisis más avanzadas y el surgimiento del *Deep Learning* [23]. Este campo del aprendizaje automático permitió a las máquinas aprender de grandes volúmenes de datos, fortaleciendo el interés en las redes neuronales aplicadas a la visión artificial y al procesamiento del lenguaje natural. Simultáneamente, la IA se volvió fundamental en el avance de vehículos autónomos, mejorando la percepción del entorno y las capacidades de control [24].

Entre 2016 y 2020, los asistentes de voz se popularizaron, integrando la IA en la vida cotidiana y transformando sectores como la salud y el comercio con automatización y decisiones más inteligentes. Este período también destacó los debates sobre los impactos éticos de la IA [25]. Desde 2021, la IA generativa ha desencadenado nuevas formas de creatividad, mientras que el aprendizaje por refuerzo ha optimizado sistemas complejos. Además, la IA ha desempeñado un papel crucial en la lucha contra el cambio climático mediante el análisis de datos y el desarrollo de tecnologías sostenibles [26].

4.3. Visión Artificial

La visión artificial, es aquella materia que busca principalmente proporcionar ojos a un equipo a través de diferentes técnicas, como la interpretación de imágenes y la distinción de objetos, permitiendo la extracción de información para la resolución de tareas [27]. Entre 2005 y 2010, la integración del aprendizaje automático en la Visión Artificial permitió un entrenamiento más eficiente y adaptable, mejorando la segmentación de imágenes y desarrollando sistemas para reconocer acciones humanas en videos, lo que amplió el alcance de esta tecnología en diversas aplicaciones.

El surgimiento del *Deep Learning* revolucionó la visión artificial, con el uso de redes neuronales como las *Convolutional Neural Networks* (CNN), se lograron resultados sin precedentes en tareas como el reconocimiento de objetos, la detección de rostros y la segmentación de imágenes. Además, hubo avances significativos en la visión computacional 3D, permitiendo la reconstrucción tridimensional de objetos y escenas. La integración de la Visión Artificial en la realidad aumentada permitió superponer información digital sobre el mundo real, abriendo nuevas posibilidades en esta tecnología. [28].

Investigaciones tales como [29] y [30], realizadas entre 2015 y 2020, demuestran que la implementación del aprendizaje por refuerzo permitió a los sistemas aprender a realizar tareas de manera autónoma. Esta tecnología se integró en dispositivos móviles, como *smartphones* y *tablets*, para aplicaciones como la realidad aumentada, el reconocimiento de imágenes y la traducción de idiomas. Además, se desarrollaron sistemas de visión artificial para la percepción del entorno y el control de vehículos autónomos, marcando un hito en el desarrollo de tecnologías de conducción autónoma [31].

4.4. Sistemas Multiagentes

Un sistema multiagente, es aquel sistema que se compone de varios agentes autónomos que interactúan entre sí y también con su entorno para lograr un objetivo de manera individual y colectiva. Es posible definir a un agente como aquella entidad que se sitúa en un entorno o ambiente, detectando diferentes parámetros, así como la toma de decisiones, basando su comportamiento en los objetivos previamente definidos. Los agentes también interactúan con otros agentes cercanos o con el entorno para aprender del contexto y acciones que se realizan [32].

Desde el año 2000, la evolución de los sistemas multiagentes ha sido notable, ya que se establecieron las bases teóricas para la arquitectura y diseño de estos sistemas, definiendo conceptos fundamentales como los agentes, la interacción y la coordinación. Durante la década de 2010, hubo avances significativos en la comunicación y negociación entre agentes, profundizando en los mecanismos de comunicación mediante protocolos, lenguajes y estrategias de negociación. Se integraron técnicas de inteligencia artificial como el aprendizaje automático y el razonamiento para mejorar la toma de decisiones de los agentes, explorando aplicaciones en entornos dinámicos como la gestión de tráfico y la respuesta a errores [33].

En años más recientes, la integración de la inteligencia artificial y el aprendizaje por refuerzo ha potenciado significativamente las capacidades de los sistemas, permitiendo el análisis de grandes volúmenes de datos y la toma de decisiones más precisas. Estos avances han demostrado aplicaciones exitosas en campos como la robótica y la conducción autónoma, respaldados por el uso de redes neuronales. Además, destacan la participación activa de sistemas multiagente en la toma de decisiones y el aprendizaje entre agentes, los cuales pueden interactuar y colaborar dentro de un entorno común [4].

4.5. Sistema Operativo de Robots

La operación de los robots autónomos y cooperativos se está convirtiendo cada vez más en una parte fundamental de los sistemas de automatización inteligentes en la industria moderna. El desarrollo, así como el control, difieren de los métodos de automatización tradicionales, generando desafíos cada vez mas complejos en diferentes campos. El sistema operativo de robots, o por sus siglas en ingles (ROS) proporciona actualmente una gran variedad de herramientas, así como utilidades que pueden colaborar en gran medida a esos desarrollos [34].

Desde la encarnación de ROS en 2007, la cantidad de usuarios de ROS ha experimentado un aumento importante con más de 16 millones de descargas totales en el año 2018. Es posible definir a ROS como colección de herramientas y bibliotecas que ayuda al desarrollo de software de robótica. El software está escrito en forma de nodos que intercambian mensajes usando una capa de transporte basada en *sockets TCP/IP* estándar [35].

El surgimiento de ROS presentó un gran potencial para superar las barreras del ecosistema de robots. Este sistema fue diseñado para permitir y acelerar el desarrollo de software y la integración de sistemas de robots colaborativos e inteligentes a través de una metodología de diseño modular. Con el tiempo, la primera versión de ROS se vio limitada por diferentes factores, incluyendo la evolución de los sistemas. Como resultado, fue reemplazado por ROS2, obteniendo

principalmente interfaz de *middleware* abstracta que mejora la comunicación entre dispositivos, componentes y módulos [36].

Las dos versiones del sistema operativo para robots, Robot Operating System (ROS), presentan varias diferencias significativas. Entre ellas se encuentran la compatibilidad multiplataforma, ya que ROS2 ha sido diseñado para funcionar en una amplia gama de sistemas operativos, lo que aumenta su versatilidad en comparación con ROS1. Además, ROS2 ofrece una comunicación más flexible y confiable, gracias a su compatibilidad con protocolos de red avanzados como DDS (Data Distribution Service). Además de ello, también se ha mejorado en temas de seguridad, ya que permite la autenticación y el cifrado de mensajes, lo que garantiza una comunicación más segura entre los nodos del sistema, ofreciendo así, una solución más confiable como versátil a la hora de desarrollar aplicaciones sobre un conjunto o grupo de plataformas robóticas [9].

Capítulo 5

Marco Teórico

Para dar inicio al proyecto, es fundamental llegar a establecer una base de conocimiento sólida. Por lo mismo, se llevará a cabo una breve introducción conceptual y teórica sobre los principales temas que se abordarán. De esta manera, será posible comprender mejor el proceso llevado a cabo durante el desarrollo del proyecto.

5.1. *STOx's Soccer Small Size*

A la hora de manejar un grupo de robots, es necesario comprender diferentes parámetros de su funcionamiento, el entendimiento de factores como su construcción mecánica, así como el software son necesarios a la hora de realizar trabajos sobre las plataformas robóticas. De igual manera, para llegar a construir un equipo capaz de competir en la *Small Size Robot League*, es necesaria la investigación multidisciplinaria en diferentes campos como el desarrollo de hardware robótico, aprendizaje automático, visión por computadora, mecánica, entre otros que permiten observar el rendimiento y eficiencia de los robots en diversas aplicaciones [7]. Según [37] el grupo de robots STOx's ha participado en diferentes ediciones de la *Small Size Robot League*, logrando diferentes hitos, así como resultados destacados desde 2011 hasta 2017.

5.1.1. Estructura Mecánica y Electrónica

El grupo de robots, más específicamente los *Small Size STOx's*, ha pasado por diferentes generaciones, lo cual está relacionado con el respectivo diseño del hardware que presentan, así como con las mejoras a nivel mecánico, electrónico y de comunicaciones que se han implementado en



FIGURA 1: Robot *Small Size*

cada versión. Según [38] los robots se encuentran actualmente en su tercera generación, adquiriendo mejoras a través de los eventos en los cuales se participaron, observado cambios en el chasis, ruedas, motores, sistema de regate y el pateador, vistos en la figura 1. De igual manera, el presente trabajo se desarrolla con esta generación del grupo de robots, con las características que allí se mencionan.

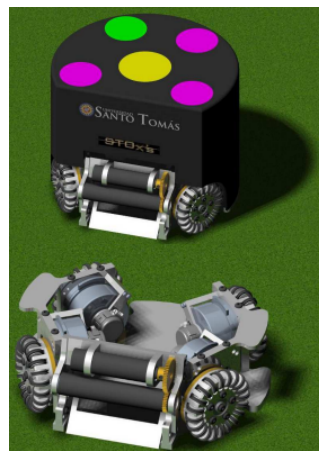


FIGURA 2: Representación en 3D de los *Small Size STOx's*, tomado de [37]

Al igual que en la parte, mecánica, la tercera generación para la parte electrónica incluyó actualizaciones. Para este caso, se diseñó una placa electrónica que alberga todos los dispositivos eléctricos y de control del robot como se observa en la figura 2, agregando cambios que brindan confiabilidad a las funcionalidades [38].

Asimismo, la placa principal se observa en la figura 3, esta integra en primer lugar los controladores del motor, los cuales están implementados como inversores trifásicos. En segundo lugar,

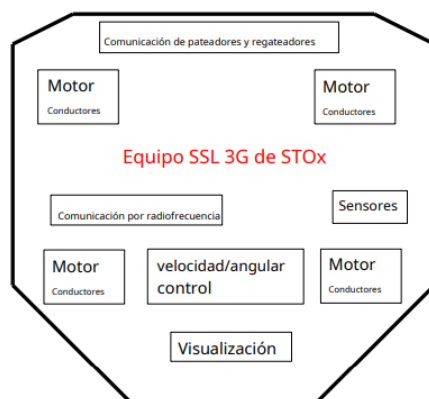


FIGURA 3: Diagrama de la placa de los STOX's, tomado de [38]

cuenta con un circuito de adquisición de sensores y acondicionamiento de señal. En tercer lugar, incluye un módulo de visualización destinado a fines de depuración [37].

5.1.2. Modelos Omnidireccionales

El control del movimiento en entornos dinámicos es uno de los problemas más importantes al utilizar robots móviles en colaboración con otros robots. Con el avance de la tecnología y la invención de nuevas ruedas con estructuras especiales, los robots móviles omnidireccionales (OMR) se han presentado y utilizado ampliamente en aplicaciones industriales, exploración espacial, campos médicos, entre otros [39]. A diferencia de los robots móviles convencionales que utilizan dos ruedas motrices para controlar el movimiento del robot en direcciones restringidas, los OMR no están limitados por restricciones y pueden moverse en cualquier dirección desde cualquier configuración [40]. Esta característica aumenta la maniobrabilidad, rapidez y precisión del robot móvil, lo que hace que el robot sea capaz de trabajar en espacios de trabajo pequeños y entornos dinámicos con alta precisión [41].

Los robots con tracción omnidireccional pueden diseñarse con tres o más ruedas, pero los *Small Size STOX's* utilizan cuatro ruedas. La elección de cuatro ruedas tiene ventajas significativas, esto debido a que ofrece una mayor aceleración. Además, con tres ruedas, el movimiento en ciertas direcciones es menos eficiente y presenta limitaciones en la colocación de ruedas y otros dispositivos periféricos [7]. Además de ello, se han desarrollado y producido muchos tipos de robots móviles omnidireccionales. A la hora de hablar de las ruedas que poseen dichos robots, es posible observar dos tipos típicos de ruedas que pueden realizar el movimiento omnidireccional, es decir, una rueda orientable y una rueda omnidireccional [39].

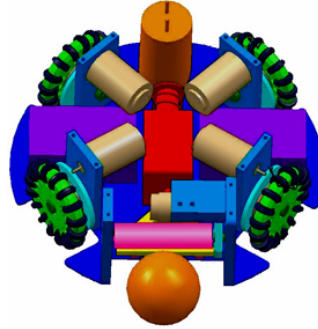


FIGURA 4: Diagrama de la disposición de las ruedas en robot tipo *Soccer Small Size*, tomado de [7]

En el caso de los STOX's, como se aprecia en la figura 4, el modelo omnidireccional permite que el robot se traslade entre dos puntos del plano definido, sin tener la necesidad de girar primero su cuerpo hacia la dirección objetivo. Según [7], para conseguirlo se emplean ruedas especiales, ensambladas con ruedas más pequeñas unidas a la periferia de la principal. Estas ruedas más pequeñas están ensambladas en una posición que les permite girar sin fricción en una dirección distinta a la rueda principal, conectada al eje del motor, observadas en la figura 5.



FIGURA 5: Diagrama de las ruedas en los robot *STOX's*, tomado de [38]

El modelo omnidireccional de un robot de fútbol se describe matemáticamente mediante una serie de matrices que relacionan las fuerzas y momentos aplicados al robot con su movimiento resultante.

5.1.2.1. Matriz de Fuerzas y Aceleración

La ecuación 5.1 establece la relación entre las fuerzas generadas por los motores de las ruedas y las aceleraciones resultantes en el robot. Esta ecuación se expresa de la siguiente manera:

$$\begin{bmatrix} a_x \\ a_y \\ R\dot{\omega} \end{bmatrix} = \frac{1}{M} \mathbf{C} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \quad (5.1)$$

Aquí, a_x y a_y son las aceleraciones del robot en las direcciones x y y , respectivamente, mientras que $R\dot{\omega}$ es la aceleración angular. La masa del robot está representada por M . La matriz $\mathbf{C}$ es fundamental en esta ecuación, ya que transforma las fuerzas de los motores (f_1, f_2, f_3, f_4) en aceleraciones. Esta matriz depende de los ángulos de las ruedas (ϕ_n) y se define como:

$$\mathbf{C} = \begin{bmatrix} -\sin(\phi_1) & -\sin(\phi_2) & \sin(\phi_3) & \sin(\phi_4) \\ \cos(\phi_1) & -\cos(\phi_2) & -\cos(\phi_3) & \cos(\phi_4) \\ \frac{1}{\alpha_1} & \frac{1}{\alpha_1} & \frac{1}{\alpha_1} & \frac{1}{\alpha_1} \end{bmatrix} \quad (5.2)$$

Los términos $\sin(\phi_n)$ y $\cos(\phi_n)$ representan las contribuciones de cada rueda a las componentes de aceleración en los ejes x y y , y el término $\frac{1}{\alpha_1}$ indica cómo la masa del robot está distribuida con respecto a cada rueda.

5.1.2.2. Matriz de Velocidad

La ecuación 5.3 relaciona las velocidades de los motores con las velocidades del cuerpo del robot. Esta ecuación se expresa de la siguiente manera:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \mathbf{D} \begin{bmatrix} v_x \\ v_y \\ R\omega \end{bmatrix} \quad (5.3)$$

En esta ecuación, v_1, v_2, v_3, v_4 son las velocidades de las ruedas, mientras que v_x y v_y son las velocidades del robot en los ejes x y y . La velocidad angular está representada por $R\omega$. La matriz $\mathbf{D}$ acopla las velocidades de las ruedas con las velocidades del cuerpo del robot y se define como:

$$\mathbf{D} = \begin{bmatrix} -\sin(\phi_1) & \cos(\phi_1) & 1 \\ -\sin(\phi_2) & -\cos(\phi_2) & 1 \\ \sin(\phi_3) & -\cos(\phi_3) & 1 \\ \sin(\phi_4) & \cos(\phi_4) & 1 \end{bmatrix} \quad (5.4)$$

Los términos $\sin(\phi_n)$ y $\cos(\phi_n)$ en esta matriz indican cómo las velocidades de las ruedas se combinan para generar las velocidades en los ejes x y y del robot. El término 1 en la tercera columna de la matriz muestra la relación directa entre las velocidades angulares de las ruedas y la velocidad angular del robot.

5.1.3. Comunicación mediante Radio Frecuencia

Los enlaces de comunicación en espacio libre desempeñan un papel crucial en diversas aplicaciones de transferencia de datos, utilizando tanto ondas ópticas como de radiofrecuencia (RF). En el pasado, la capacidad de estos sistemas se incrementaba significativamente mediante la multiplexación y transmisión simultánea de múltiples flujos de datos independientes [42]. Según [43], los enlaces RF convencionales pueden emplear la técnica bien establecida de múltiples entradas y múltiples salidas, utilizando elementos de antena separados espacialmente tanto en el transmisor como en el receptor para la transmisión de canales de datos. Además, las comunicaciones por RF con ubicaciones fijas de transmisor y receptor están ganando importancia debido a su amplio potencial en diversas aplicaciones [44].

La comunicación mediante radiofrecuencia, en este escenario llega a jugar un papel fundamental para el funcionamiento del grupo robótico, esto debido a que permiten la transferencia de datos de forma inalámbrica entre los robots y una tarjeta de control a través de un identificador. De igual manera, la tecnología proporciona de forma confiable el intercambio de información en tiempo real, facilitando la coordinación entre las plataformas para la realización de tareas.

En cuanto al sistema de comunicación de los *STOx's*, como se señala en [38], consta de un servidor RF y un módulo de comunicación integrado en cada robot, basados en el chip nRF24L01, como se aprecia en la figura 6. El servidor RF actúa como intermediario entre el módulo de control y los robots, recibiendo comandos del primero y transmitiéndolos a través de canales Tx dedicados a cada robot. Del mismo modo, recibe datos de los robots a través de canales Rx separados.

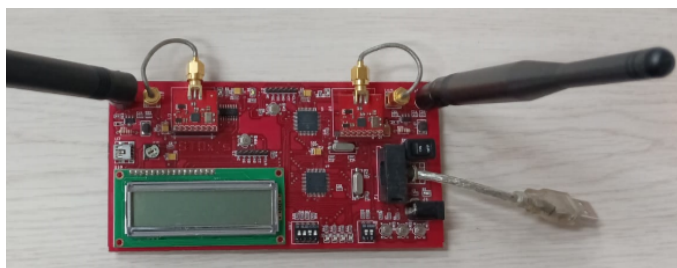


FIGURA 6: Servidor RF de los *STOX*'s

5.2. Sistemas Multiagente

Los sistemas multiagentes (MAS) se pueden describir como sistemas distribuidos con dos o más agentes inteligentes. A diferencia de un solo agente, los agentes en MAS pueden resolver cooperativamente problemas complicados con buena eficiencia, solidez y confiabilidad [45]. Si bien existe una literatura limitada que se ocupa de los problemas de seguridad del control de formación en los sistemas multiagentes, en los últimos años según [46], el control de formación en estos sistemas se usa de manera diversa en diferentes campos, como la robótica, vehículos aéreos, de superficie e inclusive satélites.

Los sistemas multiagente se han empleado en una amplia gama de aplicaciones, desde simulaciones como detección y seguimiento de objetos, hasta sistemas del mundo real como atención médica y vehículos aéreos no tripulados. Según [47], la integración de redes de sensores inalámbricos con sistemas multiagente ha dado lugar a nuevas aplicaciones, como los robots móviles. Sin embargo, este uso extensivo ha generado diversos desafíos para los investigadores, incluyendo preocupaciones de seguridad, limitaciones de recursos y restricciones de tiempo [48].

5.2.1. Consenso

El concepto de consenso, donde todos los agentes convergen hacia un acuerdo o interés común, es fundamental en el control coordinado de estos sistemas multiagente, atrayendo así la atención en perspectivas teóricas y prácticas.

En las últimas dos décadas, la investigación acerca de la coordinación de múltiples agentes ha sido objeto de contribuciones periódicas por parte de diferentes profesionales como matemáticos, físicos e ingenieros [49]. De igual manera, la coordinación de intereses entre múltiples agentes que incluyen el consenso, ha sido un área de enfoque destacada en este campo [50].

Debido a la estructura de red de los sistemas, cada agente utiliza la información de los vecinos como recursos de entrada de control para lograr un rendimiento general como grupo mediante una topología de comunicación [51]. De esta manera, el problema del consenso es fundamental en el control coordinado de los sistemas multiagentes, siendo de suma importancia en aplicaciones como el control operativo y el diseño de redes y sensores [52]. Este problema, se ha convertido en un tema destacado de investigación en los sistemas multiagentes, empleando inicialmente simulaciones computacionales para modelar el comportamiento en grupos biológicos, como aves y peces [48].

Los problemas de control cooperativo implican que varios actores autónomos busquen lograr colectivamente un objetivo global. Según [53], el problema del consenso es un ejemplo de un problema de control cooperativo, donde el objetivo global es que todos los actores lleguen a un consenso sobre un estado determinado. En el campo de la robótica, las plataformas individuales están limitadas a resolver problemas en un solo dominio y carecen de capacidades para abordar fallos de manera autónoma, lo que incrementa los costos de desarrollo. Por el contrario, un sistema robótico multiagente ofrece estrategias tolerantes a fallos, permitiendo que otros agentes sustituyan a los que fallen [54]. Además, los sistemas robóticos multiagentes presentan ventajas de adaptabilidad y procesamiento en entornos complejos de manera efectiva y eficiente, distribuyendo así la carga de trabajo entre los agentes para cubrir el escenario en un menor tiempo en comparación con una sola unidad o plataforma [55].

5.2.1.1. Juegos Potenciales

Uno de los problemas más importantes y que han llegado a captar la atención de investigadores, es la capacidad de los robots para navegar y comprender entornos desconocidos. Este reto es de gran importancia en la robótica móvil y de igual manera continúa siendo un área de investigación activa, así como de interés en la comunidad científica [56].

Investigaciones han llegado a explorar el papel de una clase especial de juegos para el control cooperativo, más específicamente los juegos potenciales. Para estos juegos, una condición especial de alineación de las funciones de utilidad individuales y la función objetivo global garantizando la existencia de al menos un punto donde se cumplen las metas generales del sistema [57].

Al hablar de juegos potenciales, es esencial referirse a la teoría de juegos, específicamente al concepto de equilibrio de Nash. El equilibrio de Nash describe una situación en la que cada jugador en el juego elige la mejor opción posible, dadas las decisiones de los demás jugadores.

En este estado, ningún jugador tiene incentivos para cambiar su estrategia, ya que cualquier cambio no mejoraría su situación individual, lo que resulta en una estabilidad donde ninguna persona puede mejorar su posición sin empeorar la de los demás [58].

En este caso, el desafío del diseño de control en el enfoque de la teoría de juegos radica en diseñar tanto las funciones objetivo del jugador como la dinámica de aprendizaje, de modo que los jugadores logren colectivamente el objetivo del planificador global [53]. Por otro lado, trabajos recientes en aprendizaje por refuerzo con un solo agente han permitido el aprendizaje en dominios que antes se consideraban demasiado desafiantes debido a sus grandes y complejos espacios de observación [59].

5.2.1.2. Aprendizaje por Refuerzo

El aprendizaje por refuerzo ha demostrado ser prometedor al permitir que los robots físicos aprendan habilidades complejas en el mundo real. Al mismo tiempo, la robótica proporciona un dominio atractivo para evaluar tales algoritmos, ya que refleja cómo los humanos aprenden al interactuar directamente con el mundo real como agentes encarnados. [60]. Los robots en los entornos de fabricación actuales suelen realizar tareas repetitivas y, a menudo, carecen de la capacidad de manejar la variabilidad y la incertidumbre. De igual manera, Aprender a percibir y moverse en el mundo real presenta numerosos desafíos, algunos de los cuales son más fáciles de abordar que otros, y algunos de los cuales a menudo no se consideran en la investigación del aprendizaje por refuerzo, ya que se centran mayormente en dominios simulados [61].

Según [62], los datos necesarios para el aprendizaje de los robots pueden generarse mediante la interacción entre el robot y el entorno o ser proporcionados por expertos. Esto permite un marco para diseñar comportamientos complejos, aprovechando los desafíos y la validación que ofrece el campo de la robótica [63]. Asimismo, a través del aprendizaje por refuerzo, los robots aprenden a lograr un objetivo en un entorno incierto y potencialmente complejo, permitiéndoles adaptarse dinámicamente a situaciones cambiantes y resolver problemas de manera eficiente [64].

5.3. Visión Artificial

Según [65], la visión artificial ha llegado a ganar relevancia y sus técnicas han ido evolucionando en las últimas tres décadas. Esta área, sigue siendo de gran interés para los investigadores

debido al avance tecnológico, facilitando así el desarrollo de nuevas técnicas de visión e inteligencia artificial.

Las técnicas de visión por computadora han desempeñado un papel fundamental en la promoción de la informatización, digitalización y mejora de la inteligencia en los sistemas de fabricación industrial [66]. Este campo, considerado como un pilar de la inteligencia artificial, ha experimentado un rápido desarrollo en los últimos años, especialmente con el avance del aprendizaje profundo [67].

La visión por computadora es un campo que ha contribuido de manera significativa al progreso académico y al desarrollo de la inteligencia artificial al procesar imágenes capturadas con cámaras [68]. Desde su inicio, este campo ha experimentado un rápido avance, especialmente con el desarrollo del aprendizaje profundo, lo que ha mejorado notablemente la precisión en el reconocimiento de imágenes [69]. Desde la perspectiva de los escenarios de aplicación, las aplicaciones industriales de las tecnologías de visión por computadora se pueden clasificar en las siguientes: inspección visual, control de procesos, identificación de piezas y mecanismos robóticos de control [66].

Actualmente, en la robótica moderna la localización precisa y la navegación son aspectos fundamentales para la mayoría de las aplicaciones de robots móviles. En las últimas décadas, se ha dedicado mucha atención al desarrollo de tecnologías y métodos innovadores para mejorar la localización y el posicionamiento de los robots [70]. Una localización precisa permite a los sistemas autónomos navegar de manera efectiva en entornos conocidos y desconocidos, lo que influye significativamente en el éxito de tareas o la reducción de los fallos [71].

El sistema de visión en un grupo de robots es de suma relevancia a la hora de manejar un grupo de robots, esto se debe a que proporciona información fundamental sobre la ubicación, así como el entorno en el que se encuentran los robots en tiempo real. Esta información es crucial a la hora de realizar tareas de navegación, planificación de rutas o la coordinación entre los robots. Además, es importante destacar que existen diferentes enfoques para la implementación del sistema de visión, como lo pueden ser la segmentación o visión por color y la visión en profundidad.

5.3.1. Segmentación

En el ámbito de la visión computacional, la segmentación surge como una técnica para identificar y delimitar áreas de interés en imágenes o secuencias de imágenes. Dicha técnica consiste en

separar la imagen en regiones significativas, lo que permite un análisis más preciso del contenido visual. A través de la segmentación, es posible hacer separación de objetos, identificación de contornos e inclusive el reconocimiento de patrones, facilitando así tareas desde la detección de objetos hasta el análisis en imágenes médicas.

En la literatura se han desarrollado varios algoritmos para la segmentación de imágenes. Recientemente, debido al éxito de los modelos de aprendizaje profundo en una amplia gama de aplicaciones de visión, ha habido una cantidad sustancial de trabajos destinados a desarrollar enfoques de segmentación de imágenes utilizando modelos de aprendizaje profundo [72].

La segmentación de imágenes implica la clasificación de píxeles con etiquetas semánticas o la partición de objetos individuales. En el caso de la segmentación semántica, se asigna una etiqueta a cada píxel de la imagen para identificar el tipo de objeto al que pertenece (por ejemplo, humano, automóvil, árbol, cielo) para todos los píxeles de la imagen, por lo que generalmente es una tarea más difícil que la clasificación de imágenes, que predice una única etiqueta para toda la imagen [73]. Mientras tanto, en la segmentación de instancias, se busca separar cada objeto individualmente en la imagen, permitiendo distinguir entre diferentes instancias de un mismo tipo de objeto, ampliando el alcance de la segmentación semántica al detectar y delimitar cada objeto de interés en la imagen (por ejemplo, dividir a personas individuales) [74].

En los últimos años, se han desarrollado diversos métodos para el procesamiento de imágenes en escala de grises y color. En la literatura, se destacan varios espacios de color para el procesamiento de imágenes en color, como RGB y HSV (matiz, saturación, valor). Mientras que el espacio RGB es ampliamente utilizado para la visualización del color, como en sistemas de televisión y adquisición de imágenes, no resulta idóneo para la segmentación o el procesamiento de colores debido a la alta correlación entre sus componentes R, G y B [75].

Capítulo 6

Diseño Metodológico

6.1. Fases y Metodología

Para el desarrollo del presente proyecto y cumplir con los objetivos planteados, se propone llevar a cabo diferentes etapas las cuales se explican a continuación.

6.2. Fase 1: Inspección y revisión bibliográfica

En esta etapa, se lleva a cabo la respectiva investigación, así como la estructuración y redacción del documento, acorde a los diferentes temas de interés propuestos para el desarrollo del proyecto, tales como la inteligencia y visión artificial, los sistemas multiagente y las diferentes técnicas de desarrollo en el sistema operativo de robots.

6.2.1. Adecuación de las plataformas robóticas

Se identifican las diferentes limitaciones y condiciones físicas presentadas en el equipo de robots con el fin de mejorar y adecuar los mismos a nuevas tecnologías, para la habilitación de su funcionamiento.

6.3. Fase 2: Actualización a ROS2

Una vez habilitado el funcionamiento de las plataformas robóticas, se actualizarán las diferentes funcionalidades que presentan dichos equipos mediante la integración de ROS2.

6.4. Fase 3: Implementación de la estrategia multiagente en las plataformas robóticas

Tras la actualización en su sistema, así como la investigación y revisión bibliográfica, se implementará un algoritmo multiagente en las plataformas robóticas con el fin de mejorar su maniobrabilidad en escenarios de prueba.

6.5. Fase 4: Validación de las técnicas implementadas en el sistema.

Por último, para esta etapa se llevarán a cabo diferentes pruebas técnicas y físicas, evaluando las facultades y el comportamiento del equipo de robots, con la finalidad de detectar posibles fallos en el sistema, para luego valorar los ajustes y modificaciones a realizar en el sistema.

Capítulo 7

Desarrollo Conceptual

7.1. Fase 1

7.1.1. Selección de Plataformas

Inicialmente, durante el proceso del desarrollo y para dar cumplimiento a los objetivos planteados, fue crucial la selección de las plataformas adecuadas, así como la comprensión de su funcionamiento, para la validación de la operatividad en cada robot. Sin embargo, debido al prolongado periodo sin uso, muchos de los equipos se encontraban en condiciones deficientes, observando acumulación de polvo, marcas que indicaban fallos en algunos robots, falta de baterías, entre otros problemas. Estos factores, limitaron el numero de robots disponibles para su utilización.

Para dar inicio al proceso de selección, se procedió a cargar las baterías disponibles para posteriormente conectarlos. De los 11 robots disponibles, todos encendieron sin llegar a presentar ningún inconveniente, lo que permitió un inicio sin contratiempos para la evaluación de su operatividad y selección posterior. Después de esto, se llevó a cabo una prueba básica utilizando un código específico que establece la comunicación con un solo robot, esto se dio por medio del servido RF de los Small Size. Para seleccionar el robot con el que se comunica, se modificó el identificador utilizando el switch ubicado en el lado izquierdo del display como se observa en la figura 7.

Este código permitió determinar que los *Small Size* funcionaban correctamente. De los 11 robots disponibles, 5 realizaron las acciones programadas. Además, se identificó que estos robots tienen tres tipos de movimiento: horizontal (eje X), vertical (eje Y) y rotacional (eje W). Para

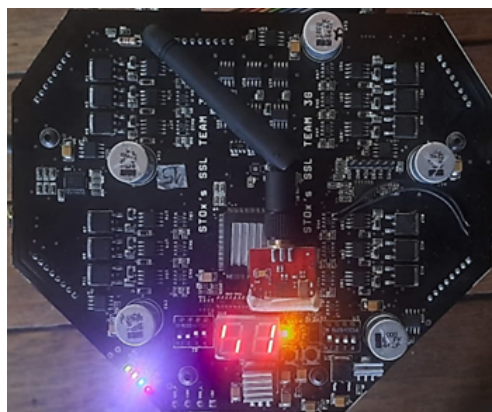


FIGURA 7: Placa de los STox's

comprender mejor su funcionamiento, se realizaron los cambios necesarios en los identificadores de tres robots y se agregaron los nuevos identificadores al código para llevar a cabo una prueba conjunta. Durante esta prueba, los robots se desplazaron juntos en cada uno de los ejes. Finalmente, se programó una pequeña formación en la que los robots se alineaban en fila, luego formaban un triángulo y finalmente regresaban a su posición inicial. Es importante destacar que los movimientos en cada eje están definidos por un valor flotante entre -15.0 y 15.0, donde el signo indica la dirección y el valor representa la velocidad. Para evitar daños tanto al entorno de trabajo como a los propios *Small Size*, se limitó la velocidad a un valor que no superara 1.0 debido a la fuerza considerable de los motores y el riesgo de errores en estas velocidades.

7.1.2. Localización Mediante Sistema Bitcraze

Una vez determinado el grupo de robots, se planteó la necesidad de desarrollar una técnica que permitiese la localización de los robots en el entorno de trabajo, para lo cual, en un principio, se propuso el sistema *CrazyFly*, el cual consta de 2 estaciones y un conjunto de drones vistas en la figura 8, lo cual permitía estimar una geometría en un área definida.

El *lighthouse positioning system*, ofrece una precisión similar a la de un sistema de captura de movimiento, pero a un costo mucho menor y con la ventaja de obtener la posición directamente en el dispositivo seguido, en lugar de depender de la infraestructura. Esto resulta especialmente beneficioso, ya que proporciona información de posición sin requerir una transmisión de posición de baja latencia por radio [76].

Estos drones se ponían estratégicamente en la parte superior de cada robot, lo que permitía adquirir las coordenadas de cada uno de ellos a partir de la geometría previamente definida. Con este sistema en funcionamiento, se logró realizar con éxito, pruebas de movimiento entre dos

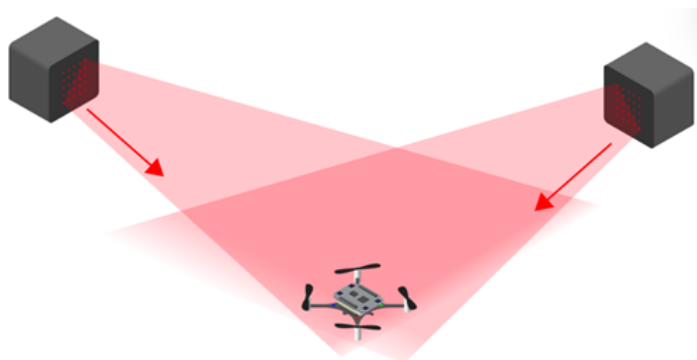


FIGURA 8: *Lighthouse Positioning System*, tomado de [76]

puntos de coordenadas con un solo robot. Sin embargo, un contratiempo surgió cuando una de las estaciones sufrió daños, lo que impidió continuar utilizando el sistema de posicionamiento proporcionado por *bitcraze*.

7.1.3. Detección por Profundidad

Ante la avería del sistema de posicionamiento de *bitcraze*, surgió la necesidad de encontrar una alternativa viable para el manejo de los robots. Para ello, se exploró la opción de utilizar la cámara *Intel RealSense D435*, la cual cuenta con un sensor de profundidad y presenta diversas ventajas con respecto al sistema previamente propuesto. Además, la cámara de profundidad *Intel RealSense D435* ofrece datos de calidad para varias aplicaciones, como robótica o realidad aumentada. Con un amplio campo de visión y un alcance de hasta 10 m, con facilidad para su integración y compatibilidad con múltiples plataformas. Por lo mismo, se procedió inicialmente a evaluar su funcionamiento en términos de detección de profundidad tomando la imagen en la figura 9:

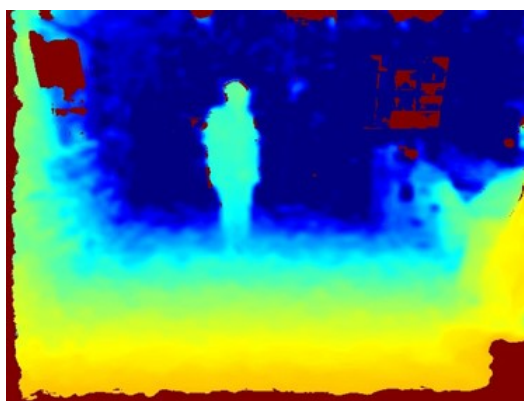


FIGURA 9: Imagen de Profundidad tomada con la cámara *Intel Realsense D435*

Considerando la situación, se planteó la estrategia de capturar dos imágenes, una del fondo, correspondiente al área de trabajo, y otra con el robot presente en la misma área. Posteriormente, se realizó la resta entre ambas imágenes con el fin de detectar la presencia del robot en una imagen resultante. En caso de obtener un resultado positivo, se procedería a utilizar técnicas de *clustering* para definir dos grupos: uno para definir el fondo y otro al robot, lo cual garantizaría una localización precisa del robot en cualquier momento de la tarea, permitiendo así su seguimiento continuo en la imagen.

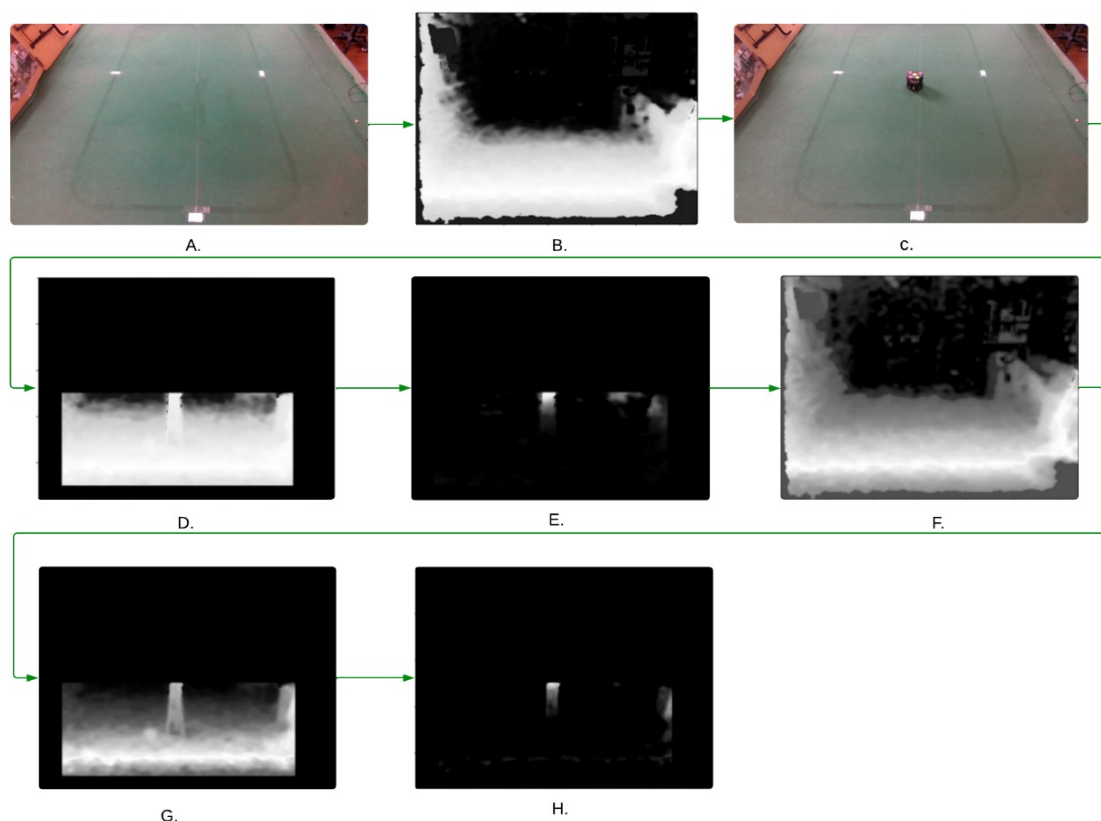


FIGURA 10: Proceso de pruebas en profundidad.

Tras llevar a cabo los pasos mencionados anteriormente, se generaron diversas imágenes que corresponden al proceso de detección mediante profundidad, como se muestra en la figura 10. Cada imagen representa una etapa del proceso. En (A), se aprecia la imagen del área de trabajo en una escala de color. En (B), la misma área se muestra en términos de profundidad. En (C), se observa la zona de trabajo con el robot presente en el campo. En (D), se utilizó un trípode como referencia visual para la posición del robot, ya que este se encontraba justo al lado. Además, se aplicó una región de interés (ROI) para reducir áreas no deseadas y minimizar el ruido en la imagen resultante.

En (E) es posible apreciar la resta de imágenes, sin embargo, el robot es apenas apreciable en la imagen original y se vuelve imperceptible en la resultante, esto debido al tamaño del robot y a la hora de realizar la resta, no se detecta al robot como una región destacable y es eliminado. Ante la falta de un resultado positivo, se exploraron diversas técnicas para mejorar la detección del robot en las imágenes originales, lo cual se aprecia de manera clara en (F) y (G), llegando a observar el robot de una manera más clara, sin embargo, de igual forma, al realizar la resta, como se observa en (H), no se reveló una mejora sobre la imagen resultante. Dadas estas limitaciones y considerando restricciones de tiempo, se decidió no continuar explorando el tema de la profundidad utilizando la cámara.

7.1.4. Detección por Color

Después de descartar otras estrategias para la detección de robots, se decidió mantener el enfoque en la misma cámara previamente mencionada. No obstante, esta vez se ajustó su configuración para reconocer y seguir colores específicos. De esta manera, en primer lugar, se establecieron umbrales de color para hacer detección de 3 colores, siendo estos el rosado, verde y amarillo. Estos umbrales buscan definir el rango de valores de los píxeles en el espacio de color RGB que se consideran como parte del color deseado. Además, los umbrales representan los mínimos y máximos de cada color que se busca detectar.

Para validar la detección de colores, se implementó un código que emplea la biblioteca *pyreal-sense2* para interactuar con la cámara *Realsense D435*, junto con *OpenCV* para el procesamiento de imágenes. En una fase inicial de prueba, se ajustaron los umbrales de color después de inicializar la cámara para visualizar la imagen en tiempo real y evaluar la efectividad de la detección de contornos vista en la figura 11.

En el caso visto en la figura 11, se busca hacer la detección de contornos cercanos, así como la agrupación de estos para dibujar un rectángulo alrededor del grupo de contornos y extraer su coordenada central en la imagen. Sin embargo, en las pruebas iniciales, la detección no se dio de manera estable, esto debido a la definición de los umbrales, encontrando múltiples contornos en toda el área observada. Para dar solución a la situación descrita, se realizó la captura de imágenes del robot en diferentes posiciones y condiciones de iluminación en el entorno. Esto se llevó a cabo considerando que las condiciones externas pueden afectar significativamente la detección de contornos en tiempo real. Posteriormente, se extrajeron los valores máximos y mínimos de cada color en el espacio RGB con el propósito de establecer múltiples umbrales, brindándole al sistema la capacidad de realizar una detección más precisa y adaptable a diferentes circunstancias dando como resultado la figura 12.

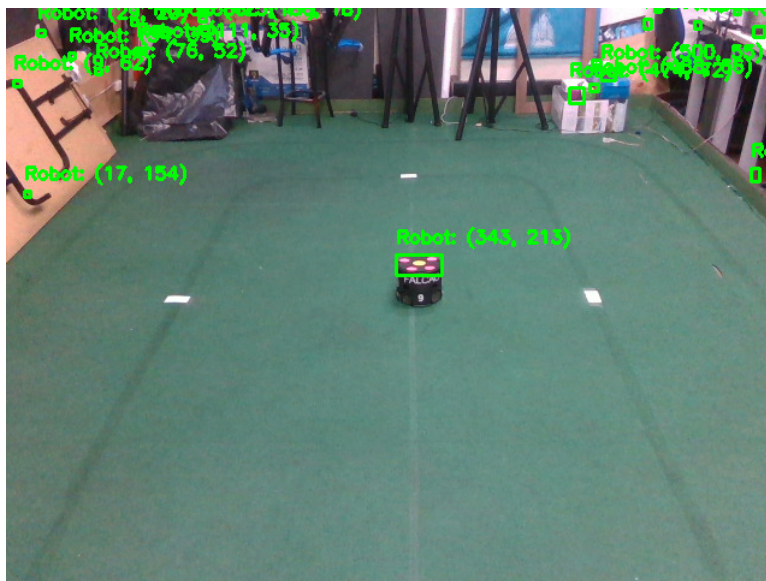


FIGURA 11: Detección de contornos.

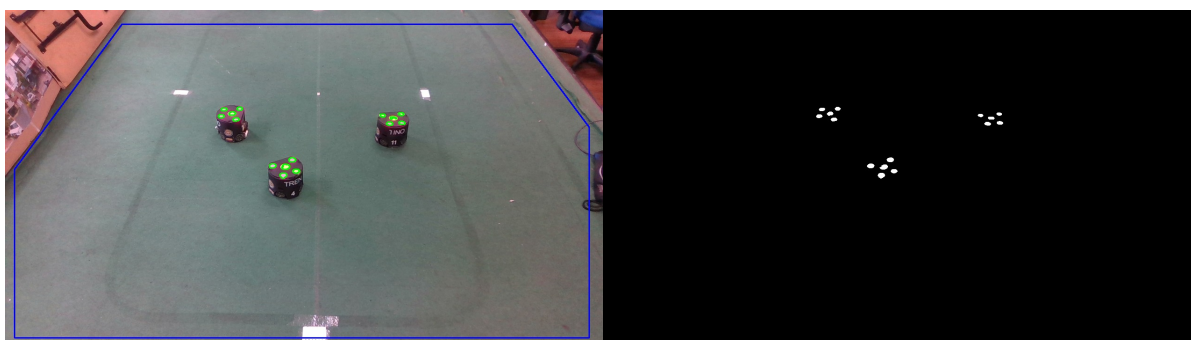


FIGURA 12: Detección de contornos estables.

Una vez aplicados los umbrales, la detección se dio de una manera más estable, sin embargo, en ocasiones se lograba apreciar la detección en pequeños puntos de la imagen. Por lo mismo, se optó por establecer una región de interés, delimitando de esta manera el campo de detección, así como la implementación de operaciones de filtrado destinadas a reducir el ruido detectado.

7.2. Fase 2

7.2.1. Nodo Publicador

A la hora de crear nodos publicadores en ROS2, es posible observar que el código sigue una estructura bien definida. El código, se encarga de publicar mensajes de un tópico específico,

creando un publicador con la capacidad de almacenar hasta 10 mensajes en una cola. Posteriormente se crea una función con el fin de establecer un temporizador que ejecuta periódicamente una función de devolución y dentro de dicha función se crea el mensaje con un contenido específico. De igual manera, cada que se llega a activar el temporizador, el mensaje se publica en el tópico definido y ejecutándose de manera continua hasta que sea detenido.

Una vez comprendido el funcionamiento del nodo publicador, se procedió a integrar la localización mediante el color a la estructura de ROS2. Esta adaptación implica la asignación de las funciones previamente probadas y se definió un tópico específico para la publicación de las coordenadas en el área delimitada. Es importante destacar que las coordenadas transmitidas están directamente relacionadas con el número de robots presentes en el campo. Por lo tanto, si durante la detección se identifica un robot, el nodo publica un único conjunto de coordenadas. Sin embargo, si se detectan múltiples robots, el nodo enviará varios pares de coordenadas correspondientes a la cantidad de robots identificados.

7.2.2. Nodo Subscriber

Al igual que el nodo publicador, el nodo subscriber presenta una estructura definida al momento de escribir el código. Este se encarga de escuchar los mensajes publicados en un tópico específico, de manera que el nodo se inicializa con un nombre o identificador. Posteriormente, se busca crear la suscripción al tópico, esperando recibir un mensaje del tipo que se asignó en el nodo publicador. Entonces, cuando el mensaje es publicado en el tópico, se activa una función de devolución, imprimiendo así el contenido del mensaje en el registro del nodo. Finalmente, el nodo continúa en ejecución, escuchando de manera continua los mensajes publicados en el tópico definido hasta que este se detenido.

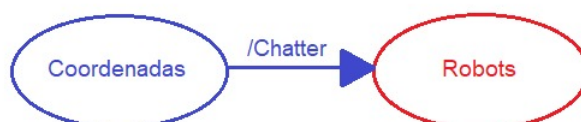


FIGURA 13: Conexión Publicador-Suscriptor.

En esta configuración, uno de los nodos actúa como publicador de datos, mientras que el otro se suscribe a esos datos como se puede ver en la figura 13. El nodo encargado de la publicación genera y envía información de coordenadas a través de un canal de comunicación designado.

El nodo receptor, suscrito a este canal, recibe los datos de coordenadas necesarios para su operación. Esta comunicación entre nodos permite que los robots obtengan información precisa sobre su posición, facilitando su navegación y coordinación dentro del entorno de trabajo.

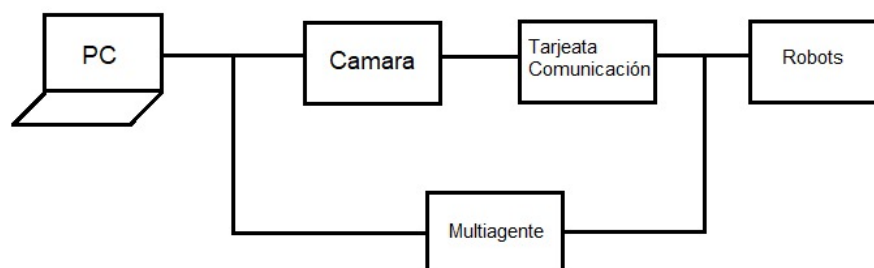


FIGURA 14: Diagrama de la composición del sistema.

Finalmente, la figura 14 presenta el diagrama de la composición del sistema. En él, se observa como entrada el computador, encargado del manejo de datos para la publicación y suscripción de la información. Posteriormente, se aprecia la cámara, mediante la cual se obtienen los datos de posicionamiento global de cada robot en el campo. La tarjeta de comunicación es la encargada de enviar los datos de velocidad a cada robot para que estos busquen su lugar de destino. Por último, se observa una retroalimentación proporcionada por el algoritmo multiagente, el cual se encarga de calcular los datos de velocidad con el fin de que los robots logren su objetivo.

Capítulo 8

Implementación y Resultados

8.1. Fase 1

8.1.1. Programación de rutinas de control

Para la programación de las rutinas de control, se llevó a cabo una investigación de métodos para acceder a las funcionalidades de los robots, enfocándose principalmente en sus capacidades de movimiento, dado que no era necesario manejar las demás funcionalidades ofrecidas. A través de la investigación, se identificó un repositorio que ya contenía la comunicación establecida entre los robots y la tarjeta controladora [77]. Para utilizar esta comunicación, fue suficiente con importar la librería correspondiente, realizar las inicializaciones necesarias y definir los identificadores de los robots utilizados. El primer método implementado, ilustrado en el flujo grama de la figura 15, explica el funcionamiento del movimiento de los robots y las velocidades utilizadas. A partir de los identificadores seleccionados en la tarjeta controladora, se establece la comunicación, y en el primer ciclo se determina la distancia que los robots recorrerán, mientras que en el segundo ciclo se establecen las velocidades que cada robot tendrá.

Con base en la figura 15, se realizaron las modificaciones para que los robots se formaran de una manera específica como se observa en la figura 16. Estas adaptaciones permitieron que los robots se movieran de una manera coordinada, manteniendo una disposición predeterminada durante sus desplazamientos.

Para lograr la formación específica de los robots, se realizaron ajustes en los ciclos mencionados anteriormente. Inicialmente, se emplearon dos ciclos; sin embargo, en este caso, se diseñó un ciclo para que los robots laterales se movieran en los ejes X y $-X$ respectivamente, y otro ciclo

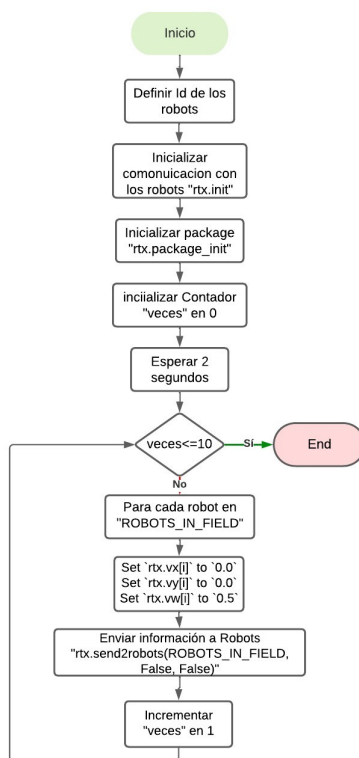


FIGURA 15: Flujo grama de rutina.



FIGURA 16: Robots en posición inicial.

para que el robot central se desplazara en el eje Y como se ve en la figura 17, permitiendo una coordinación precisa de los movimientos de los robots.

Tras comprender el movimiento y las velocidades de desplazamiento de los robots, se planteó una nueva meta: hacer que cada robot, desde su posición actual, alcanzara una coordenada específica. Esta nueva meta introdujo un desafío adicional: identificar con precisión la posición espacial de cada robot. Para abordar este desafío, se inició la fase 2 del proyecto, que se enfocó en el desarrollo de métodos y herramientas para determinar la ubicación exacta de los robots



FIGURA 17: Robots en posición final.

en el espacio, permitiendo así una navegación más precisa y controlada hacia las coordenadas objetivo en conjunto con la integración en ROS2.

8.2. Fase 2

8.2.1. Integración a ROS2

Una vez establecidas las rutinas de control y probadas en diferentes robots para su selección, se procedió a la integración de ROS2. Inicialmente, la implementación en el nodo publicador se manejó mediante el sistema de *bitcraze*. En conjunto con la implementación de ROS, se buscaba publicar las coordenadas necesarias para mover el robot de un punto a otro, utilizando el tópico correspondiente a las coordenadas de los robots. Sin embargo, debido al cambio en el sistema de detección, el proceso de publicación de las coordenadas se modificó para basarse en imágenes capturadas por la cámara. Estas imágenes se procesan para detectar áreas específicas de color mediante la creación de máscaras que filtran los colores deseados. Las máscaras se combinan y filtran para reducir el ruido, permitiendo la detección de contornos en las áreas filtradas. Los contornos se agrupan en función de su proximidad, y se calculan las coordenadas de los centroides de estos grupos.

Además, se publica el identificador único de cada robot. Esta asignación se realiza de acuerdo al número de contornos rosados por robot dentro de cada grupo detectado. Las coordenadas de los robots se formatean adecuadamente y se envían al tópico designado en intervalos regulares, lo que garantiza que las coordenadas se actualicen en tiempo real como se ven en la figura 18. Este ciclo se repite continuamente para mantener la información actualizada como se ve en la figura 19.



FIGURA 18: Grupo de robots con Identificador.

Es importante señalar que la agrupación de contornos se realizó mediante una técnica de clustering. En una primera instancia, se empleó el algoritmo K-means; no obstante, al intentar actualizar la información en tiempo real, se experimentó una notable disminución en el rendimiento del sistema, lo que resultó en una observación poco fluida del movimiento de los robots y en un retraso en la transmisión de la información publicada. Como solución al problema, se optó por usar DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Esta técnica fue seleccionada por la capacidad de identificar agrupaciones de datos basados en su densidad, utilizando dos parámetros principales: la distancia mínima entre los puntos y la cantidad mínima de puntos requeridos para formar un clúster.

```
[INFO] [1717610021.667188373] [robot_detection_node]: Robot coordinates published:
3 (431, 354)
4 (364, 380)
5 (510, 294)
Número de grupos: 3
Número de contornos en el grupo 0: 3
Robot ID: 3
Número de contornos en el grupo 1: 4
Robot ID: 4
Número de contornos en el grupo 2: 2
Robot ID: 5
[INFO] [1717610021.788145992] [robot_detection_node]: Robot coordinates published:
3 (431, 354)
4 (363, 380)
5 (510, 294)
Número de grupos: 3
Número de contornos en el grupo 0: 3
Robot ID: 3
Número de contornos en el grupo 1: 4
Robot ID: 4
Número de contornos en el grupo 2: 2
Robot ID: 5
[INFO] [1717610021.907338527] [robot_detection_node]: Robot coordinates published:
3 (431, 354)
4 (363, 380)
5 (510, 294)
```

FIGURA 19: Publicación Coordenadas e Identificadores.

En la figura 19, se puede observar el envío de información, que incluye el identificador del robot junto con su conjunto de coordenadas correspondientes. Además de esto, se visualizan varios mensajes que facilitan la comprensión de la información en tiempo real sobre los robots. Estos mensajes incluyen el número de clústeres formados y el recuento de contornos detectados por grupo. Es importante destacar que estos contornos corresponden al número de contornos rosados; por ejemplo, para los identificadores 3 y 4, se deben encontrar respectivamente 3 y

4 contornos, mientras que para el identificador 5, deben encontrarse 2 contornos en el grupo formado. La inclusión de estos mensajes se realizó con el propósito de identificar fácilmente cualquier error en la detección y facilitar su corrección.

Por otra parte, para implementar el nodo suscriptor, se busca permitir la comunicación, coordinación y control del grupo de robots en el entorno definido. Para ello, se establecieron una serie de configuraciones iniciales que incluyen una lista de identificadores para los robots en el campo, y las posiciones iniciales de estos robots almacenadas en una matriz. Estas posiciones se actualizan con los datos recibidos al suscribirse al nodo publicador y al tópico correspondiente a las coordenadas de los robots. Adicionalmente, se estableció una distancia mínima de parada y puntos de interés que delimitan zonas específicas en el campo, los cuales representan los puntos del área prohibida delimitada para el multiagente.

El sistema también se encarga de calcular las velocidades necesarias para que cada robot avance hacia su objetivo utilizando un controlador. Este controlador ajusta las velocidades lineales y angulares teniendo en cuenta los errores actuales, acumulados y la velocidad de cambio en las posiciones. En primer lugar, se determinan las posiciones actuales del robot y se calculan las diferencias en las coordenadas X y Y respecto a las posiciones deseadas. Estas diferencias se consideran como los errores actuales en las direcciones X y Y. Es importante destacar que las constantes proporcional, integral y derivativa se ajustaron empíricamente, realizando pruebas hasta alcanzar un rendimiento efectivo.

Para observar las funcionalidades mediante la integración de ROS2, se realizaron pruebas específicas en las que se indicó a los robots moverse a coordenadas predefinidas, lo cuál se observa en la figura 20. En cada iteración, se determinaron las velocidades requeridas para que cada robot alcance su destino, teniendo en cuenta la distancia restante hasta la posición objetivo, limitando las velocidades máximas permitidas para evitar movimientos bruscos o velocidades superiores a las deseadas.

```
[INFO] [1716491174.428639077] [data_subs]: Robot 3: (919.0, 397.0)
[INFO] [1716491174.429515434] [data_subs]: Robot 4: (403.0, 355.0)
[INFO] [1716491174.430368107] [data_subs]: Robot 5: (687.0, 377.0)
--- Nueva iteración ---
Coordenadas objetivo: (300, 300)
Coordenadas actuales del robot 3: (x:919.0, y:397.0)
Velocidades para el robot 3: V_lineal_x=0.8, V_lineal_y=0.8, V_angular=-1.0
--- Nueva iteración ---
Coordenadas objetivo: (600, 600)
Coordenadas actuales del robot 4: (x:403.0, y:355.0)
Velocidades para el robot 4: V_lineal_x=0.8, V_lineal_y=0.8, V_angular=1.0
--- Nueva iteración ---
Coordenadas objetivo: (900, 900)
Coordenadas actuales del robot 5: (x:687.0, y:377.0)
Velocidades para el robot 5: V_lineal_x=0.8, V_lineal_y=0.8, V_angular=1.0
```

FIGURA 20: Suscripción Coordenadas e Identificadores.

Finalmente en la figura 21 se observa la conexión en ROS2 realizada para la ejecución del movimiento en los robots, en donde uno de los nodos actúa como publicador de datos, mientras que el otro se suscribe a esos datos a través del tópico designado.

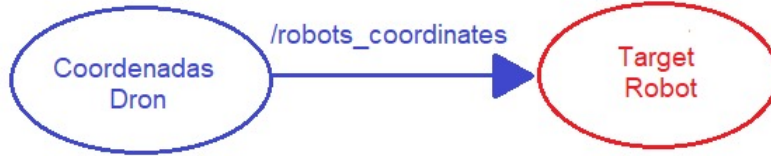


FIGURA 21: Diagrama de conexión en ROS.

8.3. Fase 3

8.3.1. Aplicación de la Técnica Multiagente

Para el desarrollo de la técnica multiagente, se optó por simular un escenario inicial de manera que se creó un entorno con 3 agentes, los cuales se representan mediante puntos, moviéndose en un área predefinida. En este entorno, los agentes interactúan entre ellos con el objetivo de evitar una zona restringida, dentro del área designada vista en la figura 22.

El código, basado en las investigaciones de [78] y [79], comienza definiendo el número de agentes presentes y sus respectivas posiciones. Posteriormente, se establece una matriz de conectividad basada en la posición de estos agentes y una distancia umbral denominada Delta. Si la distancia entre dos agentes es menor o igual a este umbral, se marca su conexión en la matriz con un valor de 1, indicando así una relación entre ellos.

En el bucle principal, se establece un ciclo con un tiempo determinado para ejecutar la tarea. En cada iteración, se elige aleatoriamente un agente y se define un conjunto de acciones posibles, que incluyen la opción de permanecer en su posición actual o moverse en diversas direcciones. Asimismo, se especifica un conjunto de acciones prohibidas que los agentes deben evitar. Luego, se filtran las acciones posibles para excluir aquellas que están prohibidas, y se calculan las probabilidades para que cada agente seleccione una nueva acción, utilizando la siguiente ecuación.

$$\Pr[\hat{a}_i = a_i] = \frac{1}{z_i}, \quad a_i \in R(a_i(t-1)) \setminus a_i(t-1) \quad (8.1)$$

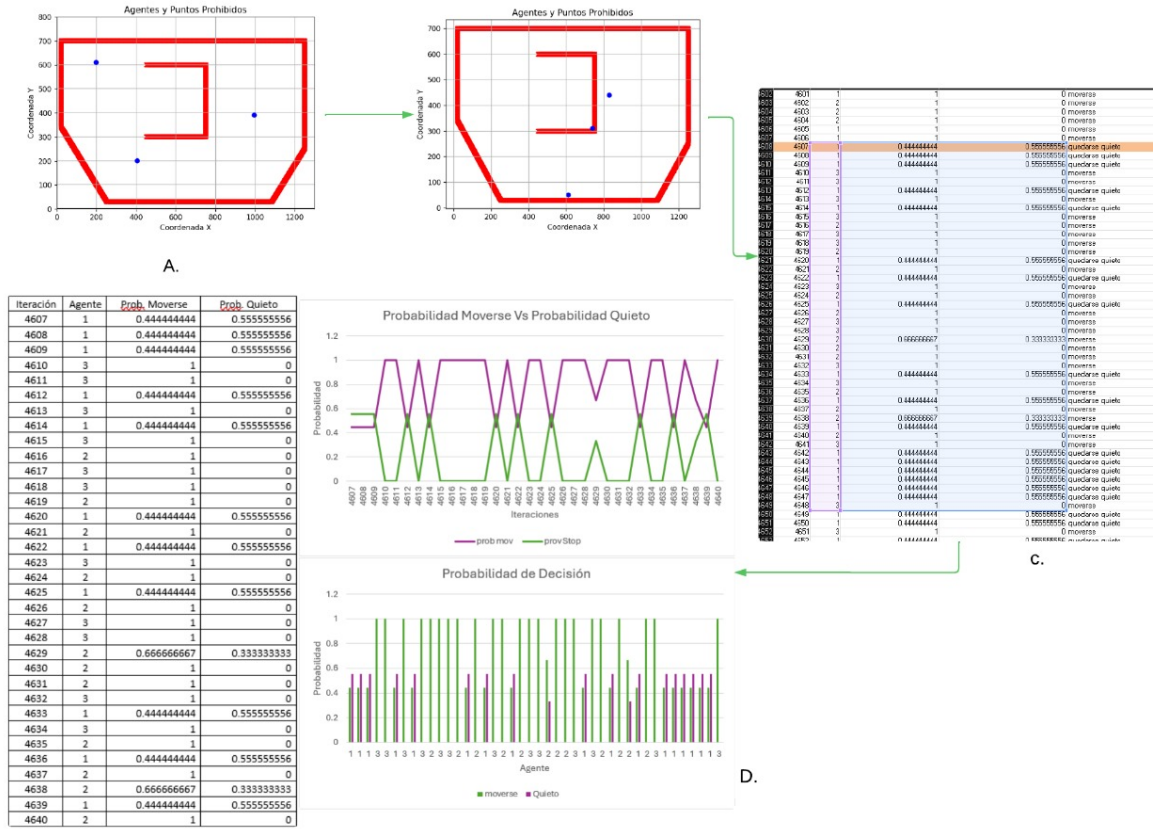


FIGURA 22: Simulación del entorno y los agentes.

o quedarse quieto

$$\Pr[\hat{a}_i = a_i(t-1)] = 1 - \frac{(|R_i(a_i(t-1))| - 1)}{z_i} \quad (8.2)$$

En estas etapas del proceso, se considera que z_i representa el número máximo de posiciones a las que puede moverse el agente. En la ecuación 8.1, no se tiene en cuenta la posición actual del agente, mientras que en la ecuación 8.2 sí se toma en consideración. Después de calcular las probabilidades, se elige una acción de prueba. Si la probabilidad de esta acción es mayor que la calculada, se selecciona aleatoriamente una acción del conjunto de acciones reales, de lo contrario, el agente permanece inmóvil.

En una matriz de influencia que considera la acción de prueba y las posiciones de los demás agentes, se analiza el posible movimiento del agente, influenciado por sus vecinos. Luego, se calcula una matriz de influencia futura, suponiendo que el agente lleva a cabo la acción de

prueba. Estas matrices se combinan en una matriz de decisión, donde se calculan las probabilidades de que el agente elija ejecutar la acción de prueba o permanecer en su posición actual, empleando las siguientes ecuaciones:

$$\Pr[a_i(t) = \hat{a}_i] = \frac{\exp\{\beta U_i(\hat{a}_i, a_{-j}(t-1))\}}{\exp\{\beta U_i(\hat{a}_i, a_{-j}(t-1))\} + \exp\{\beta U_i(a(t-1))\}} \quad (8.3)$$

$$\Pr[a_i(t) = a_i(t-1)] = \frac{\exp\{\beta U_i(\hat{a}_i, a_{-j}(t-1))\}}{\exp\{\beta U_i(\hat{a}_i, a_{-j}(t-1))\} + \exp\{\beta U_i(a(t-1))\}} \quad (8.4)$$

Donde $\hat{a}_i$ es la acción de ensayo escogida anteriormente, a_{-j} representa el set de acciones de los demás agentes en un tiempo $t-1$, U_i es la función objetivo del agente y P_i esta dada por:

$$U_i(a(t-1)) = - \sum_{P_j \in N_i} \|a_i(t-1) - a_j(t-1)\| \quad (8.5)$$

Y de igual manera:

$$U_i(\hat{a}_i, a_{-j}(t-1)) = - \sum_{P_j \in N_i} \|\hat{a}_i - a_j(t-1)\| \quad (8.6)$$

Si la probabilidad de $\Pr[a_i(t) = a_i(t-1)]$ es igual a uno el agente P_i se quedará quieto de lo contrario se actualizará a la acción de prueba, esto nos asegura un nivel de exploración a pesar de la influencia de otros vecinos. De manera que en la figura 23 se aprecia la descripción del funcionamiento del algoritmo multiagente.

Una vez explicado el desarrollo de la técnica multiagente, se procede con su implementación en el grupo de robots, adaptando variaciones pertinentes conforme a la definición de la zona prohibida. La adaptación se realiza considerando el campo abarcado por las cámaras usadas, así como por el grupo de robots en el campo. En lugar de enfocarse en los puntos internos, como se visualiza en la simulación, se priorizan los puntos externos que delimitan la región de interés donde residen los robots. Este enfoque está diseñado para asegurar que los robots permanezcan dentro de los límites definidos, evitando cualquier escapatoria de la zona especificada.

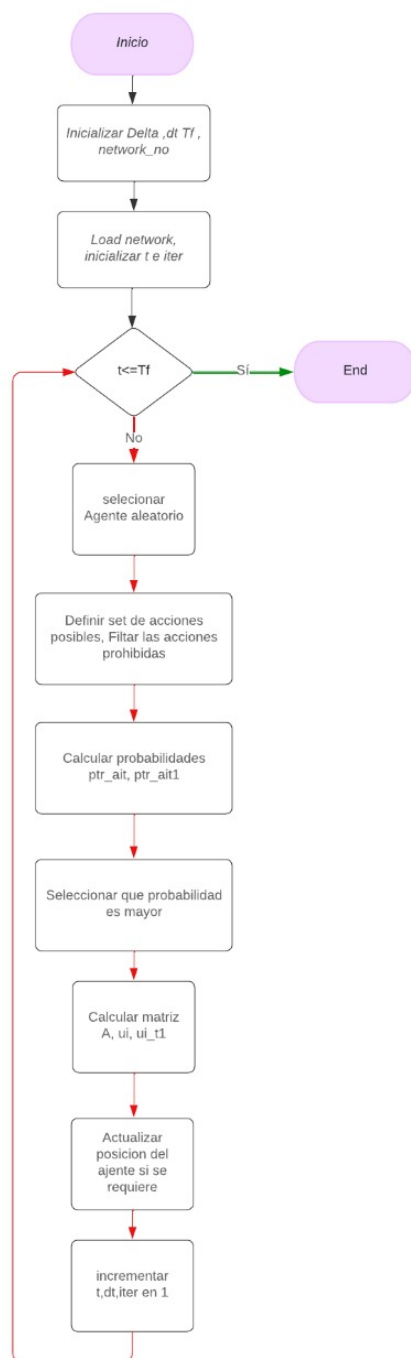


FIGURA 23: Diagrama de flujo Algoritmo Multiagente.

8.4. Fase 4

8.4.1. Validación en Escenarios de Prueba

Una vez que la detección de los robots y su movimiento fueron estabilizados en conjunto con la integración de ROS2, se procedió con la validación del sistema multiagente. La implementación se da en el grupo de robots, de manera que ahora el sistema permita mover los robots de manera autónoma y coordinada, asegurando que cada uno alcance su destino objetivo evitando una zona prohibida específica.

La función del multiagente fue desarrollada para gestionar las posiciones de los robots y calcular las velocidades necesarias para que estos se desplacen hacia sus objetivos. Esta gestión se realiza a través de un ciclo de iteraciones, donde se selecciona un robot de manera aleatoria y se define un conjunto de acciones posibles para su movimiento, incluyendo la posibilidad de quedarse en su posición actual. Las acciones prohibidas, es decir, aquellas que implican movimientos hacia zonas restringidas, son filtradas para asegurar que los robots no ingresen en dichas áreas.

Entonces, para la validación, fue necesaria la modificación de la zona prohibida en dos instancias, ya que las pruebas se realizaron con dos cámaras diferentes, la primera, la ya mencionada *Intel RealSense* y la segunda una cámara web genérica. El cambio se realizó pensando en la portabilidad del sistema, de manera que este pudiese ser llevado a cualquier parte, además de observar diferencia de comportamientos entre las dos cámaras.

De esta manera, en el diagrama de flujo del nodo publicador de la figura 24 se ejemplifica el proceso de publicación de coordenadas. Como se mencionó previamente, la publicación se realizó a través de dos cámaras, cada una con una zona prohibida definida de manera distinta debido al área cubierta por el tamaño de la imagen capturada como se observa en las figuras 25 y 26. A pesar de estas variaciones, la publicación de coordenadas se llevó a cabo de la misma manera. Los cambios más significativos se observaron en los códigos de color de las máscaras debido a las diferentes condiciones de iluminación a las que cada cámara fue expuesta.

Al comparar las imágenes capturadas, se pueden observar diferencias significativas entre ambas cámaras, especialmente en las condiciones de iluminación a las que están expuestas. La cámara *RealSense* demuestra una detección más estable frente a cambios bruscos de luz, lo que

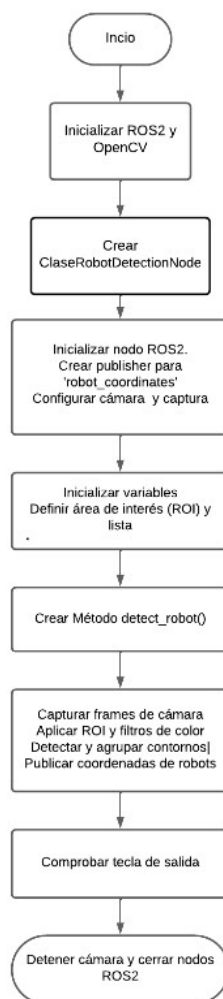


FIGURA 24: Diagrama de flujo nodo publicador.

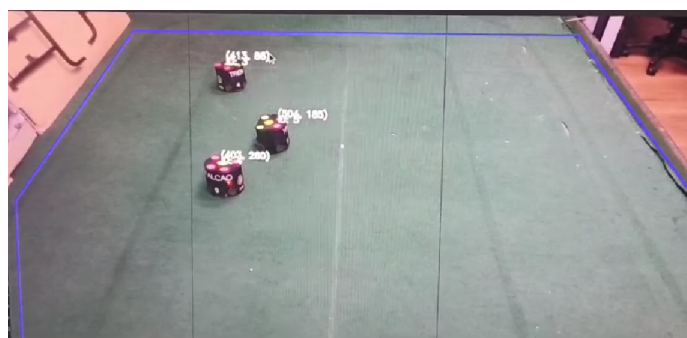


FIGURA 25: Detección con *RealSense*.

facilita el proceso de detección. En contraste, la cámara web requiere condiciones de iluminación controladas y mínimas para asegurar la visibilidad adecuada de los contornos. En situaciones de iluminación similar a la de la *RealSense*, la cámara web no logra identificar correctamente los contornos ni sus centros, especialmente a distancias más largas, donde los contornos

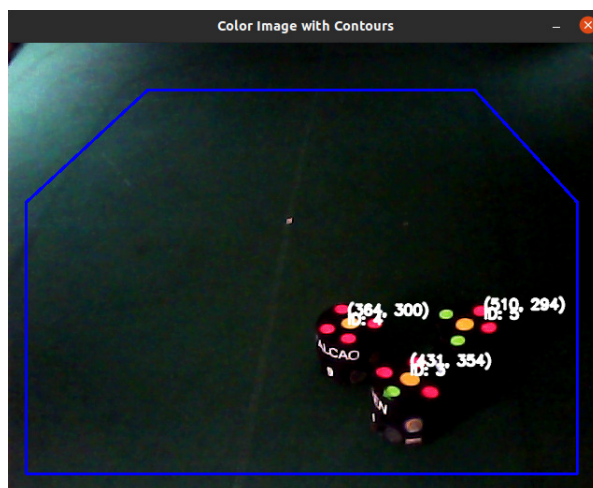


FIGURA 26: Detección con cámara web.

a menudo son detectados como blancos. Además, cabe destacar que cada cámara tiene un área de cobertura diferente debido al tamaño máximo de la imagen que pueden manejar. Aunque la geometría de la zona prohibida permanece igual, el alcance efectivo de la cámara web es menor en comparación con la *RealSense*.

La zona prohibida se delimitó trazando líneas entre puntos predefinidos que forman un perímetro, dichos puntos son correspondientes a la región de interés que se manejó para la detección de los robots. Estos puntos y las líneas que los conectan se almacenan y se consideran como áreas que los robots deben evitar. Durante el movimiento de los robots, se calcula la distancia de las acciones posibles a estas zonas prohibidas y se descartan aquellas que se encuentran dentro de un rango determinado, garantizando que los robots mantengan una distancia segura de estas áreas restringidas.

En el diagrama de flujo de la figura 27 se ejemplifica el desarrollo del nodo suscriptor, el cual recibe las coordenadas y asigna los identificadores correspondientes. Es importante destacar que las imágenes capturadas por ambas cámaras muestran a los robots en sus posiciones finales vista en la figura 28, lo que demuestra que la detección continúa incluso cuando los robots están detenidos. Esto asegura que, si alguno de los robots se separa del grupo y es ubicado en un área fuera de la zona de parada, el sistema sea capaz de guiar al robot de vuelta hacia una zona cercana al grupo que permanece detenido.

La máscara corresponde a la detección de los contornos de los robots en color rosado mediante la cámara web, cuando estos ya han alcanzado su posición final. Esta máscara es equivalente a la imagen en color mostrada previamente, la cual muestra claramente la cantidad de contornos cercanos utilizados para definir cada uno de los identificadores de los robots.

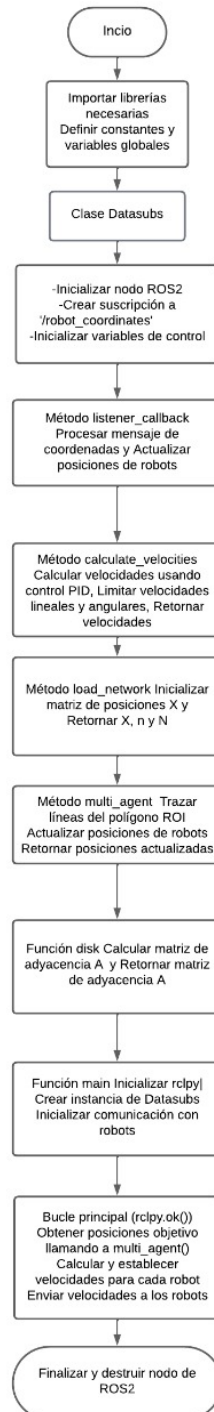


FIGURA 27: Diagrama de flujo nodo Suscriptor.

Por último, en el bucle principal del suscriptor, el sistema selecciona acciones para los robots basándose en un modelo multiagente, asegurando que las acciones seleccionadas no los lleven a zonas prohibidas. La posición de cada robot se actualiza de acuerdo con las acciones válidas

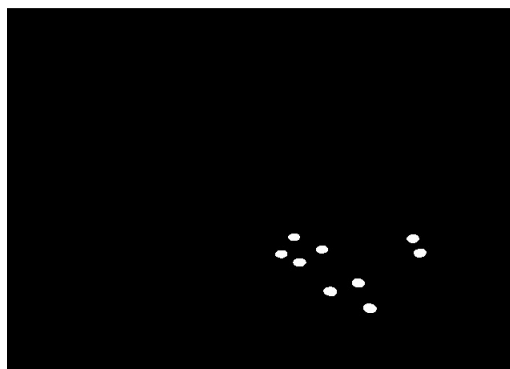


FIGURA 28: Máscara de los contornos en posición final.

seleccionadas, y se recalculan las velocidades necesarias para alcanzar las nuevas posiciones objetivo. Si dos robots están demasiado cerca entre sí, el sistema detiene el movimiento de esos robots para evitar colisiones, sin embargo, es posible que se dé el caso de colisión debido a la inercia que estos llevan. Finalmente, las velocidades calculadas se envían a los robots a través del sistema de comunicación, y el nodo de ROS 2 se mantiene activo para continuar recibiendo y procesando datos en tiempo real, esperando a ser detenido aún con los robots ya en sus posiciones finales.

De esta manera, se visualizan los mensajes finales en la figura 29 que indican la llegada de los robots a sus respectivos objetivos, es decir, cuando están más cercanos entre sí. El mensaje se genera cuando uno de los robots se acerca lo suficiente a otro. En este caso, se muestra el identificador del primer robot y se detiene. De manera similar, el segundo robot, al acercarse al primero, muestra su identificador y se detiene también. Así, cuando dos robots alcanzan sus objetivos, el tercer robot continúa buscando y, al encontrarlos, se muestra un tercer mensaje con el identificador correspondiente al último robot, validando de esta manera, el funcionamiento del multiagente.

```
[3, 0, 0, 0, 0, 170, 0, 0, 86, 4, 0, 0, 0, 0, 170, 0, 0, 81, 5, 0, 0, 0, 0, 170, 0, 0, 80]
[INFO] [1717113631.485887281] [data_subs]: Robot 3: (413.0, 86.0)
[INFO] [1717113631.486746173] [data_subs]: Robot 4: (403.0, 260.0)
[INFO] [1717113631.487802093] [data_subs]: Robot 5: (504.0, 185.0)
Robot 3 está lo suficientemente cerca de otro robot. Deteniéndolo.
Robot 4 está lo suficientemente cerca de otro robot. Deteniéndolo.
Robot 5 está lo suficientemente cerca de otro robot. Deteniéndolo.
3
4
5
[3, 0, 0, 0, 0, 170, 0, 0, 86, 4, 0, 0, 0, 0, 170, 0, 0, 81, 5, 0, 0, 0, 0, 170, 0, 0, 80]
[INFO] [1717113633.160691757] [data_subs]: Robot 3: (413.0, 86.0)
[INFO] [1717113633.167488054] [data_subs]: Robot 4: (403.0, 260.0)
[INFO] [1717113633.168260979] [data_subs]: Robot 5: (504.0, 185.0)
Robot 3 está lo suficientemente cerca de otro robot. Deteniéndolo.
Robot 4 está lo suficientemente cerca de otro robot. Deteniéndolo.
Robot 5 está lo suficientemente cerca de otro robot. Deteniéndolo.
```

FIGURA 29: Mensajes de paro.

Finalmente, se realizó una comparativa sencilla entre las cámaras utilizadas para visualizar el tiempo de ejecución del programa, como se observa en la tabla 1. Se aprecia que el tiempo de ejecución con la cámara genérica es significativamente menor en comparación con la *RealSense*. Esto se debe, en parte, a la reducción del tamaño de la imagen, lo cual reduce el área de movimientos de los robots, permitiendo que se localicen más rápidamente.

Cámara	Tiempo R1	Tiempo R2	Tiempo R3
RealSense	35	35	75
Genérica	13	15	28

TABLA 1: Comparación de tiempos entre cámaras RealSense y Genérica

Dada la comparativa presentada, se muestra un gráfico en la figura 30 que ilustra de manera más clara los tiempos de ejecución correspondientes a cada conjunto de datos. Este gráfico ofrece una visión un poco más detallada de las variaciones en los tiempos de ejecución, permitiendo una identificación de los tiempos asociados a cada cámara.

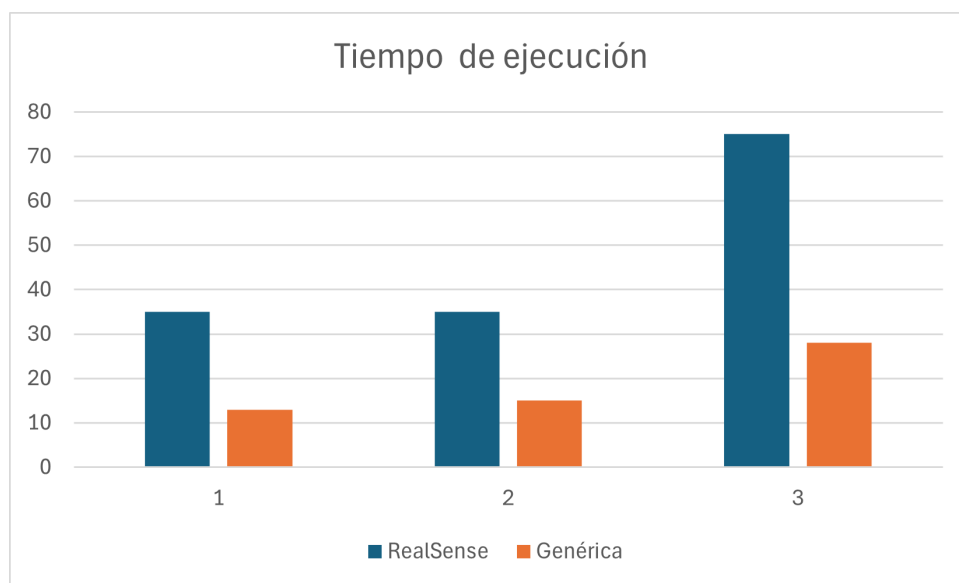


FIGURA 30: Comparación de tiempos entre cámaras.

Capítulo 9

Conclusiones y Trabajos Futuros

9.1. Conclusiones

- Debido al tiempo prolongado de inactividad, las funcionalidades de los robots se vieron significativamente afectadas, resultando en respuestas inadecuadas durante la ejecución de las rutinas básicas necesarias para comprender su funcionamiento. En consecuencia, los robots presentaron dificultades para moverse adecuadamente en el plano delimitado, mostrando movimientos irregulares. A raíz de estas limitaciones, se optó por seleccionar solo tres plataformas para el desarrollo de las actividades restantes. Las plataformas seleccionadas, a pesar de haber estado inactivas, mantenían una precisión extraordinaria, lo que las hacía ideales para cumplir con los objetivos del proyecto y para su uso en futuros desarrollos.
- La implementación en ROS2 permitió identificar diversas ventajas en la gestión del grupo de robots. ROS2 facilitó la gestión eficiente de múltiples nodos, integrando diversas funcionalidades y equipos de manera fluida. Esto se evidenció al integrar los robots con el sistema de localización de *bitcraze*, combinando las capacidades de ambos sistemas para guiar a los robots hacia coordenadas específicas con precisión. La comunicación en tiempo real ofrecida por ROS2 proporcionó mayor flexibilidad y precisión en las operaciones. Además, ROS2 permitió la operatividad de diferentes equipos durante la detección de los robots, destacándose en las pruebas realizadas con diversas cámaras. La capacidad de ROS2 para soportar y coordinar diferentes dispositivos mejoró significativamente la eficiencia y efectividad del sistema.

- En la implementación del sistema multiagente, se realizaron simulaciones con el objetivo de evitar que los robots salieran de la región designada y esquivaran obstáculos. En las simulaciones, estos objetivos se lograron sin problemas. Sin embargo, durante la implementación, surgieron dificultades debido a las dimensiones físicas de los robots y la fuerza de sus movimientos, lo que requirió limitar su velocidad para reducir la inercia. Mientras que en la simulación los robots se movían en unidades pequeñas, en la implementación se aumentó la distancia de movimiento, lo que llevó a una variación de la región prohibida. Esto causó que los robots pudieran salir de la región debido a los saltos en las coordenadas, ya que el algoritmo consideraba una acción válida si la coordenada de la región prohibida no se encontraba exactamente entre las coordenadas actuales y futuras del robot. Adicionalmente, se decidió eliminar el obstáculo del entorno de prueba, ya que su implementación interna en el código no permitió que los robots lo esquivaran adecuadamente. Aumentar el grosor del obstáculo se presentó como una solución, sin embargo, redujo el área de movimiento de los robots, impidiendo que se movieran de manera adecuada, sobre todo en la cámara web ya que esta abarca un área aún más pequeña. Por lo tanto, se optó por evaluar el algoritmo multiagente centrado en la organización de los robots sin la presencia del obstáculo.
- Durante la implementación se identificaron varios factores que pueden causar fallos en el sistema, tanto externos como relacionados con el hardware de los robots. En primer lugar, se constató que la iluminación es un factor sumamente complejo de gestionar. Independientemente de la cámara utilizada, las condiciones de luz pueden variar significativamente, afectando la precisión del sistema. Esto es particularmente problemático en el entorno de trabajo, donde la luz ambiental puede hacer que el sistema identifique erróneamente un objeto como un robot debido a la detección de color, lo que podría resultar en la pérdida de robots en el campo. Para reducir este problema, se implementaron filtros que reducen el ruido en la imagen, mejorando así la precisión de la detección y evitando la pérdida de robots durante la ejecución.
- Los fallos relacionados con el hardware de los robots se centran en la tarjeta de control, especialmente durante ejecuciones prolongadas. Se observó que, dado el tiempo que los robots pueden tardar en encontrarse entre sí y la cantidad de pruebas realizadas, la tarjeta de control experimentaba desconexiones frecuentes. Este problema recurrente obligaba a reiniciar la tarjeta, lo cual interrumpía los procesos en ejecución. Para observar el proceso en su totalidad, era necesario detener y reiniciar las pruebas, lo que afectaba la continuidad y eficiencia del desarrollo.

- En el sistema de detección, a la hora de asignar los identificadores también se puede incurrir en fallos debido a la forma en que se asignan estos, la cual se realiza mediante agrupación en un ciclo repetitivo para actualizar las coordenadas. Esto puede resultar en asignaciones incorrectas de identificadores. Uno de los escenarios posibles es que, si un robot está quieto en su coordenada objetivo o en movimiento, un cambio brusco de luz puede causar que el robot y su identificador se pierdan, o incluso que se marquen en un área incorrecta de la zona de trabajo, resultando en coordenadas erróneas y en la pérdida del robot. Otro problema surge de la técnica de clustering utilizada, la cual define una distancia a la cual los contornos pueden ser agrupados. Esto puede provocar que, si los robots están extremadamente cerca, se generen identificadores no deseados, causando confusión y pérdida de los robots hasta que se separen y se realice nuevamente el conteo de contornos y la asignación correcta de los identificadores.
- Se realizaron diversas pruebas del movimiento de los robots utilizando diferentes cámaras (Cámara de profundidad RealSense, webcam genérica, HD300Life), evidenciando que la mejor cámara para un desempeño efectivo de los robots fue la RealSense. Esta cámara ofrece mayores campos de visión tanto horizontal como vertical, en comparación con las demás, proporcionando un mayor rango para ubicar los robots y las regiones prohibidas. Además, la RealSense cuenta con una resolución de color superior, lo que mejora significativamente la calidad de las imágenes capturadas. Esta característica reduce el impacto del entorno sobre los filtros de detección, mejorando la precisión al identificar los robots. En contraste, las otras cámaras fueron afectadas considerablemente por la luz ambiental, lo que dificultó la detección precisa de los colores en la parte superior de los robots. Esta limitación impidió que el algoritmo multiagente implementado identificara correctamente la cantidad de robots, evitando que el sistema pudiera llevar a los robots a sus objetivos.
- Al realizar la comparativa entre las distintas cámaras, se observa una disminución en la región de interés en función del tamaño de la imagen que cada cámara es capaz de soportar. En particular, la cámara genérica presenta una región de movimiento mucho menor para los robots, lo que resulta en tiempos de llegada más reducidos a sus destinos. Sin embargo, esta reducción en el espacio de movimiento incrementa la probabilidad de colisiones entre los robots debido a la mayor concentración en áreas de trabajo válidas.

9.2. Trabajos Futuros

Como se mencionó previamente, la detección de los robots puede verse afectada por varios factores que pueden provocar errores durante la ejecución. Para mejorar el rendimiento del sistema, se sugiere explorar diversas metodologías para la detección e identificación de los robots. Esto incluye investigar nuevas técnicas de procesamiento de imágenes, algoritmos de visión por computadora o métodos avanzados de aprendizaje.

Integración y reconocimiento de obstáculos físicos para aplicación de técnicas multiagentes más robustas, con el fin de detectar y navegar alrededor de obstáculos de manera eficiente, facilitando la operación autónoma en entornos controlados o dinámicos y la exploración en un terrenos complejos.

Exploración de técnicas de reconocimiento a través de profundidad con el fin de mitigar las limitaciones provocadas por condiciones lumínicas, mejorando la percepción de los robots en entornos con variaciones de luz, permitiéndoles operar de manera más confiable y precisa en aplicaciones como la navegación autónoma y la detección de objetos en tiempo real.

Exploración en técnicas de optimización en control para el manejo de las velocidades del grupo de robots para evasión de objetos y evitar colisiones, desarrollando algoritmos que permitan a los robots ajustar dinámicamente sus velocidades basadas en la detección de agentes cercanos.

Investigación y desarrollo de sistemas y protocolos de comunicación robustos para entornos adversos y escenarios de pérdida de señal. El proyecto se enfocaría en diseñar protocolos de comunicación que permitan a los robots mantener una comunicación efectiva y confiable entre sí, incluso en condiciones de interferencia o pérdida de señal.

Desarrollo de técnicas de visión local para ampliar el número de agentes en el campo, facilitando la reorganización posicional y reduciendo colisiones. El enfoque se da en implementar sistemas de percepción avanzados en cada plataforma robótica, permitiendo a múltiples agentes detectar y evitar obstáculos de manera eficiente.

Desarrollo de algoritmos avanzados de planificación de rutas para robots móviles en entornos dinámicos y cambiantes, diseñando sistemas que permitan a los robots calcular trayectorias óptimas considerando obstáculos en tiempo real, variaciones externas y dinámicas del entorno, buscando la exploración de enfoques como el aprendizaje para mejorar la eficiencia y seguridad de la navegación autónoma de los robots en diversas aplicaciones prácticas.

Investigación y desarrollo de estrategias avanzadas de control PID adaptativo para sistemas robóticos autónomos, explorando métodos para ajustar dinámicamente los parámetros del controlador PID en tiempo real, optimizando el rendimiento del sistema mediante técnicas de identificación de modelos, control predictivo y ajuste automático de parámetros.

Bibliografía

- [1] J. G. Guarnizo Marin, D. Bautista Díaz y J. S. Sierra Torres, «Una revisión sobre la evolución de la robótica móvil», 2021.
- [2] S. G. Tzafestas, «Mobile robot control and navigation: A global overview», *Journal of Intelligent & Robotic Systems*, vol. 91, págs. 35-58, 2018.
- [3] F. Rubio, F. Valero y C. Llopis-Albert, «A review of mobile robots: Concepts, methods, theoretical framework, and applications», *International Journal of Advanced Robotic Systems*, vol. 16, n.º 2, pág. 1729 881 419 839 596, 2019.
- [4] K. Zhang, Z. Yang y T. Başar, «Multi-agent reinforcement learning: A selective overview of theories and algorithms», *Handbook of reinforcement learning and control*, págs. 321-384, 2021.
- [5] A. Oroojlooy y D. Hajinezhad, «A review of cooperative multi-agent deep reinforcement learning», *Applied Intelligence*, vol. 53, n.º 11, págs. 13 677-13 722, 2023.
- [6] I. F. of Robotics, *Top 5 Robot Trends 2023*, <https://ifr.org/ifr-press-releases/news/top-5-robot-trends-2023>, 2023.
- [7] J. A. Gurzoni, M. F. Martins, F. Tonidandel y R. A. Bianchi, «On the construction of a RoboCup small size league team», *Journal of the Brazilian Computer Society*, vol. 17, págs. 69-82, 2011.
- [8] C. Camacho, C. Higuera, J. Guarnizo, Y. Suarez y N. Garzon, *STOx's Team description paper 2018*. ene. de 2018.
- [9] *Robot Operating System*, <https://www.ros.org>, 2021.
- [10] D. McNulty, A. Hennessy, M. Li, E. Armstrong y K. M. Ryan, «A review of Li-ion batteries for autonomous mobile robots: Perspectives and outlook for the future», *Journal of Power Sources*, vol. 545, pág. 231 943, 2022.

-
- [11] A. Loganathan y N. S. Ahmad, «A systematic review on recent advances in autonomous mobile robot navigation», *Engineering Science and Technology, an International Journal*, vol. 40, págs. 101-134, 2023.
 - [12] G. Fragapane, R. De Koster, F. Sgarbossa y J. O. Strandhagen, «Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda», *European Journal of Operational Research*, vol. 294, n.º 2, págs. 405-426, 2021.
 - [13] S. Macenski, T. Foote, B. Gerkey, C. Lalancette y W. Woodall, «Robot Operating System 2: Design, architecture, and uses in the wild», *Science Robotics*, vol. 7, n.º 66, eabm6074, 2022.
 - [14] A. J. Lee, W. Song, B. Yu, D. Choi, C. Tirtawardhana y H. Myung, «Survey of robotics technologies for civil infrastructure inspection», *Journal of Infrastructure Intelligence and Resilience*, vol. 2, n.º 1, págs. 100-118, 2023.
 - [15] Y. Bai, B. Zhang, N. Xu, J. Zhou, J. Shi y Z. Diao, «Vision-based navigation and guidance for agricultural autonomous vehicles and robots: A review», *Computers and Electronics in Agriculture*, vol. 205, págs. 107-154, 2023.
 - [16] M. Tavakoli, J. Carriere y A. Torabi, «Robotics, smart wearable technologies, and autonomous intelligent systems for healthcare during the COVID-19 pandemic: An analysis of the state of the art and future vision», *Advanced Intelligent Systems*, vol. 2, n.º 7, págs. 2000-071, 2020.
 - [17] J. G. Guarnizo y M. M. Arteché, «Robot soccer strategy based on hierarchical finite state machine to centralized architectures», *IEEE Latin America Transactions*, vol. 14, n.º 8, págs. 3586-3596, 2016.
 - [18] E. Antonioni, V. Suriani, F. Riccio y D. Nardi, «Game strategies for physical robot soccer players: a survey», *IEEE Transactions on Games*, vol. 13, n.º 4, págs. 342-357, 2021.
 - [19] J. G. Guarnizo, C. L. Trujillo y N. L. Díaz, «Fútbol de robots: orígenes, federaciones, ligas y horizontes de investigación-Robot Soccer: Origins, Federations, Leagues and Research Horizons», *Ingenium Revista de la facultad de ingeniería*, vol. 17, n.º 33, págs. 54-67, 2016.
 - [20] A. Weitzenfeld, J. Biswas, M. Akar y K. Sukvichai, «Robocup small-size league: Past, present and future», en *RoboCup 2014: Robot World Cup XVIII* 18, Springer, 2015, págs. 611-623.
 - [21] Y. Jiang, X. Li, H. Luo, S. Yin y O. Kaynak, «Quo vadis artificial intelligence?», *Discover Artificial Intelligence*, vol. 2, n.º 1, págs. 4, 2022.
 - [22] P. P. Angelov, E. A. Soares, R. Jiang, N. I. Arnold y P. M. Atkinson, «Explainable artificial intelligence: an analytical review», *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 11, n.º 5, e1424, 2021.

-
- [23] Q. Zhang, L. T. Yang, Z. Chen y P. Li, «A survey on deep learning for big data», *Information Fusion*, vol. 42, págs. 146-157, 2018.
 - [24] M. Gheisari, G. Wang y M. Z. A. Bhuiyan, «A survey on deep learning in big data», en *2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC)*, IEEE, vol. 2, 2017, págs. 173-180.
 - [25] S. F. Nasim, M. R. Ali y U. Kulsoom, «Artificial intelligence incidents & ethics a narrative review», *International Journal of Technology, Innovation and Management (IJTIM)*, vol. 2, n.º 2, págs. 52-64, 2022.
 - [26] L. H. Kaack, P. L. Donti, E. Strubell, G. Kamiya, F. Creutzig y D. Rolnick, «Aligning artificial intelligence with climate change mitigation», *Nature Climate Change*, vol. 12, n.º 6, págs. 518-527, 2022.
 - [27] A. G. Marcos, F. J. M. de Pisón Ascacíbar, F. A. Elías, M. C. Limas, J. B. O. Meré, E. P. V. González et al., «Técnicas y algoritmos básicos de visión artificial», *Técnicas y Algoritmos Básicos de Visión Artificial*, 2006.
 - [28] A. Voulodimos, N. Doulamis, A. Doulamis y E. Protopapadakis, «Deep learning for computer vision: A brief review», *Computational intelligence and neuroscience*, vol. 2018, n.º 1, pág. 7068349, 2018.
 - [29] N. O'Mahony, S. Campbell, A. Carvalho et al., «Deep learning vs. traditional computer vision», en *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1*, Springer, 2020, págs. 128-144.
 - [30] M. A. Ponti, L. S. F. Ribeiro, T. S. Nazare, T. Bui y J. Collomosse, «Everything you wanted to know about deep learning for computer vision but were afraid to ask», en *2017 30th SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T)*, IEEE, 2017, págs. 17-41.
 - [31] J. Janai, F. Güney, A. Behl, A. Geiger et al., «Computer vision for autonomous vehicles: Problems, datasets and state of the art», *Foundations and Trends® in Computer Graphics and Vision*, vol. 12, n.º 1-3, págs. 1-308, 2020.
 - [32] A. Dorri, S. S. Kanhere y R. Jurdak, «Multi-agent systems: A survey», *Ieee Access*, vol. 6, págs. 28573-28593, 2018.
 - [33] S. Gronauer y K. Diepold, «Multi-agent deep reinforcement learning: a survey», *Artificial Intelligence Review*, vol. 55, n.º 2, págs. 895-943, 2022.

-
- [34] E. Erős, M. Dahl, K. Bengtsson, A. Hanna y P. Falkman, «A ROS2 based communication architecture for control in collaborative and intelligent automation systems», *Procedia Manufacturing*, vol. 38, págs. 349-357, 2019.
 - [35] M. Quigley, B. Gerkey y W. D. Smart, *Programming Robots with ROS: a practical introduction to the Robot Operating System*. " O'Reilly Media, Inc.", 2015.
 - [36] Z. Gao, T. Wanyama, I. Singh, A. Gadhri y R. Schmidt, «From industry 4.0 to robotics 4.0-a conceptual framework for collaborative and intelligent robotic systems», *Procedia manufacturing*, vol. 46, págs. 591-599, 2020.
 - [37] E. C. Camacho, C. Higuera, J. G. Guarnizo, Y. Suarez, N. Garzon y N. Bonifaz, «STOx's Team description paper 2018»,
 - [38] S. Rodríguez, E. Rojas, K. Pérez, J. Lopez-Jimenez, C. Quintero y J. Calderon, «STOx's 2014 Extended Team Description Paper», jul. de 2014.
 - [39] J. Qian, B. Zi, D. Wang, Y. Ma y D. Zhang, «The design and development of an omnidirectional mobile robot oriented to an intelligent manufacturing system», *Sensors*, vol. 17, n.º 9, pág. 2073, 2017.
 - [40] M. R. Azizi, A. Rastegarpanah y R. Stolkin, «Motion planning and control of an omnidirectional mobile robot in dynamic environments», *Robotics*, vol. 10, n.º 1, pág. 48, 2021.
 - [41] A. Sheikhlari y A. Fakharian, «Adaptive optimal control via reinforcement learning for omnidirectional wheeled robots», en *2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA)*, IEEE, 2016, págs. 208-213.
 - [42] M. A. Khalighi y M. Uysal, «Survey on free space optical communication: A communication theory perspective», *IEEE communications surveys & tutorials*, vol. 16, n.º 4, págs. 2231-2258, 2014.
 - [43] A. J. Paulraj, D. A. Gore, R. U. Nabar y H. Bolcskei, «An overview of MIMO communications-a key to gigabit wireless», *Proceedings of the IEEE*, vol. 92, n.º 2, págs. 198-218, 2004.
 - [44] A. E. Willner, Y. Ren, G. Xie et al., «Recent advances in high-capacity free-space optical and radio-frequency communications using orbital angular momentum multiplexing», *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 375, n.º 2087, pág. 20150439, 2017.
 - [45] Y. J. Lee, I. Atkinson, J. Trevathan, W. Read y T. Myers, «An Intelligent Agent System for Managing Heterogeneous Sensors in Dispersed and Disparate Wireless Sensor Network Systems»,
 - [46] Y. Yang, Y. Xiao y T. Li, «Attacks on formation control for multiagent systems», *IEEE Transactions on Cybernetics*, vol. 52, n.º 12, págs. 12805-12817, 2021.

-
- [47] F. Derakhshan y S. Yousefi, «A review on the applications of multiagent systems in wireless sensor networks», *International Journal of Distributed Sensor Networks*, vol. 15, n.º 5, pág. 1 550 147 719 850 767, 2019.
 - [48] Y. Li y C. Tan, «A survey of the consensus for multi-agent systems», *Systems Science & Control Engineering*, vol. 7, n.º 1, págs. 468-482, 2019.
 - [49] R. Olfati-Saber, J. A. Fax y R. M. Murray, «Consensus and cooperation in networked multi-agent systems», *Proceedings of the IEEE*, vol. 95, n.º 1, págs. 215-233, 2007.
 - [50] Y. Zheng, J. Ma y L. Wang, «Consensus of hybrid multi-agent systems», *IEEE transactions on neural networks and learning systems*, vol. 29, n.º 4, págs. 1359-1365, 2017.
 - [51] H. Zhang y S. Liyanage, «Finite-time formation control for multi-agent systems underlying heterogeneous communication typologies», en *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, IEEE, 2020, págs. 1441-1446.
 - [52] L. Xue, Y. Liu, Z.-Q. Gu, Z.-H. Li y X.-P. Guan, «Joint design of clustering and in-cluster data route for heterogeneous wireless sensor networks», *International Journal of Automation and Computing*, vol. 14, n.º 6, págs. 637-649, 2017.
 - [53] J. R. Marden, G. Arslan y J. S. Shamma, «Cooperative Control and Potential Games», *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, n.º 6, págs. 1393-1407, 2009. DOI: 10.1109/TSMCB.2009.2017273.
 - [54] A. C. Jiménez, V. García-Díaz y S. Bolaños, «A decentralized framework for multi-agent robotic systems», *Sensors*, vol. 18, n.º 2, pág. 417, 2018.
 - [55] B. Xin, G.-Q. Gao, Y.-L. Ding, Y.-G. Zhu y H. Fang, «Distributed multi-robot motion planning for cooperative multi-area coverage», en *2017 13th IEEE International Conference on Control & Automation (ICCA)*, IEEE, 2017, págs. 361-366.
 - [56] G. Philip, S. Givigi y H. Schwartz, «Multi-robot exploration using potential games», ago. de 2013, págs. 1203-1210, ISBN: 978-1-4673-5557-5. DOI: 10.1109/ICMA.2013.6618085.
 - [57] Y. L. Lim, «Potential game based cooperative control in dynamic environments», 2011.
 - [58] D. S. Leslie y J. R. Marden, «Equilibrium selection in potential games with noisy rewards», en *International Conference on Network Games, Control and Optimization (NetGCooP 2011)*, 2011, págs. 1-4.
 - [59] J. K. Gupta, M. Egorov y M. Kochenderfer, «Cooperative multi-agent control using deep reinforcement learning», en *Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers 16*, Springer, 2017, págs. 66-83.

-
- [60] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor y S. Levine, «How to train your robot with deep reinforcement learning: lessons we have learned», *The International Journal of Robotics Research*, vol. 40, n.º 4-5, págs. 698-721, 2021.
 - [61] T. Johannink, S. Bahl, A. Nair et al., «Residual reinforcement learning for robot control», en *2019 international conference on robotics and automation (ICRA)*, IEEE, 2019, págs. 6023-6029.
 - [62] T. Zhang y H. Mo, «Reinforcement learning for robot research: A comprehensive review and open issues», *International Journal of Advanced Robotic Systems*, vol. 18, n.º 3, pág. 17298814211007305, 2021.
 - [63] J. Kober, J. Bagnell y J. Peters, «Reinforcement Learning in Robotics: A Survey», *The International Journal of Robotics Research*, vol. 32, págs. 1238-1274, sep. de 2013. DOI: 10.1177/0278364913495721.
 - [64] H. Bae, G. Kim, J. Kim, D. Qian y S. Lee, «Multi-robot path planning method using reinforcement learning», *Applied sciences*, vol. 9, n.º 15, pág. 3057, 2019.
 - [65] C. Russo, H. Ramón, L. Cicerchia et al., «Visión artificial aplicada en agricultura de precisión», 2018.
 - [66] L. Zhou, L. Zhang y N. Konz, «Computer vision techniques in manufacturing», *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, n.º 1, págs. 105-117, 2022.
 - [67] Z. Liu, H. Ukida, P. Ramuhalli y D. Forsyth, «Integrated imaging and vision techniques for industrial inspection: A special issue on machine vision and applications», *Mach. Vis. Appl.*, vol. 21, págs. 597-599, ago. de 2010. DOI: 10.1007/s00138-010-0277-9.
 - [68] A. M. Al-Oraiqat, T. Smirnova, O. Drieiev et al., «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology», *Sensors*, vol. 22, n.º 16, 2022, ISSN: 1424-8220. DOI: 10.3390/s22166223. dirección: <https://www.mdpi.com/1424-8220/22/16/6223>.
 - [69] Y. Matsuzaka y R. Yashiro, «AI-based computer vision techniques and expert systems», *AI*, vol. 4, n.º 1, págs. 289-302, 2023.
 - [70] C. Patruno, V. Renò, M. Nitti, N. Mosca, M. di Summa y E. Stella, «Vision-based omnidirectional indoor robots for autonomous navigation and localization in manufacturing industry», *Heliyon*, 2024.
 - [71] G. Nirmala, S. Geetha y S. Selvakumar, «Mobile robot localization and navigation in artificial intelligence: Survey», *Computational Methods in Social Sciences*, vol. 4, n.º 2, pág. 12, 2016.

-
- [72] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz y D. Terzopoulos, «Image segmentation using deep learning: A survey», *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, n.º 7, págs. 3523-3542, 2021.
 - [73] Y. Guo, Y. Liu, T. Georgiou y M. S. Lew, «A review of semantic segmentation using deep neural networks», *International journal of multimedia information retrieval*, vol. 7, págs. 87-93, 2018.
 - [74] S. Peng, W. Jiang, H. Pi, X. Li, H. Bao y X. Zhou, «Deep snake for real-time instance segmentation», en *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, págs. 8533-8542.
 - [75] F. Garcia-Lamont, J. Cervantes, A. López y L. Rodríguez, «Segmentation of images by color features: A survey», *Neurocomputing*, vol. 292, págs. 1-27, 2018.
 - [76] *Lighthouse Positioning System*, <https://www.bitcraze.io/documentation/system/positioning/lighthouse-positioning-system/>, 2015.
 - [77] *Lighthouse Positioning System*, https://github.com/edgarcamilocamacho/promocionUsta/blob/master/futbolistas_joystick/promocion.pyL170/.
 - [78] D. A. Martínez, E. Mojica-Nava, K. Watson y T. Usländer, «Multiagent Self-Redundancy Identification and Tuned Greedy-Exploration», *IEEE Transactions on Cybernetics*, vol. 52, n.º 7, págs. 5744-5755, 2022. DOI: 10.1109/TCYB.2020.3035783.
 - [79] D. Martínez y E. Mojica-Nava, «Distortion based potential game for distributed coverage control», *Information Sciences*, vol. 600, págs. 209-225, 2022, ISSN: 0020-0255. DOI: <https://doi.org/10.1016/j.ins.2022.03.090>. dirección: <https://www.sciencedirect.com/science/article/pii/S0020025522003176>.