

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL
INSTITUTO POLITÉCNICO NACIONAL

Unidad Saltillo
Robótica y Manufactura Avanzada

“Coordinación de robots móviles en ambientes estructurados”

Tesis
que presenta

Edgar Baizabal Montañez

para obtener el Grado de

Maestro en Ciencias
en
Robótica y Manufactura Avanzada

Director de tesis: Dra. América Berenice Morales Díaz
Dr. Keny Ordaz Hernández

Dedico este trabajo a mi madre Claudia Montañez, mi padre Manuel Baizabal, mi abuela Hilda Carvallo y a Paúl Hernández.

Agradecimientos

- 1 Este trabajo fue posible gracias al apoyo incondicional de mis padres, Claudia Montañez y
2 Manuel Baizabal, por estar siempre a mi lado y orientarme en mi camino.
3
- 4 A mis asesores, la Doctora América Berenice Morales Díaz y el Doctor Keny Ordaz Hernández,
5 por la paciencia y enseñanzas que me brindaron durante el desarrollo de este trabajo.
6
- 7 A mis compañeros de laboratorio, Hugo Rico y Misael Reyna; por toda la amistad, convivencia
8 y apoyo durante todo el proceso formativo de la maestría.
9
- 10 A mis amigos de generación; Jesús, Jorge, Carlos, Isaias, Juan, Misael y Hugo. Su amistad
11 fue clave para crecer profesional y personalmente.
12
- 13 A CONACYT, por el apoyo económico otorgado que me permitió realizar este proyecto.
14
- 15 A CINVESTAV, por la oportunidad de estudiar una maestría en robótica.

1 Resumen

2 Los altos niveles de producción industrial y la constante innovación en los procesos de manu-
3 factura, han llevado a la investigación y desarrollo de nuevas estrategias de automatización.
4 Por esta razón, en este proyecto de tesis, se desarrolló una estrategia de navegación coordinada
5 para robots móviles diferenciales, como propuesta para mover bienes dentro de un almacén in-
6 dustrial de forma automatizada. Esta estrategia integra un algoritmo de control no lineal basado
7 en consenso, junto con el algoritmo A^* como planificador de rutas, en un ambiente estructurado
8 con obstáculos virtuales.

9 Se comenzó con el diseño del control de regulación no lineal para un robot, el cual le permi-
10 tió seguir todos los puntos de la ruta encontrada con A^* . Sin embargo, esta ruta no contemplaba
11 la restricción no holónoma del robot; por esta razón, fue necesario implementar un ajuste de
12 rutas por medio de un recorte de esquinas al camino encontrado.

13 Para establecer el algoritmo de coordinación entre varios robots, se propuso un control no
14 lineal acoplado; el cual, primeramente, se implementó en simulación para la navegación de dos
15 robots en una ruta encontrada con A^* . Posteriormente, se llevó a cabo de forma experimental
16 con el apoyo de marcadores ArUco, para establecer la detección de los robots en el ambiente
17 estructurado.

18 Se realizaron un total de seis experimentos, donde los resultados mostraron de forma satis-
19 factoria cómo la navegación coordinada fue capaz de seguir la secuencia de puntos obtenidos en
20 las rutas encontradas con A^* . En los experimentos se varió el entorno de navegación, la cantidad
21 de robots y las conexiones entre ellos.

22 Como trabajo futuro, se propone la adición de restricciones a los robots para lograr la eva-
23 sión de obstáculos. Por otro lado, se propone una mejora mediante la implementación de algo-
24 ritmos de aproximación polinómica que permitan convertir las rutas encontradas en trayectorias
25 continuas y, con esto, implementar un control de consenso para seguimiento de trayectorias.

¹ Abstract

² The high levels of industrial production and the constant innovation in manufacturing processes have led to the research and development of new automation strategies. For this reason, in this thesis project, a coordinated navigation strategy for differential mobile robots was developed as a proposal to move goods within an industrial warehouse. This strategy integrates a consensus-based nonlinear control algorithm, together with the A^* algorithm as a path planner in a structured environment with virtual obstacles.

⁸ First, the design of a nonlinear regulatory control for a robot was developed, which allowed it to follow all points of the path found with A^* . However, this path did not contemplate the non-holonomic constraint of the robot; for this reason, it was necessary to implement a path adjustment by corner trimming to the path found.

¹² To establish the coordination algorithm between several robots, a coupled nonlinear control was proposed; which was first implemented in simulation for the navigation of two robots on a path found with A^* . Subsequently, it was carried out experimentally from using ArUco markers to establish the detection of the robots in the structured environment.

¹⁶ A total of six experiments were performed, where the results showed successfully how the coordinated navigation was able to follow the sequence of points obtained in the routes found with A^* . In the experiments, the navigation environment, the number of robots and the connections between them were varied.

²⁰ As future work, it is intended to add constraints to the robots to achieve obstacle avoidance. On the other hand, it is intended to implement polynomial approximation algorithms to convert the routes found into continuous trajectories and thus implement a consensus control for trajectory tracking.

Índice general

2	Lista de Figuras	XII
3	1. Introducción	1
4	1.1. Motivación	1
5	1.2. Estado del arte	1
6	1.3. Definición del problema	3
7	1.4. Hipótesis	3
8	1.5. Objetivos	4
9	1.5.1. Objetivo general	4
10	1.5.2. Objetivos específicos	4
11	1.6. Contribución	4
12	1.7. Organización de la tesis	4
13	2. Preliminares	7
14	2.1. Consenso	7
15	2.2. Sistemas de integrador simple	7
16	2.2.1. Prueba de estabilidad	8
17	2.3. Distancias relativas	9
18	2.4. Consenso usando una referencia	10
19	2.5. Búsqueda de rutas	11
20	2.6. Búsqueda en anchura	11
21	2.7. Dijkstra	12
22	2.8. Algoritmo A^*	12
23	2.8.1. Descripción de A^*	14
24	2.8.2. Ejemplo de funcionamiento	16
25	2.9. Algoritmo A^* para múltiples agentes.	17
26	3. Control	19
27	3.1. Modelo cinemático	19
28	3.1.1. Restricción no holónoma	20
29	3.1.2. Mapeo a velocidades de rueda	21
30	3.2. Diseño del control	21
31	3.2.1. Prueba de estabilidad	23

1	3.2.2. Regulación a un punto	24
2	3.3. Ajuste de la ruta calculada con A*	26
3	3.4. Navegación con puntos de A*	27
4	4. Coordinación de robots móviles en ambientes estructurados	31
5	4.1. Consenso con A*	31
6	4.1.1. Diseño de control acoplado	31
7	4.1.2. Prueba de estabilidad	33
8	4.1.3. Simulaciones	36
9	4.2. Entorno experimental	39
10	4.2.1. Entorno de navegación	40
11	4.2.2. Sistema robótico	41
12	4.2.3. Retroalimentación con marcadores ArUco	42
13	4.3. Experimentos	46
14	4.3.1. Dos robots	48
15	4.3.1.1. Experimento 1	49
16	4.3.1.2. Experimento 2	51
17	4.3.2. Tres robots	54
18	4.3.2.1. Experimento 3	54
19	4.3.2.2. Experimento 4	57
20	4.3.3. Cuatro robots	60
21	4.3.3.1. Experimento 5	60
22	4.3.3.2. Experimento 6	63
23	5. Conclusiones y trabajo futuro	67
24	5.1. Conclusiones	67
25	5.2. Trabajo futuro	68
26	Apéndices	71
27	A. Publicaciones producidas durante la tesis	71
28	B. Velocidades experimentales de las ruedas de los robots	73
29	Bibliografía	77

Índice de figuras

2	2.1. Ejemplo de algoritmo BFS.	11
3	2.2. Grafo de algoritmo Dijkstra.	12
4	2.3. Ejemplo del algoritmo A*.	16
5	2.4. Ejemplo de ejecución del algoritmo.	17
6	2.5. Ejemplo de ejecución del algoritmo para múltiples agentes.	18
7	3.1. Diagrama de robot móvil diferencial.	19
8	3.2. Simulación de diseño de control para un robot.	25
9	3.3. Comparación de ruta con A^* original y ajustada.	26
10	3.4. Navegación con ruta ajustada de A^*	28
11	3.5. Explicación del comportamiento de la navegación.	29
12	3.6. Diseño de control para múltiples puntos de A^*	30
13	4.1. Ruta de A^* con puntos intermedios para dos robots uniciclo acoplados.	37
14	4.2. Capturas del visualizador para la navegación de dos robots.	38
15	4.4. Control y errores para múltiples puntos de A^* para dos agentes. La Figura (4.4a) indica la convergencia al origen de los errores e_x conforme se asigna un nuevo punto de la ruta y, la oscilación de los errores e_y con respecto al origen, debido a los cambios de orientación.	39
16	4.5. Sistema de visión para entorno experimental.	40
17	4.6. Entorno virtual de navegación.	40
18	4.7. Robots para parte experimental.	41
19	4.8. Funcionamiento del control de bajo nivel.	42
20	4.9. Ejemplo de marcadores ArUco.	42
21	4.10. Implementación de marcadores ArUco	43
22	4.11. Marcador ArUco con vectores para ubicar su centro.	44
23	4.12. Entorno de navegación con robot móvil.	44
24	4.13. Funcionamiento de control de alto nivel.	46
25	4.14. Rutas obtenidas con A^* para experimentos.	46
26	4.15. Topologías de comunicación entre cada par de robots.	47
27	4.16. Diagrama de convergencia del robot a uno de los puntos de la ruta de A^*	48
28	4.17. Entorno para experimentos 1 y 2.	48
29	4.18. Topología y rutas recorridas del experimento 1.	49
30	4.19. Entradas de control del experimento 1.	50

1	4.20. Errores de navegación del experimento 1.	51
2	4.21. Topología y rutas recorridas del experimento 2.	52
3	4.22. Entradas de control del experimento 2.	53
4	4.23. Errores de navegación del experimento 2.	53
5	4.24. Entorno para experimentos 3 y 4.	54
6	4.25. Topología y rutas recorridas del experimento 3.	55
7	4.26. Entradas de control del experimento 3.	55
8	4.27. Errores de navegación del experimento 3.	56
9	4.28. Topología y rutas recorridas del experimento 4.	57
10	4.29. Reacomodo de robots para experimento 4.	58
11	4.30. Entradas de control del experimento 4.	58
12	4.31. Errores de navegación del experimento 4.	59
13	4.32. Entorno para experimentos 5 y 6.	60
14	4.33. Topología y rutas recorridas del experimento 5.	61
15	4.34. Entradas de control del experimento 5.	61
16	4.35. Errores de navegación del experimento 5.	62
17	4.36. Topología y rutas recorridas del experimento 6.	63
18	4.37. Entradas de control del experimento 6.	64
19	4.38. Errores de navegación del experimento 6.	65
20	B.1. Velocidades de rueda izquierda y derecha de los robots en el experimento 1.	73
21	B.2. Velocidades de rueda izquierda y derecha de los robots en el experimento 2.	74
22	B.3. Velocidades de rueda izquierda y derecha de los robots en el experimento 3.	74
23	B.4. Velocidades de rueda izquierda y derecha de los robots en el experimento 4.	75
24	B.5. Velocidades de rueda izquierda y derecha de los robots en el experimento 5.	75
25	B.6. Velocidades de rueda izquierda y derecha de los robots en el experimento 6.	76

¹ Capítulo 1

² Introducción

³ 1.1. Motivación

⁴ La globalización y producción en masa de las industrias, ha llevado a la automatización a
⁵ la necesidad de implementar sistemas cada vez más complejos para lograr los objetivos de pro-
⁶ ducción. La robótica, al ser el pilar fundamental de los procesos de manufactura ha llevado a
⁷ investigar y desarrollar estrategias de navegación más efectivas en robots móviles en ambien-
⁸ tes estructurados; dentro de las cuales se encuentran los sistemas coordinados compuestos por
⁹ múltiples robots, que permiten tener mayor capacidad y versatilidad en la asignación de tareas
¹⁰ complejas, que involucran un transporte de bienes en los centros de distribución y almacenaje.
¹¹ Como ejemplos de aplicación de estos sistemas multirobóticos, se encuentran los procesos de
¹² producción que involucran recolección o entrega de bienes en un almacén o transporte de mate-
¹³ rial entre líneas de producción en pasillos de difícil acceso para montacargas o trabajadores. Por
¹⁴ lo cual, el presente trabajo propone una estrategia de navegación coordinada de robots móviles
¹⁵ en ambientes estructurados mediante un algoritmo de búsqueda de rutas libres de obstáculos.
¹⁶ La coordinación implementada se realizó mediante ingeniería de control basada en consenso,
¹⁷ el cual es clave para establecer la comunicación entre los diferentes agentes de un sistema y
¹⁸ desempeñar una tarea de forma colectiva. De igual manera, en este proyecto, se trabaja la na-
¹⁹ vegación desde la perspectiva del desplazamiento autónomo de robots móviles diferenciales
²⁰ en pasillos, debido a que los entornos de navegación se desarrollaron a partir de asemejar un
²¹ almacén industrial a escala.

²² 1.2. Estado del arte

²³ Uno de los ejemplos más representativos de automatización con robots móviles surge en el
²⁴ año 2012, cuando Amazon adquirió la empresa Kiva Systems [D'Andrea, 2012], con la cual,
²⁵ pudo lograr una automatización de sus centros de distribución, que le permitieron a la compañía

1 satisfacer la sobre demanda gracias a la construcción de almacenes automatizados. En este
2 ejemplo, los robots son capaces de realizar tareas de entrega y devolución desde los racks que
3 contienen los productos hacia los puntos de destino, en donde los trabajadores se encargan de
4 completar los pedidos. Esto disminuyó de forma considerable la labor humana; sin embargo,
5 una de las limitantes más significativas en la automatización de estos centros de distribución es
6 que los robots operan mediante rutas fijas; debido a que tienen una cámara en la parte inferior
7 que les permite leer y seguir una serie de pegatinas en el suelo hasta los puntos de interés
8 [Li and Liu, 2016].

9 Posterior a la empresa Kiva Systems, han surgido investigaciones enfocadas en la coordi-
10 nación de robots. Dentro de los cuales, en diversos proyectos se han utilizado algoritmos de
11 control enfocados a consenso, el cual es un término que se fundamenta en la teoría de grafos
12 para establecer la coordinación de los robots mediante interconexiones establecidas entre ellos.
13 Las investigaciones de consenso se han trabajado desde un enfoque teórico a nivel partícula, co-
14 mo se presenta en [Duo et al., 2021] que hace un estudio de consenso de segundo orden donde
15 se construyen las condiciones necesarias para analizar el algoritmo de control con y sin satura-
16 ción de entrada; o bien, el presentado por [Zhong and Lv, 2021], que estudia el consenso a partir
17 de proponer un algoritmo que vuelve más exacta la convergencia al valor final designado por
18 las condiciones iniciales de los agentes en el grafo. A nivel experimental con robots móviles,
19 se encuentran trabajos como el realizado por [Ning et al., 2020], donde se presenta el consenso
20 de seguimiento de un líder para múltiples robots móviles bajo un grafo directo. Los robots son
21 capaces de establecer patrones complejos de formación alrededor del robot líder. Sin embargo,
22 este trabajo está más enfocado en establecer formaciones que en implementar una estrategia
23 de navegación. Por otro lado, se encuentra el trabajo de [Koung et al., 2020], que presenta un
24 algoritmo de consenso con evasión de obstáculos, donde los robots son capaces de seguir una
25 formación, no obstante la limitante es que se trabajó en un entorno estructurado muy reduci-
26 do en obstáculos; además de que no se implementó un algoritmo de búsqueda de rutas para la
27 evasión de obstáculos.

28 El presente proyecto de tesis no sólo involucra aspectos de coordinación, sino también re-
29 quiere de una planificación de rutas para poder establecer la navegación en el ambiente estructu-
30 rado. Existen trabajos con algoritmos de planificación de rutas, como lo es el algoritmo Dijkstra
31 presentado en [Alyasin et al., 2019], el cual encuentra el camino más corto (mínimo costo) en-
32 tre el nodo fuente y el nodo de destino. La limitante de este proyecto, es que el grafo se diseñó
33 con pegatinas en el suelo; esto quiere decir que el robot tiene sensores abordo para seguir la
34 ruta procesada, además de que se habla poco del control ocupado. Existen otros trabajos con
35 el algoritmo implementado en este proyecto de tesis para la planificación de rutas, el cual es el
36 algoritmo A^* , como el presentado en [Kuang et al., 2022], que propone optimizar el algoritmo
37 de A^* utilizando un algoritmo de cartografía y mapas topológicos. Al igual que el presentado en

1 [Wang, 2021], que optimiza el algoritmo de A^* con base en un algoritmo bio-inspirado basado
2 en colonia de hormigas, para resolver la secuencia transversal a el punto de destino.

3 El algoritmo A^* ha sido ampliamente utilizado, no sólo para la planificación de rutas de
4 un robot, sino también como solucionador para la búsqueda de caminos de multi agentes o
5 *Multi-Agent Path Finding* (MAPF). [Grenouilleau et al., 2019] extendieron el algoritmo clásico
6 de A^* para permitir la búsqueda de caminos con varias metas ordenadas y generar con esto, una
7 propuesta para resolver el problema de entrega y devolución. También se encuentra el presen-
8 tado por [Liu et al., 2019], que propone un algoritmo cooperativo basado en A^* que introduce
9 costos adicionales de búsqueda de rutas para lograr que los robots naveguen a base de menos
10 giros y lograr una superposición entre caminos. Asimismo, existen trabajos que han dado solu-
11 ción al MAPF de diferentes maneras, como el presentado en [Chudý et al., 2020], que propone
12 dar solución a MAPF a partir de un control por reflejo de los robots a través de la animación
13 de trayectorias en una superficie tipo pantalla, la cual siguen los robots de forma individual.
14 También, se encuentra el presentado por [Hönig et al., 2019] que implementa un grafo de de-
15 pendencia de acciones para una ejecución robusta del MAPF en tiempo continuo; permitiendo
16 una replaneación y ejecución simultánea que evita el paro de algún robot. Sin embargo, estos
17 trabajos coordinan a los robots a partir de asignarle una ruta individual a cada uno. Por lo cual,
18 no hacen referencia a un principio de coordinación basado en consenso, como el que se propone
19 en este proyecto.

20 1.3. Definición del problema

21 La coordinación de robots en almacenes requiere una estrategia persistente en la ejecución
22 de multitareas y evasión de obstáculos, sin implicar un incremento significativo en la comple-
23 jidad del algoritmo cuando se añaden más robots a la planificación de rutas u obstáculos al
24 ambiente.

25 1.4. Hipótesis

26 La implementación de estrategias híbridas que integran el control basado en consenso con
27 los algoritmos de búsqueda de rutas, mejoraron la eficiencia en la coordinación de sistemas
28 multirobóticos en ambientes estructurados.

1.5. Objetivos

2 1.5.1. Objetivo general

3 Desarrollar una estrategia para coordinar de manera eficiente un grupo de robots móviles
4 para mover bienes en un ambiente estructurado.

5 1.5.2. Objetivos específicos

- 6 1. Desarrollar una estrategia híbrida de movimiento de robots.
- 7 2. Desarrollar una ley de control que ejecute lo obtenido en el punto 1.
- 8 3. Implementar un algoritmo de búsqueda de rutas.
- 9 4. Probar el sistema a nivel simulación.
- 10 5. Desarrollar el entorno de navegación estructurado.
- 11 6. Construir los robots.
- 12 7. Probar el sistema a nivel experimental.

13 1.6. Contribución

14 Esta tesis propone una estrategia híbrida de coordinación y navegación de robots móviles
15 diferenciales, en ambientes estructurados, mediante la integración de algoritmos de control y de
16 búsqueda de rutas libres de obtáculos; específicamente la integración de la técnica de consenso
17 con A^* (ver publicaciones producidas durante la tesis en el Apéndice A).

18 1.7. Organización de la tesis

19 El presente trabajo comienza con los preliminares para el desarrollo del proyecto. Esta sec-
20 ción se divide en los fundamentos para establecer el consenso entre los agentes y por ende la
21 coordinación, junto con los algoritmos de búsqueda de rutas, en donde se exponen algunos de
22 los algoritmos más usados en la exploración y búsqueda de caminos de menor costo en un grafo.
23 Además del algoritmo implementado en este proyecto: A^* .

24 Posteriormente, el capítulo 3, expone el diseño del control de regulación no lineal para un
25 robot, junto con el algoritmo de ajuste de rutas propuesto para posprocesar la ruta no derivable
26 de A^* ; para así, integrar estos dos elementos en una propuesta de navegación para un solo robot.

- 1 Más adelante, en el capítulo 4, se integra la ruta procesada obtenida con A^* y el diseño del
- 2 control no lineal con acoplamiento; que permite que los agentes se sincronicen y sean capaces
- 3 de navegar de forma coordinada sobre la ruta encontrada y con esto exponer el diseño del
- 4 entorno experimental, el lazo cerrado de todo el sistema de navegación y los robots utilizados.
- 5 Por último, se presentan los seis experimentos realizados; donde se varió la cantidad de robots
- 6 (dos, tres y cuatro), el entorno de navegación y las interconexiones entre ellos. Finalizando el
- 7 trabajo con las conclusiones y trabajo futuro.

¹ Capítulo 2

² Preliminares

³ 2.1. Consenso

⁴ El término consenso hace referencia a un acuerdo entre los diferentes agentes de un sistema.
⁵ Los cuales deben de realizar una o más tareas de forma colectiva, mediante un intercambio de
⁶ información regido por una topología de conexiones. Esto les permite a cada uno formar parte
⁷ del objetivo a cumplir en la actividad que se está desempeñando de forma coordinada. Por
⁸ lo cual, en este capítulo se presentan los fundamentos del consenso en el control cooperativo
⁹ enfocado a los integradores de primer orden y topologías constantes (no variantes en el tiempo).

¹⁰ Con el objetivo de modelar las interacciones entre los agentes, en consenso, se utiliza un
¹¹ grafo representado de la forma: $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$, donde $\mathcal{V} = \{1, \dots, n\}$ es el conjunto de nodos
¹² y $\mathcal{E} = \{(i, j) | i, j \in V\}$ es el conjunto de aristas. Cada nodo puede representar un sistema
¹³ completo; e.g. un sistema caótico, una computadora, un robot, entre otros. Por otro lado, las
¹⁴ aristas se utilizan para modelar las interconexiones entre los nodos y son expresadas de la forma:
¹⁵ (i, j) , donde i es un vecino de j . Para modelar las interconexiones entre todos los nodos, existe
¹⁶ la matriz de adyacencia $\mathbf{A}(\mathcal{G}) = [a_{ij}] \in \mathbf{R}^{n \times n}$, donde $a_{ij} = 1$ si $(j, i) \in \mathcal{E}$ y $a_{ij} = 0$ si sucede
¹⁷ lo contrario. Por otro lado, la matriz Laplaciana no simétrica se representa como $\mathbf{L}(\mathcal{G}) = [l_{ij}] \in$
¹⁸ $\mathbf{R}^{n \times n}$, donde $l_{ij} = -a_{ij}$, $i \neq j$; $l_{ii} = \sum_{j=1, j \neq i}^n a_{ij}$.

¹⁹ 2.2. Sistemas de integrador simple

²⁰ En los sistemas de integrador de primer orden cada nodo del grafo es representado por la
²¹ siguiente ecuación:

$$\dot{\xi}_i = u_i \quad i = 1, \dots, n; \tag{2.1}$$

1 donde $\xi_i \in R^m$ es el estado y $u_i \in R^m$ es la entrada de control representada de la forma:

$$2 u_i = - \sum_{j \in \mathcal{N}_i(t)} a_{ij} (\xi_i - \xi_j). \quad (2.2)$$

3 En la ecuación (2.2), el valor a_{ij} representa la entrada (i, j) de la matriz de adyacencia A_n .

4 El cual se toma como $a_{ij} = 1$ si existe una conexión entre el nodo i y el nodo j , de lo contrario
5 entonces $a_{ij} = 0$. Aunado a eso, $j \in \mathcal{N}_i(t)$ hace referencia al conjunto de nodos de los cuales
6 el nodo i está recibiendo información. Por lo cual, el consenso es alcanzado si a partir de la
condición inicial de $\xi_i(0)$, se obtiene que $\xi_i(t) \rightarrow \xi_j(t)$ conforme $t \rightarrow \infty$.

7 El sistema compuesto por las ecuaciones (2.1) y (2.2) puede escribirse en lazo cerrado de
8 forma matricial como:

$$9 \dot{\xi}(t) = -[\mathcal{L}_n(t) \otimes I_m] \xi(t); \quad (2.3)$$

10 donde $\xi = [\xi_1^\top, \dots, \xi_n^\top]^\top$, \mathcal{L}_n es la matriz Laplaciana, \otimes denota el producto de Kronecker e I_m
11 es la matriz de identidad $m \times m$. Por otro lado, en un grafo fijo (no variante en el tiempo), el
12 consenso es alcanzado sí y sólo sí el grafo proviene de un árbol expandido directo. A partir de
13 esto, la condición inicial de un nodo va a contribuir al valor del equilibrio del sistema siempre
y cuando, el nodo tenga una ruta directa a los demás nodos [Randal Beard, 2007].

14 2.2.1. Prueba de estabilidad

15 Para la prueba de estabilidad, además de que las gráficas directas provengan de un árbol ex-
16 pandido, también requieren que el sistema en lazo cerrado representado por la ecuación (2.3) sea
17 estable. Esto se logra sí y sólo sí la matriz Laplaciana tiene valores propios definidos positivos.
18 Para mostrar esto, se retoma la ecuación (2.3) de forma general como:

$$19 \dot{\xi}(t) = -A\xi(t). \quad (2.4)$$

20 Se procede a resolver la ecuación diferencial de primer orden (2.4) por el método de varia-
bles separables:

$$21 \frac{d\xi(t)}{dt} = -A\xi(t) \Rightarrow \frac{d\xi(t)}{\xi(t)} = -Adt.$$

22 Integrando:

$$23 \ln |\xi(t)| = -At;$$

24 la solución del sistema es entonces:

$$25 \xi(t) = e^{-At}.$$

1 Se observa que, para que sea estable la ecuación diferencial que representa el lazo cerrado
 2 del sistema y por ende se llegue al consenso, se debe de cumplir que $A > 0$. Por lo cual, la
 3 matriz Laplaciana para el sistema propuesto cumple que tiene un eigenvalor en cero y todos los
 4 demás mayores que cero.

5 2.3. Distancias relativas

6 Considerando la dinámica de integradores simples (2.1) y el algoritmo de consenso (2.2). Es
 7 posible obtener diferentes resultados de convergencia; por ejemplo que cada uno de los agentes
 8 de una formación no convergan al mismo valor, sino a un valor designado por un desfase entre
 9 ellos. Este tipo de consenso es fundamental cuando se quiere implementar de forma experimen-
 10 tal una serie de sistemas coordinados, ya que dos agentes no pueden ocupar un mismo lugar.
 11 Esta convergencia puede representarse de la forma:

$$\xi_i - \xi_j \rightarrow \Delta_{ij} ; \quad (2.5)$$

12 donde Δ_{ij} representa el desfase entre ξ_i y ξ_j . Para esto, se puede aplicar el siguiente algorit-
 13 mo de control:

$$14 u_i = \dot{\delta} - \sum_{j=1}^n a_{ij}(t)[(\xi_i - \xi_j) - (\delta_i - \delta_j)], \quad i = 1, \dots, n ; \quad (2.6)$$

15 con $\Delta_{ij} \triangleq \delta_i - \delta_j, \forall i \neq j$.

16 A partir de las ecuaciones (2.1) y (2.6) y definiendo el error entre un estado ξ y una separa-
 17 ción δ como:

$$\hat{\xi}_i \triangleq \xi_i - \delta_i .$$

18 Se establece un procedimiento para obtener una expresión del sistema a lazo cerrado en
 19 donde no se converja al estado ξ_i sino al error entre ξ_i y δ_i . Por lo cual, se tiene que:

$$\dot{\hat{\xi}}_i = \dot{\xi}_i - \dot{\delta}_i = u_i - \dot{\delta}_i . \quad (2.7)$$

20 Se sustituye la ecuación (2.6) en (2.7) para obtener:

$$\dot{\hat{\xi}}_i = - \sum_{j=1}^n a_{ij}(\hat{\xi}_i - \hat{\xi}_j) ; \quad (2.8)$$

21 donde: $\hat{\xi}_i = \xi_i - \delta_i$ representa el error de regulación del agente i y $\hat{\xi}_j = \xi_j - \delta_j$ hace referencia
 22 de cómo contribuyen los miembros de la formación con los que se recibe información, a partir
 23 de su error entre la posición y separación δ [Randal Beard, 2007].

2.4. Consenso usando una referencia

Si se tiene un conjunto que consiste de n vehículos. Es posible adicionar un vehículo virtual llamado $n + 1$ que actúe como líder. Este puede funcionar como centroide de una formación compuesta por los n vehículos durante la navegación.

Se denomina al vehículo líder $n + 1$, el cual contiene la información de referencia del consenso (ξ^r) tal que:

$$\xi_{n+1} \triangleq \xi^r \in R^n; \quad (2.9)$$

$$\dot{\xi}^r = f(t, \xi^r); \quad (2.10)$$

donde $f(t, \xi^r)$ es una función acotada y continua en el tiempo para que exista la derivada. Además de ser localmente Lipschitz en ξ^r . Con el objetivo de que el valor de ninguno de los estados se eleve de forma abrupta; sino que todos se mantengan continuos y uniformes.

El problema de consenso usando una referencia se resuelve cuando $\xi_i \rightarrow \xi^r$, $i = 1, \dots, n$ conforme $t \rightarrow \infty$. Por lo cual, para un problema de regulación se considera $f(t, \xi^r) \equiv 0$ en los sistemas con dinámica de tipo simple integrador compuestos por las ecuaciones (2.9) y (2.10), debido a que para trabajar regulación la derivada del líder virtual debe de ser cero ya que se pretende alcanzar un punto fijo, en vez de una trayectoria.

El algoritmo de control que resuelve el problema de regulación está dado por:

$$u_i = - \sum_{j=1}^n a_{ij} (\xi_i - \xi_j) - a_{i(n+1)} (\xi_i - \xi^r), \quad \forall i = 1, \dots, n. \quad (2.11)$$

En la ecuación (2.11), el término $\xi_i \in R^m$ es la información del estado i y a_{ij} , $i = 1, \dots, n$, $j = 1, \dots, n + 1$ es la entrada (i, j) de la matriz $\mathcal{A}_{n+1} \in R^{(n+1) \times (n+1)}$.

Este algoritmo de control además de relacionar todas las conexiones del nodo ξ_i con los nodos ξ_j , también se establece una relación de comunicación entre el nodo ξ_i y la referencia ξ^r . Por lo cual, se quiere que el error entre los estados converja a cero pero también mutuamente hacia la referencia. Entonces, considerando que ξ^r es constante. El sistema a lazo cerrado del conjunto de vehículos tomando en cuenta el líder virtual. Puede verse de forma similar al sistema (2.3), de tal forma que:

$$\dot{\bar{\xi}}(t) = -[\mathcal{L}_{n+1}(t) \otimes I_m] \bar{\xi}(t); \quad (2.12)$$

donde $\bar{\xi} = [\xi_1^\top, \dots, \xi_n^\top, \xi^r^\top]^\top$ y $\mathcal{L}_{n+1} \in R^{(n+1) \times (n+1)}$ es la matriz Laplaciana la cual representa un grafo directo que tiene un árbol expandido sí y sólo sí \mathcal{G}_{n+1} tiene un árbol expandido directo [Randal Beard, 2007].

1 2.5. Búsqueda de rutas

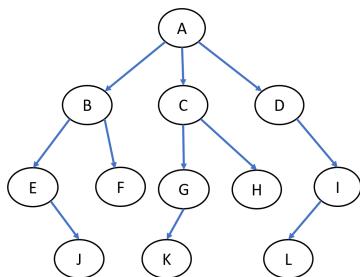
2 Así como los grafos pueden ser ocupados para modelar la topología de comunicación entre
 3 agentes y lograr el consenso. También pueden ser ocupados para definir la ruta más corta a re-
 4 correr; por medio de la implementación de un algoritmo de búsqueda de rutas. Estos algoritmos
 5 son de vital importancia para el estudio y análisis de grafos, ya que realizan una exploración
 6 hacia todos los nodos y encuentran el camino de menor costo (más corto) entre el nodo de inicio
 7 y el nodo meta. Por lo cual, desde la perspectiva de búsqueda de rutas, los nodos del grafo no
 8 representan a los agentes, sino a las posibles posiciones donde los agentes pueden ubicarse y
 9 desplazarse dentro del grafo.

10 2.6. Búsqueda en anchura

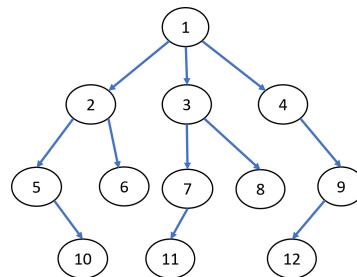
11 La búsqueda en anchura (en inglés *Breadth-First Search*, BFS) realiza una exploración de
 12 un grafo $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. El algoritmo encuentra la distancia de cada nodo alcanzable. Comenzando
 13 la exploración a partir de un nodo s definido como nodo base o inicial tal que $s \in \mathcal{V}$.

14 BFS tiene una complejidad de $\mathcal{O}(\mathcal{V} + \mathcal{E})$. Esto implica que el algoritmo se ejecuta en tiempo
 15 lineal con respecto a la cantidad de nodos y ramificaciones de \mathcal{G} . Sin embargo, la limitante de
 16 este algoritmo es que ejecuta la exploración y encuentra el mínimo camino de cada nodo al
 17 nodo actual en grafos no ponderados. Es decir, sólo de desplaza (ya sea de forma unidireccional
 18 o bidireccional) en grafos donde trasladarse a cada nodo adyacente al nodo actual no implica
 19 ningún costo [Rivest et al., 2009].

20 La Figura (2.1a) ilustra el funcionamiento de BFS; tomando al nodo A como nodo base de
 21 la exploración ramificada. Este algoritmo se ejecuta mediante una lista de adyacencia que alma-
 22 cena todos los nodos adyacentes al nodo actual; esto con el objetivo de hacer una exploración
 23 por capas del grafo. La Figura (2.1a) presenta un grafo para exploración con BFS y la Figura
 24 (2.1b) muestra la secuencia en la cual, el algoritmo explora el grafo.



(a) Grafo para BFS.



(b) Secuencia de búsqueda ramificada.

Figura 2.1: Ejemplo de algoritmo BFS.

1 2.7. Dijkstra

2 En contraste con BFS, uno de los algoritmos más populares para trabajar grafos ponderados
 3 es Dijkstra, el cual encuentra el camino de mínimo costo entre un nodo fuente, a cualquier otro
 4 nodo destino del grafo. La complejidad de este algoritmo es $\mathcal{O}(n^2)$, donde n es el número de
 5 vértices. Este algoritmo puede implementarse en problemas de planificación; tomando a los
 6 nodos del grafo como metas temporales y los vértices como las transiciones para llegar a cada
 7 una de estas metas hasta llegar al nodo deseado [Kent D. Lee, 2015].

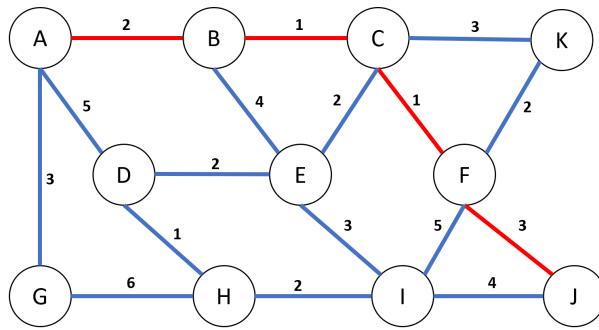


Figura 2.2: Grafo de algoritmo Dijkstra.

8 La Figura (2.2) presenta un ejemplo de aplicación del algoritmo Dijkstra en el cual se en-
 9 contró el camino de mínimo costo entre el nodo fuente (o inicial) A y el nodo de destino J . El
 10 funcionamiento de este algoritmo comienza con una exploración desde el nodo inicial hacia sus
 11 nodos adyacentes; siendo el nodo adyacente de menor costo, el nuevo nodo fuente. A partir de
 12 este nodo, se realiza nuevamente la exploración hacia sus nodos vecinos y posteriormente se
 13 selecciona nuevamente el nodo de menor costo acumulado de todo el grafo como nuevo nodo
 14 fuente. En el ejemplo de la Figura (2.2), el camino de mínimo costo es trazado con una serie de
 15 líneas rojas consecutivas que dan un costo total de siete al camino encontrado, es decir:

$$16 \quad D_{AJ} = 7 : A - B - C - F - J.$$

17 2.8. Algoritmo A^*

18 A^* es un algoritmo de búsqueda inteligente que encuentra el camino más corto en un grafo
 19 desde un nodo inicial a un nodo meta mediante una heurística óptima. El hecho de que esta
 20 heurística permita tener información no sólo del nodo anterior, sino también de la distancia con
 21 respecto al nodo destino, es lo que hace que A^* predomine sobre el algoritmo Dijkstra.

22 Para encontrar el camino libre de obstáculos, éste algoritmo debe de conocer previamente:
 23 el nodo inicial, el nodo destino y las coordenadas de todos los nodos propuestos como obstácu-
 24 los en el grafo. Esto evita que el algoritmo emprenda búsquedas a ciegas, logrando con esto que

1 se encuentren soluciones con más eficiencia. Por esta razón, se le conoce a A^* como un algoritmo respaldado por información [Takeyas, 2005]. Donde el cálculo del costo de cada nodo es asignado por la función:

$$f(n) = g(n) + h(n).$$

4 Donde:

5 • n : es el conjunto de todos los nodos adyacentes a explorar con respecto al nodo actual.

6 • $h(n)$: función heurística; encargada de calcular el costo desde el nodo actual, hasta el
7 nodo meta.

8 Las funciones heurísticas más utilizadas son las siguientes [Karur et al., 2021]:

9 Sea (x_i, y_i) la posición del agente i en un nodo en específico del grafo. Entonces se puede
10 asignar alguna de las siguientes funciones heurísticas para $h(n)$.

11 a) Distancia Euclidiana: $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

12 b) Distancia Manhattan: $|x_1 - x_2| + |y_1 - y_2|$

13 • $g(n)$: es el costo del camino entre el nodo inicial y el nodo actual. Es posible asignarle el
14 mismo comportamiento que el de la función heurística.

15 • $f(n)$: calcula el costo total del camino a partir del nodo actual.

16 El algoritmo A^* es óptimo y completo si:

17 1. Todo nodo tiene un número finito de sucesores.

18 2. El costo de cada operador es mayor que cero.

19 3. La función $h(n)$ es una heurística admisible.

20 La heurística admisible indica que nunca se debe sobreestimar el costo que implica alcanzar la
21 meta. Por otro lado, el algoritmo A^* hace referencia a ser óptimo porque encuentra la mejor
22 solución de todas las posibles. Esta propiedad se presenta de forma global siempre y cuando se
23 tenga total conocimiento del ambiente y del hecho de que los obstáculos sean estáticos. Por otro
24 lado, también se le considera un algoritmo completo, porque garantiza encontrar una solución,
25 si es que existe [Takeyas, 2005].

26 A diferencia del algoritmo BFS y el algoritmo Dijkstra, el algoritmo A^* tiene un com-
27 plejidad computacional que está directamente relacionada con la heurística que se asigne pa-
28 ra el problema. En los peores desempeños, A^* se ejecuta con una complejidad de orden ex-
29 ponencial $\mathcal{O}(2^n)$. En el mejor de los casos, A^* tiene complejidad algorítmica lineal $\mathcal{O}(n)$
30 [Savina et al., 2019].

1 2.8.1. Descripción de A^*

2 Se presentan a continuación los pasos del algoritmo A^* a modo de listado como lo presenta
3 [Lester, 2014]. La lista cerrada hace referencia a la lista donde se guarda el nodo actual. Por
4 otro lado, la lista abierta es en la cual se almacenan todos los nodos adyacentes a analizar
5 con respecto al nodo actual. De tal modo que el nodo de este grupo que tenga la menor $f(n)$,
6 cambiará a ser ahora el nodo actual y se agregará a la lista cerrada.

7 **1.-** Añadir el nodo inicial a la lista cerrada.

8 **2.-** Repetir lo siguiente:

9 **2.1.-** Buscar el nodo, de la lista abierta, con menor costo. Éste ahora será el nodo actual.

10 **2.2.-** Se manda este nodo a la lista cerrada.

11 **2.3.-** Para cada nodo adyacente al nodo actual realizar lo siguiente:

- 12 • Si no es caminable o si está en la lista cerrada, ignorarlo.
- 13 • Si no está en la lista abierta, se añade a la lista abierta. Se hace el nodo actual
14 el padre de este nodo. Se calcula y asignan las funciones f , g y h a este nodo
15 adyacente.
- 16 • Si ya está en la lista abierta, revisar si el camino con el nodo actual es mejor,
17 usando la función g como medida. Si es así, cambia el padre de ese nodo y que
18 ahora el padre sea el nodo actual, y recalcula la función f y g de ese nodo.

19 **2.4.-** Detener el algoritmo cuando:

- 20 • Se añada el nodo de la meta a la lista cerrada, en tal caso quiere decir que el
21 camino ha sido encontrado.
- 22 • O si hubo un fallo en encontrar el nodo de la meta, en este caso la lista abierta
23 está vacía y se concluye que no existe camino.

24 **3.-** Guarda el camino encontrado, empezando desde el último nodo hasta el inicio.

25 El Algoritmo 1 presenta los pasos de A^* expresados en pseudocódigo:

Algoritmo 1: Algoritmo A^*

Datos: Declaración de lista abierta y lista cerrada.

N_{actual} = nodo actual

N_{menor} = nodo con menor costo $f(n)$

$N_{adyacente}$ = nodo adyacente

lista abierta \leftarrow nodo inicial **para cada** $N_{adyacente}$ **hacer**

$N_{menor} \leftarrow$ nodo con menor $f(n)$;

$N_{actual} \leftarrow N_{menor}$;

lista cerrada $\leftarrow N_{actual}$;

para para cada $N_{adyacente}$ **a** N_{actual} **hacer**

si es obstáculo **o** está en lista cerrada **entonces**

Ignorar;

si no está en lista abierta **entonces**

lista abierta $\leftarrow N_{adyacente}$;

calcular:

$N_{adyacente} \leftarrow g(n), h(n), f(n)$

N_{actual} se vuelve N_{padre} de $N_{adyacente}$

en otro caso

evaluar con el valor de $g(n)$ si el camino con N_{actual} es mejor;

si Verdadero **entonces**

$N_{actual} \leftarrow N_{padre}$

recalcular:

$N_{actual} \leftarrow g(n), f(n)$

si se añade N_{meta} a lista cerrada **entonces**

camino ha sido encontrado

fin

si N_{meta} no se encontró **entonces**

no es posible encontrar un camino

fin

2.8.2. Ejemplo de funcionamiento

Se presenta un ejemplo del funcionamiento del algoritmo A^* . Cada nodo que constituye el grafo va a respetar la notación propuesta en la Figura (2.3a), donde el ID hace referencia a un nombre único que tendrá cada nodo, junto con sus respectivas funciones $g(n)$, $h(n)$ y $f(n)$. Por otro lado, como se observa en la Figura (2.3b), los obstáculos están dados por los nodos $\{B, C, E, J, L, Q, R, S, W, X\}$. Esto quiere decir que, el algoritmo los descarta al momento de ejecutar la búsqueda del camino.

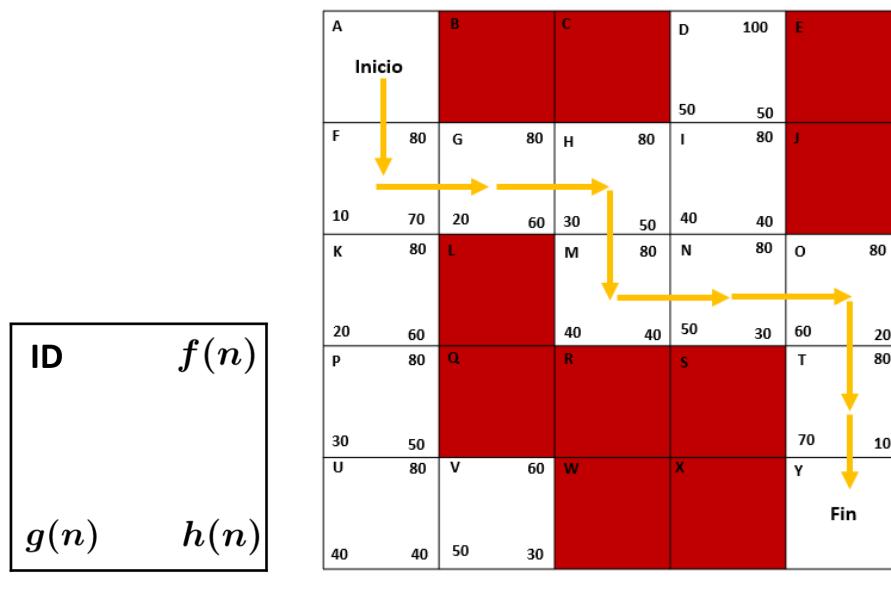


Figura 2.3: Ejemplo del algoritmo A^* .

En la Figura (2.3b), se observa que el nodo A es el nodo de inicio y el nodo Y es el nodo meta. El camino más corto encontrado entre estos dos nodos de acuerdo a su costo $f(n)$ se trazó con una serie de flechas amarillas consecutivas. De este modo, el camino está formado por los nodos: {A, F, G, H, I, N, O, T, Y}.

Se programó el diagrama presentado de la Figura (2.3b). En este grafo de 25 nodos, la posición del nodo de inicio se encuentra en la posición (5, 0). Por otro lado, el nodo final ocupa la posición (0, 5). El resultado de la búsqueda del camino libre de obstáculos se presenta en la Figura (2.4). Para la programación del algoritmo y adaptación del entorno se partió de el código explicado en [Roy, 2019].

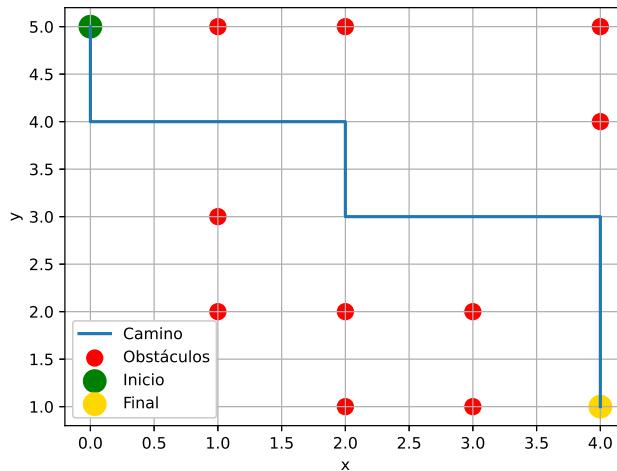


Figura 2.4: Ejemplo de ejecución del algoritmo.

1 2.9. Algoritmo A^* para múltiples agentes.

2 El algoritmo de A^* se implementa para encontrar una ruta libre de obstáculos de un sólo
 3 agente. Sin embargo, el problema de MAPF, demanda encontrar la ruta para múltiples agentes.
 4 A^* es un algoritmo que puede ser adaptado para esta tarea [Silver, 2005]. Aunado a esto, el
 5 objetivo de este proyecto de tesis consiste en situar a más de un agente en una sola ruta encon-
 6 trada mediante consenso. Esto quiere decir que una sola ruta puede ser utilizada para más de un
 7 agente; reduciendo con esto el número de rutas requeridas para la navegación coordinada.

8 Para el planteamiento de este proyecto no se requirió que la búsqueda de rutas fuera coope-
 9 rativa. Es otras palabras, para el algoritmo de A^* en multiagentes, cada ruta se calculó sin co-
 10 nocimiento de las otras rutas; sin embargo se agregaron restricciones de movimiento para que
 11 cada ruta encontrada evada no sólo a los obstáculos, sino también a las posiciones ya utilizadas
 12 por otras rutas. Esto quiere decir que la primera ruta que se encuentre es para el camino más
 13 óptimo. Posteriormente, la segunda ruta encontrada será el camino más óptimo encontrado con
 14 respecto al primero, ya que los nodos ocupados por el primer camino ya están reservados.

15 El abordar la búsqueda de caminos de multiagentes cómo se trabajó en este proyecto, tiene la
 16 ventaja de que en un entorno grande con varios caminos trazados, es posible asignar el camino
 17 más óptimo al conjunto de robots que tengan más prioridad, ya que ellos llegarán en menor
 18 tiempo a su destino. Por otro lado, si se requieren hacer tareas de entrega y devolución; esto
 19 quiere decir que cada agente puede navegar por su camino reservado en ambas direcciones, sin
 20 necesidad de ejecutar el algoritmo para encontrar una nueva ruta. La Figura (2.5) muestra el
 21 resultado de el algoritmo de A^* para múltiples agentes implementado en este proyecto, seguido

de su funcionamiento en pseudocódigo, presentado en el Algoritmo 2.

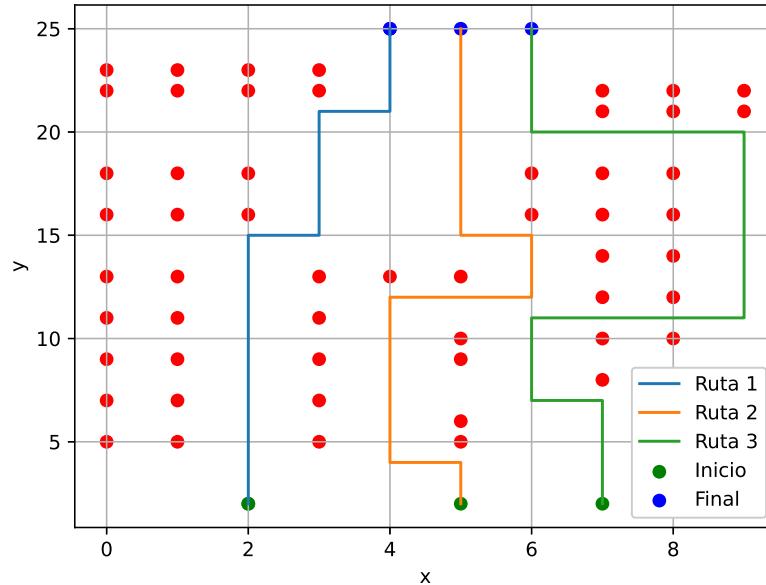


Figura 2.5: Ejemplo de ejecución del algoritmo para múltiples agentes.

Algoritmo 2: Algoritmo A^* para múltiples agentes.

Datos: Declaración de lista de obstáculos y lista de rutas.

para cada agente en lista de agentes **hacer**

Ejecutar algoritmo A^* (Algoritmo 1) ; lista de rutas \leftarrow ruta encontrada ; lista de obstáculos \leftarrow ruta encontrada ; cálculo del estado del entorno (lista de obstáculos) ; se declara el nuevo estado del entorno (en forma matricial) con la ruta encontrada ya reservada ;

fin

¹ Capítulo 3

² Control

³ 3.1. Modelo cinemático

⁴ La cinemática describe el movimiento inducido en un cuerpo sin tomar en cuenta las fuerzas
⁵ que interactúan en él. Para el caso de un robot móvil diferencial, el modelo cinemático parte
⁶ de las coordenadas generalizadas q de este sistema, el cual sólo puede desplazarse en un plano.
⁷ Esto quiere decir que la posición y orientación son descritas por el vector de coordenadas gene-
⁸ ralizadas $q = [x, y, \theta]$, donde θ es el ángulo de orientación, que representa una rotación con
⁹ respecto al eje z.

¹⁰ En la Figura (3.1) se presenta el diagrama de un robot móvil diferencial, en el cual se expresa
¹¹ su posición (x, y) en el plano, así como su orientación θ y los vectores ω (velocidad angular) y
¹² v (velocidad lineal) [Malu and Majumdar, 2014].

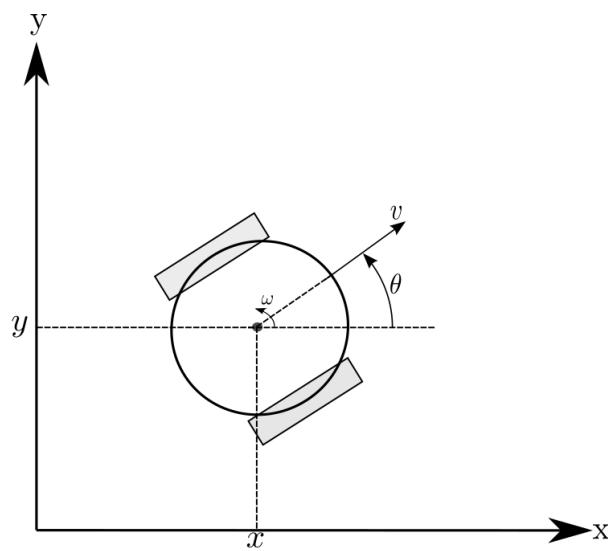


Figura 3.1: Diagrama de robot móvil diferencial.

¹ Al obtener los componentes (x , y) del vector v con respecto al ángulo θ . El modelo cinemático queda expresado de la forma:

$$\begin{aligned}\dot{x} &= v \cos(\theta) ; \\ \dot{y} &= v \sin(\theta) ; \\ \dot{\theta} &= \omega .\end{aligned}\tag{3.1}$$

³ Representado en forma matricial:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} .\tag{3.2}$$

⁴ 3.1.1. Restricción no holónoma

⁵ El modelo cinemático (3.2) tiene una restricción no holónoma. Es decir, una restricción al
⁶ no deslizamiento lateral de las ruedas; lo cual indica que el robot no puede tener movimientos
⁷ instantáneos en cualquier dirección.

⁸ Esta restricción establece que la velocidad en la componente perpendicular a la rueda es
⁹ cero. Esto es debido a que el vector que representa la velocidad lineal está expresado de la
¹⁰ forma:

$$\mathbf{v} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} ;\tag{3.3}$$

¹¹ con un vector unitario dado por:

$$\mathbf{e}_v = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} .\tag{3.4}$$

¹² De modo que la dirección perpendicular a (3.4) puede representarse como:

$$\mathbf{e}_{v\perp} = \begin{bmatrix} \sin(\theta) \\ -\cos(\theta) \end{bmatrix} .\tag{3.5}$$

¹³ Entonces, al ser la proyección de la velocidad lineal cero en sus componentes laterales. La
¹⁴ restricción no holónoma puede representarse como el producto punto de (3.3) y (3.5). Es decir
¹⁵ $\mathbf{v} \cdot \mathbf{e}_{v\perp} = 0$ [Siegwart et al., 2011]. Esto indica que:

$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0 .\tag{3.6}$$

1 3.1.2. Mapeo a velocidades de rueda

2 El estudio del modelo cinemático del uniciclo ayuda a analizar el comportamiento del sistema
 3 a partir de las entradas de control (velocidad lineal y angular) que se asignen. Sin embargo,
 4 para trabajar el sistema a nivel experimental, es necesario hacer un mapeo que transforme estas
 5 señales de control a velocidades de rueda izquierda y derecha como se observa a continuación
 6 [Oriolo et al., 2002] :

$$v = \frac{R(\omega_r + \omega_l)}{2}, \quad \omega = \frac{R(\omega_r - \omega_l)}{L}; \quad (3.7)$$

7 donde R es el radio de la rueda y L es la distancia entre los centros de las ruedas. Además, ω_r
 8 y ω_l son las velocidades de rueda derecha e izquierda respectivamente. Si se expresa (3.7) de
 9 forma matricial se obtiene que:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix}; \quad (3.8)$$

10 por lo cual, la matriz Jacobiana queda expresada de la forma:

$$\mathbf{J} = \frac{R}{2L} \begin{bmatrix} L & L \\ 2 & -2 \end{bmatrix}. \quad (3.9)$$

11 Debido a que las señales de control se representan en velocidad lineal y angular, entonces
 12 el mapeo para velocidades de rueda izquierda y derecha queda expresado de la forma:

$$\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (3.10)$$

13 3.2. Diseño del control

14 El diseño del control para este proyecto requiere que, a partir de las entradas del sistema, el
 15 robot pueda desplazarse y ubicarse en el punto deseado mediante un comportamiento determinado.
 16 Por lo cual, para abordar el problema de navegación que se expone en este proyecto, se
 17 propone el siguiente control de regulación:

$$v = k_x e_x; \quad (3.11)$$

$$\omega = \frac{\sin(e_\theta)}{e_\theta} k_y e_y + k_\theta e_\theta. \quad (3.12)$$

1 El término no lineal $\frac{\sin(e_\theta)}{e_\theta}$ suaviza la trayectoria en los cambios de dirección, conforme el
 2 robot se posiciona en la orientación deseada $e_\theta \rightarrow 0$; causando que la entrada de control (3.12)
 3 dependa únicamente de e_y . Ya que por la regla de L'Hopital se tiene:

$$\lim_{e_\theta \rightarrow 0} \frac{\sin(e_\theta)}{e_\theta} = 1. \quad (3.13)$$

4 Dado que el vector q de un robot diferencial no solamente tiene posición (x, y) , sino también
 5 orientación en el plano con respecto al eje z (ángulo de guiñada). Es necesario hacer un mapeo
 6 del eje de coordenadas del marco de referencia inercial al marco local del robot tomando en
 7 cuenta esta orientación [Siciliano et al., 2010]. Esta transformación se lleva a cabo de la forma:

$$\mathbf{e} = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix}. \quad (3.14)$$

8 Desarrollando el producto matricial de (3.14) se obtiene:

$$e_x = \cos(\theta)(x_r - x) + \sin(\theta)(y_r - y); \quad (3.15)$$

$$e_y = -\sin(\theta)(x_r - x) + \cos(\theta)(y_r - y); \quad (3.16)$$

$$e_\theta = \theta_r - \theta. \quad (3.17)$$

9 Se procede a derivar las ecuaciones (3.15), (3.16) y (3.17) para obtener la dinámica del
 10 error. Sin embargo, al tratarse de una dinámica de error para un control de regulación se tiene
 11 que $[\dot{x}_r, \dot{y}_r, \dot{\theta}_r] = [0, 0, 0]$.

12 Derivando la ecuación (3.15) se obtiene:

$$13 \dot{e}_x = -\sin(\theta)\dot{\theta}(x_r - x) + \cos(\theta)(\dot{x}_r^0 - \dot{x}) + \cos(\theta)\dot{\theta}(y_r - y) + \sin(\theta)(\dot{y}_r^0 - \dot{y})$$

$$15 \dot{e}_x = \omega[-\sin(\theta)(x_r - x) + \cos(\theta)(y_r - y)] - \dot{x}\cos(\theta) - \dot{y}\sin(\theta)$$

$$17 \dot{e}_x = \omega e_y - v \cos^2(\theta) - v \sin^2(\theta);$$

18 por lo cual, la dinámica del error en e_x queda expresada de la forma:

$$\dot{e}_x = \omega e_y - v. \quad (3.18)$$

19 Derivando ahora la ecuación (3.16) se obtiene:

$$\begin{aligned}
 1 \quad \dot{e}_y &= -\cos(\theta)\dot{\theta}(x_r - x) - \sin(\theta)(\overset{0}{\vec{x}_r} - \dot{x}) - \sin(\theta)\dot{\theta}(y_r - y) + \cos(\theta)(\overset{0}{\vec{y}_r} - \dot{y}) \\
 2 \\
 3 \quad \dot{e}_y &= -\cos(\theta)\dot{\theta}(x_r - x) + \dot{x}\sin(\theta) - \sin(\theta)\dot{\theta}(y_r - y) - \dot{y}\cos(\theta) \\
 4 \\
 5 \quad \dot{e}_y &= -\omega[\cos(\theta)(x_r - x) + \sin(\theta)(y_r - y)] + \cancel{v\cos(\theta)\sin(\theta)} - \cancel{v\cos(\theta)\sin(\theta)}; \\
 6 \\
 7 \quad \text{la dinámica del error } e_y \text{ es entonces:}
 \end{aligned}$$

$$\dot{e}_y = -\omega e_x. \quad (3.19)$$

8 Debido a que $\dot{\theta}_r = 0$. El error de orientación queda expresado como:

$$\dot{e}_\theta = -\omega. \quad (3.20)$$

9 Retomando las ecuaciones (3.18), (3.19) y (3.20). La dinámica del error es entonces:

$$\begin{cases} \dot{e}_x = \omega e_y - v \\ \dot{e}_y = -\omega e_x \\ \dot{e}_\theta = -\omega. \end{cases} \quad (3.21)$$

10 3.2.1. Prueba de estabilidad

11 A partir de la dinámica en (3.21) y las entradas de control (3.11) y (3.12), se propone la
12 función de Lyapunov con respecto a los errores de (3.21):

$$V(e_x, e_y, e_\theta) \triangleq \frac{1}{2}e_x^2 + \frac{1}{2}e_y^2 + \frac{1}{2}e_\theta^2; \quad (3.22)$$

13 derivando (3.22) se obtiene:

$$\dot{V}(e_x, e_y, e_\theta) = e_x \dot{e}_x + e_y \dot{e}_y + e_\theta \dot{e}_\theta; \quad (3.23)$$

14 entonces al sustituir (3.21) en (3.23) el resultado es:

$$\dot{V}(e_x, e_y, e_\theta) = e_x(\omega e_y - v) + e_y(-\omega e_x) - e_\theta \omega$$

$$\dot{V}(e_x, e_y, e_\theta) = \cancel{\omega e_x e_y} - ve_x - \cancel{\omega e_x e_y} - e_\theta \omega$$

$$\dot{V}(e_x, e_y, e_\theta) = -ve_x - e_\theta\omega. \quad (3.24)$$

¹ A partir de las entradas de control (3.11) y (3.12) en (3.24) se obtiene:

$$\dot{V}(e_x, e_y, e_\theta) = -ke_x^2 - \sin(e_\theta)k_ye_y - k_\theta e_\theta^2 \quad (3.25)$$

$$\begin{aligned} \dot{V}(e_x, e_y, e_\theta) &= -k_xe_x^2 - \gamma k_ye_y^2 - k_\theta e_\theta^2 + \gamma k_ye_y^2 - \sin(e_\theta)k_ye_y \\ &\leq -k_xe_x^2 - \gamma k_ye_y^2 - k_\theta e_\theta^2 + \gamma k_ye_y^2 - k_ye_y. \end{aligned} \quad (3.26)$$

² En la ecuación (3.26) se cumple que $-k_xe_x^2 - \gamma k_ye_y^2 - k_\theta e_\theta^2 \leq 0$; sin embargo, no se puede
³ garantizar que $\gamma k_ye_y^2 - k_ye_y \leq 0$. Debido a que sólo uno de los dos términos tiene el error
⁴ al cuadrado en y . Por lo cual, éste no es un control que tenga estabilidad asintótica global;
⁵ porque cumple que $V(e_x, e_y, e_\theta) \leq 0$ sí y sólo sí $e_y \geq \frac{1}{\gamma}$, donde $0 < \gamma$. Esto implica que
⁶ $\gamma k_ye_y^2 - k_ye_y = 0$. Donde $e_y \in (-\infty, 0) \cup (0, \infty)$. Esto quiere decir que, aunque $e_y \neq 0$ es
⁷ posible que $e_y \rightarrow 0$ [Khalil, 1996].

⁸ Además, para que se cumpla $\dot{V}(e_x, e_y, e_\theta) \leq 0$ es necesario que:

$$k_x > 0, k_y > 0, k_\theta > 0.$$

¹⁰ 3.2.2. Regulación a un punto

¹¹ Se realizó, a nivel simulación, el estudio del control propuesto en (3.11) y (3.12). Se ocupó
¹² el modelo cinemático (3.1) y el paso de Euler como método de integración; con un tiempo de
¹³ simulación de 60 segundos y un tiempo de muestreo de 0.1 segundos. Con condiciones iniciales:

$$x(0) = 1, y(0) = 0, \theta(0) = 0.$$

¹⁵ Se define el vector de la posición y orientación deseada del robot $[x_r, y_r, \theta_r]$ de la forma:

$$x_r = 4, y_r = 2, \theta_r = \arctan\left(\frac{y_d - y(0)}{x_d - x(0)}\right);$$

¹⁷ utilizando las siguientes ganancias:

$$k_x = 0.1, k_y = 0.5, k_\theta = 0.5.$$

- 1 Es importante recalcar que a diferencia de los errores e_x y e_y , el error de orientación e_θ es
 2 un error que siempre va a estar acotado en $-\pi < e_\theta < \pi$. Sin embargo, se espera que el robot
 3 siempre rote en la dirección de giro con menor ángulo para poder ubicarse en la orientación
 4 deseada. Esto quiere decir que si el robot demanda girar más de 180° , rotará en al dirección
 5 contraria. Para esto, es necesario entonces tomar a e_θ de (3.17) como variable independiente y
 6 una función que permita definir a un error acotado en orientación (e_{θ_a}) de la forma:

$$e_{\theta_a} = (\theta + \pi) \text{mod}(2\pi) - \pi . \quad (3.27)$$

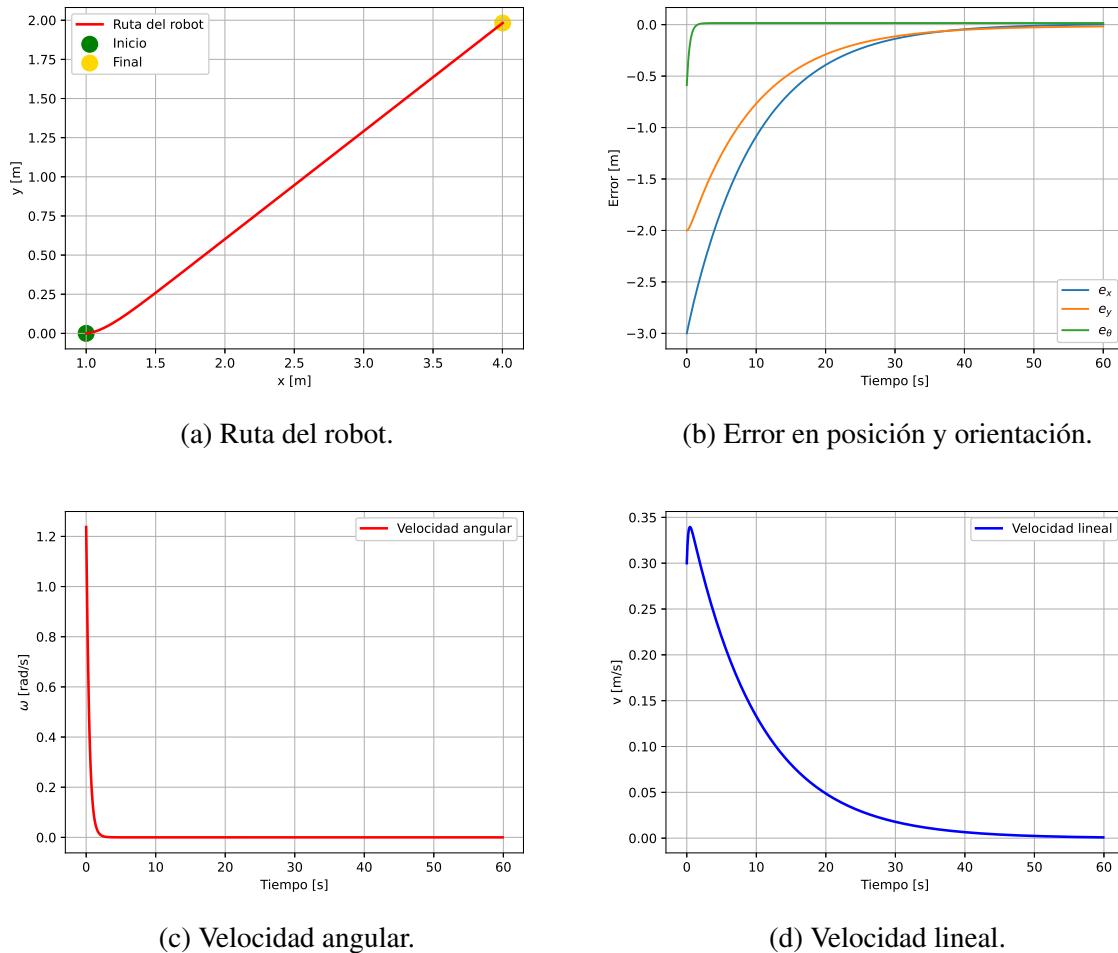


Figura 3.2: Simulación de diseño de control para un robot.

- 7 La Figura (3.2a) presenta el diagrama fase de la ruta del robot desde sus condiciones inicia-
 8 les hacia la posición y orientación deseada, donde el error de posición (x, y) y de orientación
 9 se van aproximando a cero, como se observa en la Figura (3.2b). Por otro lado, la Figura (3.2c)
 10 presenta la entrada de velocidad angular, la cual demanda un esfuerzo de control elevado al

comienzo de la simulación. Esto es debido a que el robot primero rota hacia la orientación deseada antes de comenzar a desplazarse. Por último, la Figura (3.2c) expone la velocidad lineal del sistema y cómo ésta disminuye en medida que el error en x se aproxima a cero.

3.3. Ajuste de la ruta calculada con A^*

La planificación de rutas se encarga de definir una estrategia que pueda llevar a los robots móviles a la posición y orientación deseada durante la navegación. Al utilizar el algoritmo A^* , para esta planificación, se obtienen rutas no aptas, debido a que no contempla la restricción no holónoma para los robots (3.6).

A pesar de que A^* es uno de los métodos más usados en robots móviles para encontrar un camino óptimo libre de obstáculos en ambientes estructurados [Gunawan et al., 2019]; es necesario hacer un post-procesamiento que ajuste la ruta. La finalidad de esto, es evitar que el robot de giros abruptos de 90° en cada cambio de dirección que demanden un esfuerzo de control que no sea posible de ejecutar, debido a las limitantes físicas de los actuadores.

Para eliminar los giros de 90° . Se optó por agregar puntos intermedios antes y después de cada esquina que hay en el camino encontrado; eliminando posteriormente las coordenadas (x, y) de cada esquina y dejando sólo sus puntos intermedios, obteniendo un ajuste de la ruta que reduce la energía requerida en la navegación. La Figura (3.3) muestra un conjunto de

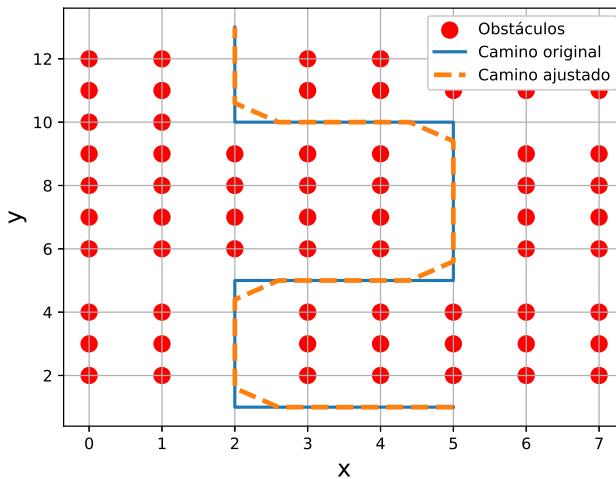


Figura 3.3: Comparación de ruta con A^* original y ajustada.

puntos rojos (obstáculos) y una ruta encontrada. El camino original (color azul), es el procesado directamente con el algoritmo de A^* . Por otro lado, el camino de color naranja, surge de

- 1 un post-procesamiento del camino original con el algoritmo de ajuste de ruta; este camino es
- 2 entonces clave para evitar un sobre esfuerzo en las entradas de control.

Algoritmo 3: Algoritmo de ajuste de ruta de A^*

Datos: des = constante de desplazamiento ;
 X = Coordenadas x de A^* ;
 Y = Coordenadas y de A^* ;
 $x_{salida} = []$;
 $y_{salida} = []$;

para cada punto (X, Y) **hacer**

- si** $X_i - X_{i-1} = 0 \&& X_{i+1} - X_i \neq 0$ **entonces**
 - si** $X_{i+1} - X_i > 0$ **entonces**
 - $x_{salida} \leftarrow X_i$;
 - $x_{salida} \leftarrow (X_i + des)$
 - en otro caso**
 - $x_{salida} \leftarrow X_i$;
 - $x_{salida} \leftarrow (X_i - des)$
 - $y_{salida} \leftarrow (Y_i - des)$
 - $y_{salida} \leftarrow Y_i$
- si** $Y_i - Y_{i-1} = 0 \&& Y_{i+1} - Y_i \neq 0$ **entonces**
 - si** $X_i - X_{i-1} > 0$ **entonces**
 - $x_{salida} \leftarrow (X_i - des)$;
 - $x_{salida} \leftarrow X_i$
 - en otro caso**
 - $x_{salida} \leftarrow (X_i + des)$;
 - $x_{salida} \leftarrow X_i$
 - $y_{salida} \leftarrow Y_i$
 - $y_{salida} \leftarrow (Y_i + des)$
- en otro caso**
 - $x_{salida} \leftarrow X_i$;
 - $y_{salida} \leftarrow Y_i$

fin

³ 3.4. Navegación con puntos de A*

- 4 Una vez encontrado el camino con A^* y el posterior ajuste de esta ruta con el Algoritmo 3.
 - 5 Se presenta la simulación de la navegación de un robot uniciclo en un ambiente estructurado.
-

Para esto se tomó a (3.1) como modelo cinemático, con (3.11) y (3.12) como las entradas de control lineal y angular respectivamente. Esto quiere decir que, a diferencia de la simulación presentada en la Figura (3.2), ahora se tiene un conjunto de puntos de regulación (x, y) que el robot debe de recorrer en el orden que establece la ruta.

La Figura (3.4) presenta el diagrama fase de la navegación del robot uniciclo en una ruta encontrada con A^* y ajustada con el Algoritmo 3. La navegación se llevó a cabo en un grafo representado por una matriz de 8 renglones y 15 columnas; con 66 nodos (puntos rojos) declarados como obstáculos. Es decir, nodos por los cuales el robot no puede pasar, donde el nodo marcado con color verde indica la posición inicial del robot y el nodo de color amarillo su posición final o meta. Las condiciones iniciales para dicha simulación fueron:

$$x(0) = 1, y(0) = 6, \theta(0) = -\pi/2; \quad (3.28)$$

utilizando las siguientes ganancias:

$$k_x = 1, k_y = 0.001, k_\theta = 5.$$

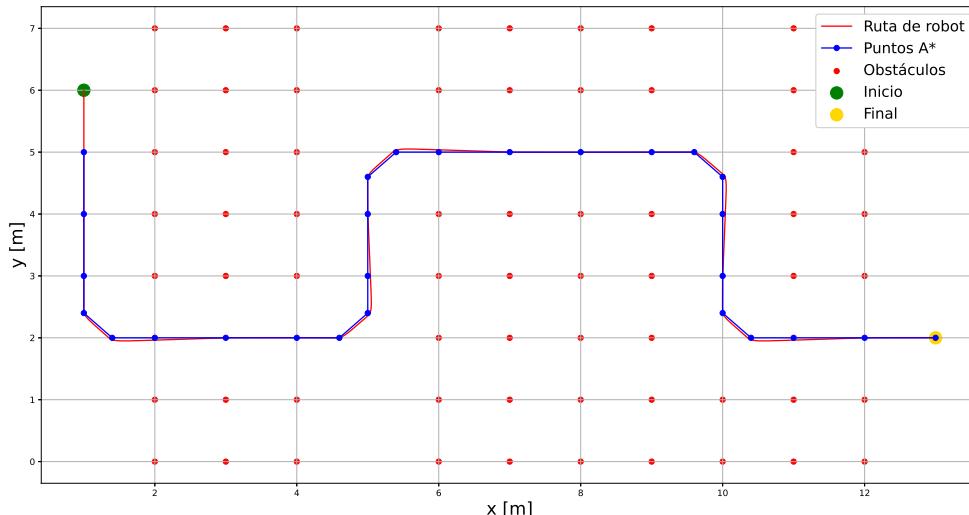
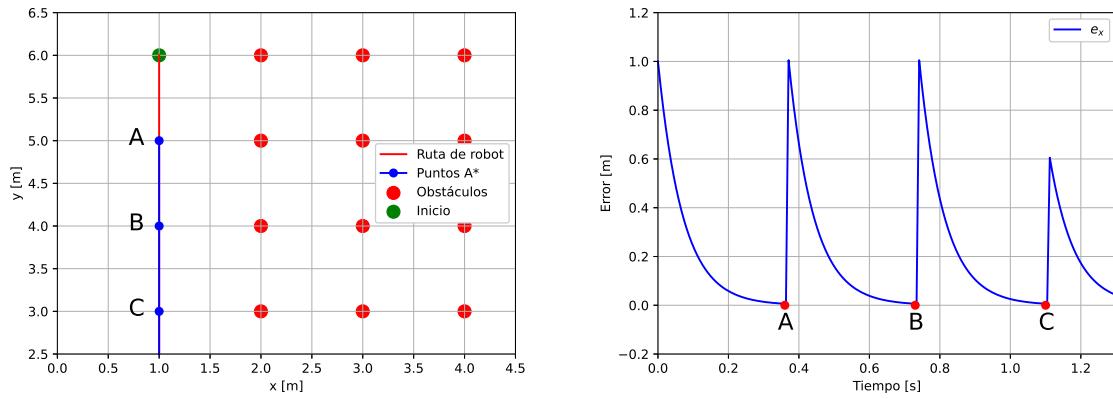


Figura 3.4: Navegación con ruta ajustada de A^* .

La Figura (3.5) presenta el comportamiento de la navegación. La Figura (3.5a) es un acercamiento de la Figura (3.4). En esta se agregó una etiqueta a los primeros tres puntos de la ruta encontrada con A^* . El punto A, es el primer punto en el que el robot se sitúa una vez que comienza la navegación. Por otro lado, se observa en la Figura (3.5b) cómo en el momento que ocurre esto, e_x converge a cero en medida que el robot llega al punto A. Posteriormente, la



(a) Acercamiento de la Figura(3.4).

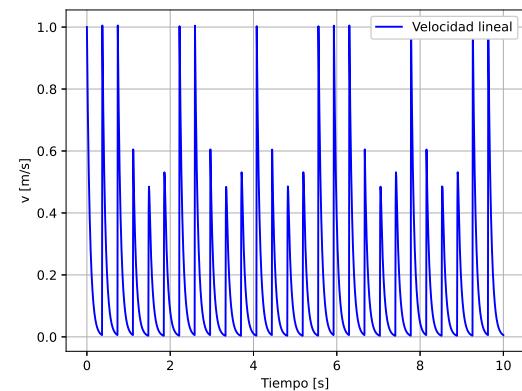
(b) Error e_x de los primeros puntos de la ruta.

Figura 3.5: Explicación del comportamiento de la navegación.

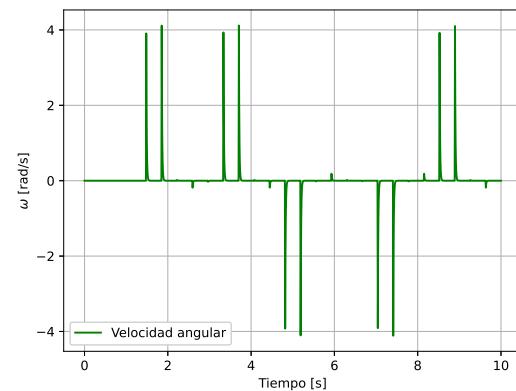
1 posición y orientación del robot en el punto A, se convierten ahora en las nuevas condiciones
 2 iniciales del robot y la nueva posición y orientación deseada es la del siguiente punto. Es decir,
 3 el punto B; por esta razón, en la Figura (3.5b) una vez alcanzada la posición del punto A, el error
 4 se eleva de forma abrupta y converge nuevamente a cero hasta llegar al punto B. Este proceso
 5 se repite para el punto C y así sucesivamente.

6 Llevando el comportamiento de la navegación explicado en la Figura (3.5) a todos los puntos
 7 de la ruta encontrada por A^* de la Figura (3.4). Se obtiene como resultado lo que se presenta
 8 en la Figura (3.6). En la Figura (3.6a) la velocidad, suministrada por la entrada de control de
 9 velocidad lineal, va decreciendo conforme el robot se aproxima a cada uno de los puntos de la
 10 ruta. Por otro lado, la Figura (3.6b) muestra la entrada de control angular. Esta consta de picos
 11 de velocidad en determinados instantes de tiempo, dados por la velocidad que el robot demanda
 12 en los cambios de orientación. La Figura (3.6c) presenta los errores en (x, y) del robot uniciclo,
 13 los cuales van disminuyendo conforme el robot se aproxima al punto de la ruta designado en
 14 determinado momento como se presentó en la Figura(3.5b). Por último, la Figura (3.6d) muestra
 15 el error de orientación y cómo tiene sobre tiros en cada cambio de dirección del robot.

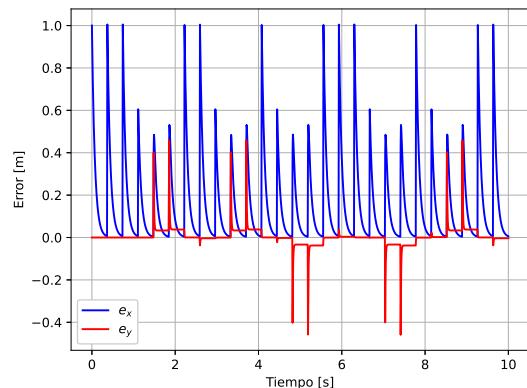
16 Se observa que el error e_x tiene un comportamiento similar a la entrada de velocidad lineal;
 17 debido a como se diseñó (3.11). Por otro lado, el error e_y y e_θ mantienen un comportamiento
 18 similar a la velocidad angular de la Figura (3.6b); debido a la relación entre dichos errores y la
 19 entrada de control angular (3.12).



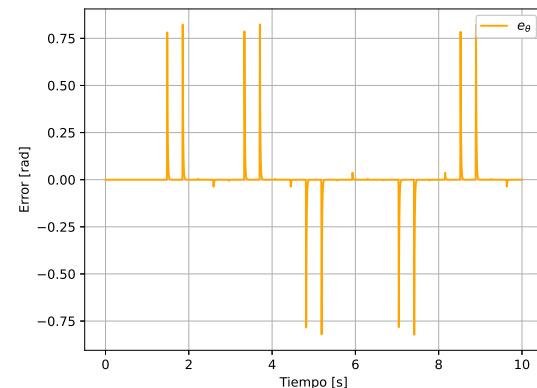
(a) Velocidad lineal.



(b) Velocidad angular.



(c) Errores de posición.



(d) Error de orientación.

Figura 3.6: Diseño de control para múltiples puntos de A^* .

¹ Capítulo 4

² Coordinación de robots móviles en ³ ambientes estructurados

⁴ 4.1. Consenso con A*

⁵ El problema de MAPF indica que, a partir de un conjunto de posiciones iniciales, cada
⁶ miembro debe de llegar a su punto de destino con base en la búsqueda de caminos libres de
⁷ obstáculos y sin colisiones entre ellos. Aunque el algoritmo A^* se ha ocupado como solucionador para MAPF, la complejidad del algoritmo crece exponencialmente conforme se incrementa el número de agentes u obstáculos en el entorno. Se han publicado nuevos algoritmos para resolver el MAPF, como lo es el algoritmo basado en la búsqueda de conflictos, conocido como Conflict-Based Search (CBS), el cual garantiza soluciones óptimas mediante una búsqueda de caminos basada en ramificaciones de restricciones para cada uno de los agentes. La ejecución de CBS se basa, al igual que A^* , en una búsqueda individual de caminos para cada uno de los agentes [Sharon et al., 2021].

¹⁵ El enfoque de consenso con A^* que se propone en este proyecto de tesis; se centra en ocupar para más de un agente las rutas libres de obstáculos encontradas. De tal modo que cada camino puede aplicarse ya sea como centroide de una formación o como base para una configuración líder-seguidor. Donde el consenso, establecido mediante la ingeniería de control, evita la colisión entre los robots al momento de navegar de forma simulánea en los caminos; reduciendo con esto la cantidad de rutas calculadas con A^* y por ende, la complejidad algorítmica de la navegación.

²¹ 4.1.1. Diseño de control acoplado

²² Dado que se requiere un sistema coordinado de navegación. El diseño del control para cada uno de los robots parte de las ecuaciones (3.11) y (3.12). Sin embargo, es necesario añadir a cada entrada de control un componente de acoplamiento en función de los errores locales de

1 cada robot, que permita establecer una comunicación e interacción cinemática entre ellos para
2 mantener la formación durante la navegación [Gutiérrez et al., 2017].

3 Es importante destacar el hecho de que en la navegación coordinada propuesta en este pro-
4 yecto. Los robots navegan y mantienen su formación a partir de los puntos que constituyen la
5 ruta calculada con A^* . En otras palabras, la ruta encontrada con A^* , es la que actúa como líder
6 virtual de la formación durante la navegación.

7 El acoplamiento ϵ_{ij} parte de los errores locales del robot i y el robot j obtenidos con las
8 ecuaciones (3.15), (3.16) y (3.17); sin embargo sólo el robot i recibe directamente la informa-
9 ción de la ruta encontrada con A^* . Esto quiere decir que, para calcular sus errores locales se
10 parte de que su pose deseada son los puntos de A^* . Por otro lado, para obtener los errores lo-
11cales del robot j , su pose deseada es la pose del robot i . Entonces, el acoplamiento entre los
12 robots, puede definirse como la diferencia entre sus errores locales de la forma:

$$\epsilon_{ij} = \begin{bmatrix} \epsilon_{x_{ij}} \\ \epsilon_{y_{ij}} \end{bmatrix} = \begin{bmatrix} e_{x_i} - e_{x_{ji}} \\ e_{y_i} - e_{y_{ji}} \end{bmatrix}; \quad (4.1)$$

13 De tal modo que, el control queda expresado como:

$$v_i = k_x e_{x_i} + u_{x_i}; \quad (4.2)$$

$$14 \quad \omega_i = \frac{\sin(e_{\theta_i})}{e_{\theta_i}} [k_y e_{y_i} + u_{y_i}] + k_{\theta_i} e_{\theta_i}. \quad (4.3)$$

15 Donde el acoplamiento se establece por:

$$u_{x_i} = C_x \sum_{j \in N_i} (\epsilon_{x_{ij}} + \Delta_{x_{i,j}}), \quad u_{y_i} = C_y \sum_{j \in N_i} \epsilon_{y_{ij}}; \quad (4.4)$$

16 El término $\Delta_{x_{i,j}}$ representa el desfase entre el robot i y el robot j . Por otro lado, C_x y C_y
17 son las constantes de acoplamiento definidas positivas.

18 Se observa en las ecuaciones de acoplamiento en (4.4), que el acoplamiento en y , expresado
19 como u_{y_i} , no involucra ningún desfase. Esto quiere decir que $\Delta_{y_{i,j}} = 0$, debido a que este aco-
20 plamiento asociado a la entrada de velocidad angular de la ecuación (4.3); conlleva solamente
21 a que los robots i y j traten de mantener la misma orientación, lo cual se logra una vez que
22 $\epsilon_{y_{ij}} = 0$, entonces $u_{y_i} = 0$.

23 El acoplamiento en x expresado como u_{x_i} de (4.4), sí involucra un desfase $\Delta_{x_{i,j}} \neq 0$. Debido
24 a que este, se encarga de que los robots mantengan una cierta distancia entre ellos una vez que
25 están alineados; por lo cual éste acoplamiento está asociado a la velocidad lineal de cada robot
26 de acuerdo a la ecuación (4.2). Entonces, una vez que se logra que $u_{x_i} = 0$, o en otras palabras

- ¹ $((e_{x_i} - e_{x_j}) + \Delta_{x_{i,j}}) = 0$. Esto indica que el acoplamiento ya no contribuye a suministrar más
² velocidad lineal al robot i ; sin embargo se mantendrá la posición designada por el valor $\Delta_{x_{i,j}}$.

³ Debido a que se trata de un desfasamiento entre los robots adyacentes una vez que están
⁴ alineados en la formación; es necesario que los desfases se asignen con respecto al líder virtual.
⁵ Es decir, la ruta de A^* . Entonces el valor del desfase de un robot es $\Delta_x > 0$ si se requiere que
⁶ se ubique en la parte delantera de cada uno de los puntos de A^* durante la navegación o un
⁷ desfase $\Delta_x < 0$ si se requiere lo contrario. Por otro lado, las entradas de control del robot j se
⁸ mantendrán de forma similar a (3.11) y (3.12).

⁹ 4.1.2. Prueba de estabilidad

¹⁰ Con base en la dinámica de error para cada agente propuesto en (3.21) y las entradas del
¹¹ control acoplado (4.2) y (4.3), se propone la función de Lyapunov:

$$V = \frac{1}{2}[e_x^\top e_x + e_y^\top e_y + e_\theta^\top e_\theta + \epsilon_x^\top \epsilon_x + \epsilon_y^\top \epsilon_y]. \quad (4.5)$$

¹² Se tiene que: $\epsilon_x = \mathcal{L}e_x$, $\epsilon_y = \mathcal{L}e_y$;

¹³ donde: $e_x = \begin{bmatrix} e_{xi} \\ e_{xji} \end{bmatrix}$, $e_y = \begin{bmatrix} e_{yi} \\ e_{yji} \end{bmatrix}$;

¹⁴ y \mathcal{L} representa la matriz Laplaciana que establece las conexiones de comunicación entre los
¹⁵ robots.

¹⁶ Al factorizar (4.5) se tiene:

$$V = \frac{1}{2}[e_x^\top (\mathbf{I} + \mathcal{L}^\top \mathcal{L})e_x + e_y^\top (\mathbf{I} + \mathcal{L}^\top \mathcal{L})e_y + e_\theta^\top e_\theta]. \quad (4.6)$$

¹⁷ La expresión (4.6) puede derivarse de la forma:

$$\dot{V} = e_x^\top (\mathbf{I} + \mathcal{L}^\top \mathcal{L})\dot{e}_x + e_y^\top (\mathbf{I} + \mathcal{L}^\top \mathcal{L})\dot{e}_y + e_\theta^\top \dot{e}_\theta; \quad (4.7)$$

¹⁸ sustituyendo la dinámica del error (3.21) en (4.7) se tiene:

¹⁹ $\dot{V} = e_x^\top (\mathbf{I} + \mathcal{L}^\top \mathcal{L})(\omega e_y - v) + e_y^\top (\mathbf{I} + \mathcal{L}^\top \mathcal{L})(-\omega e_x) + e_\theta^\top \dot{e}_\theta;$

²⁰ desarrollando términos:

²¹ $\dot{V} = \cancel{e_x^\top \omega e_y} + \cancel{e_x^\top \mathcal{L}^\top \mathcal{L} \cancel{\omega e_y}} - e_x^\top \mathbf{I} v - e_x^\top \mathcal{L}^\top \mathcal{L} v - \cancel{e_y^\top \omega e_x} - \cancel{e_y^\top \mathcal{L}^\top \mathcal{L} \cancel{\omega e_x}} - e_\theta^\top \omega.$

1 Se tiene entonces:

$$\dot{V} = -e_x^\top (\mathbf{I} + \mathcal{L}^\top \mathcal{L})(k_x e_x + C_x(\mathcal{L} e_x + \Delta_x)) - e_\theta^\top \omega. \quad (4.8)$$

2 Ya que de forma matricial el control acoplado de velocidad lineal en (4.2) puede expresarse
3 de la forma $v = k_x e_x + C_x(\mathcal{L} e_x + \Delta_x)$.

4 Al reacomodar términos, la ecuación (4.8) queda como:

$$\dot{V} = -e_x^\top (\mathbf{I} + \mathcal{L}^\top \mathcal{L})((k_x + C_x \mathcal{L})e_x + C_x \Delta_x) - e_\theta^\top \omega. \quad (4.9)$$

5 El término $-e_\theta^\top \omega$ de la ecuación (4.9) se desarrolla sustituyendo la entrada de control an-
6 gular (4.3):

$$-e_\theta^\top \omega = -e_\theta^\top \mathbf{H}(k_y \mathbf{I} + C_y \mathcal{L})e_y - e_\theta^\top k_\theta e_\theta; \quad (4.10)$$

7 donde:

$$8 \quad \mathbf{H} = \begin{bmatrix} \frac{\sin(e_{\theta i})}{e_{\theta i}} & 0 \\ 0 & \frac{\sin(e_{\theta j})}{e_{\theta j i}} \end{bmatrix}.$$

9 La ecuación (4.10) puede simplificarse de forma tal que:

$$-e_\theta^\top \omega = -M^\top (k_y \mathbf{I} + C_y \mathcal{L})e_y - e_\theta^\top k_\theta e_\theta; \quad (4.11)$$

10 donde:

$$11 \quad M = \begin{bmatrix} \sin(e_{\theta i}) \\ \sin(e_{\theta j i}) \end{bmatrix}.$$

12 Esto quiere decir que la función de Lyapunov de la ecuación (4.9), queda expresada de la
13 forma:

$$\dot{V} = -e_x^\top (\mathbf{I} + \mathcal{L}^\top \mathcal{L})((k_x + C_x \mathcal{L})e_x + C_x \Delta_x) - M^\top (k_y \mathbf{I} + C_y \mathcal{L})e_y - e_\theta^\top k_\theta e_\theta \quad (4.12)$$

$$14 \quad \leq -e_x^\top (\mathbf{I} + \mathcal{L}^\top \mathcal{L})(k_x + C_x \mathcal{L})e_x - e_x^\top (\mathbf{I} + \mathcal{L}^\top \mathcal{L})C_x \Delta_x - m^\top (k_y \mathbf{I} + C_y \mathcal{L})e_y - e_\theta^\top k_\theta e_\theta;$$

15 donde $m^\top = [1, 1]$.

16 Simplificando (4.12) con los siguientes cambios de variables:

$$17 \quad \mathbf{A} = \mathbf{I} + \mathcal{L}^\top \mathcal{L}, \quad D = k_y \mathbf{I} + C_y \mathcal{L};$$

1 se tiene:

$$\dot{V} \leq -e_x^\top A(k_x + C_x \mathcal{L}) e_x - e_x^\top A C_x \Delta_x - m^\top D e_y - e_\theta^\top k_\theta e_\theta; \quad (4.13)$$

2 entonces agregando el término $e_y^\top \Omega D e_y - e_y^\top \Omega D e_y = 0$ a la ecuación (4.13) y simplificando
3 la función de \dot{V} con las variables Φ , Ψ y Γ , se tiene:

$$\begin{aligned} \dot{V} &\leq \underbrace{-e_x^\top A(k_x + C_x \mathcal{L}) e_x - e_\theta^\top k_\theta e_\theta - e_y^\top \Omega D e_y}_{\Phi} \\ &\quad \underbrace{-e_x^\top A C_x \Delta_x + e_y^\top \Omega D e_y - m^\top D e_y}_{\Psi} ; \end{aligned} \quad (4.14)$$

4 donde Ω es una matriz diagonal con valores positivos tal que $\alpha_i, \alpha_j > 0$. Expresada de la forma:

$$5 \quad \Omega = \begin{bmatrix} \alpha_i & 0 \\ 0 & \alpha_j \end{bmatrix}.$$

6 En la ecuación (4.14) se cumple que $\Phi \leq 0$; sin embargo para el término Ψ no se puede decir lo
7 mismo, ya que no tiene un término cuadrático del error. Por lo cual, para Ψ se debe de satisfacer:

$$8 \quad \lambda_{\max}(A) \|e_x^\top\| + \gamma \|e_x^\top\| > 0 \text{ tal que } \gamma > -\lambda_{\max}(A) \|e_x^\top\|.$$

9 Esto quiere decir que a partir del eigenvalor de máximo valor de la matriz A y, la norma del
10 error e_x , va a existir un valor γ que conlleve a que $e_x^\top A C_x \Delta_x > 0$. Adicionalmente se debe de
11 cumplir que $(k_x + C_x \mathcal{L}) e_x > C_x \Delta_x$; con el objetivo de que los valores seleccionados para el
12 desfase no afecten la estabilidad del sistema; independientemente de si se selecciona un desfase
13 positivo o negativo para alguno de los robots.

14 Finalmente, no se puede garantizar que $\Gamma \leq 0$, debido a que sólo uno de los dos términos
15 tiene el error e_y al cuadrado. Por lo cual, al igual que la prueba de estabilidad para un robot, éste
16 no es un control que tenga estabilidad asintótica global, ya que se debe de cumplir que $e_{yi} \geq \frac{1}{\alpha_i}$
17 y $e_{yj} \geq \frac{1}{\alpha_j}$. Por lo cual, esto implica que $\Gamma = 0$, con $e_y \in (-\infty, 0) \cup (0, \infty)$. Entonces, aunque
18 $e_y \neq 0$, se cumple que $e_y \rightarrow 0$. Además, para que se cumpla $\dot{V} \leq 0$ es necesario que todas las
19 ganancias se definan como:

$$20 \quad k_x > 0, k_y > 0, k_\theta > 0, C_x > 0, C_y > 0.$$

4.1.3. Simulaciones

El control acoplado se trabajó a nivel simulación con dos robots. Se tomó a (3.1) como modelo cinemático para cada uno de los robots, junto con (4.2) y (4.3) como las entradas de velocidad lineal y angular respectivamente.

Se ocupó el paso de Euler como método de integración, con un tiempo de simulación de 5 segundos y un tiempo de muestreo de 0.1 segundos. Con condiciones iniciales para el robot 1:

$$x(0) = 2 , \quad y(0) = 6 , \quad \theta(0) = -\pi/2 ;$$

y el robot 2 de:

$$x(0) = 1 , \quad y(0) = 6 , \quad \theta(0) = -\pi/2 .$$

Se retoma en la Figura (4.1) la ruta ajustada de A^* previamente implementada en la Figura (3.4), donde ahora el robot 1 establece su navegación con base en una serie de puntos (x, y) de regulación dados por A^* . Sin embargo, el robot 2 mantiene solamente comunicación con el robot 1. Lo cual le permite navegar sobre la ruta de A^* ; pero manteniendo un desfase sobre esta que le permite estar alineado con el robot 1. Logrando con esto que a partir de sus condiciones iniciales, sea capaz de seguir toda la ruta encontrada libre de obstáculos.

Las ganancias del control son:

$$k_x = 1 , \quad k_y = 0.001 , \quad k_{\theta_1} = 2.2 , \quad k_{\theta_2} = 0.8 ; \quad (4.15)$$

con ganancias de acoplamiento:

$$C_x = 2 , \quad C_y = 1 . \quad (4.16)$$

La separación de los robots se designa por los valores de $\Delta_{x_{i,j}}$ en u_{ix} de la ecuación (4.4) para cada uno de los robots, donde:

$$\Delta_{12} = 0 , \quad \Delta_{21} = -0.5 \text{ m} . \quad (4.17)$$

El robot 1 tiene en su control las ganancias k_x y k_y propuestas en (4.16). Por otro lado, el robot 2, en vez de tener estas ganancias, tiene las ganancias C_x y C_y propuestas en (4.16). Se observa que $C_x > k_x$, ya que el robot 2 requiere más velocidad lineal para seguir la pose del robot 1; que el robot 1 en seguir la ruta de A^* . Aunado a esto, el valor de $C_y > k_y$ es mayor, ya que el error e_y corresponde a un error no acotado, el cual ayuda al robot 2 a girar en los cambios de dirección.

La ganancia angular k_{θ_1} corresponde al robot 1 y la ganancia angular k_{θ_2} corresponde al robot 2. Con base a esto, se propuso $k_{\theta_1} > k_{\theta_2}$, ya que una ganancia angular menor permite que

1 la ruta de ese robot sea menos recta en los cambios de orientación. Esto brinda en su navegación,
 2 un desplazamiento más curvo al dar una vuelta. De modo que, al mantener el robot 2 un desfase
 3 al navegar en la ruta de A^* . Se requiere que en las curvas este tenga un comportamiento menos
 4 recto en su desplazamiento, por medio de disminuir su ganancia angular k_{θ_2} . Esto permite que
 5 tenga más libertad de girar y evadir con menor dificultad los obstáculos en las esquinas.

6 Se observa en (4.17) que entre el robot 2 y el robot 1, el robot 2 tiene un desfase negativo,
 7 es decir: $\Delta_{21} = -0.5$ m. Lo cual conlleva a que el robot 2 se va a mantener en una posición
 8 colineal con respecto al robot 1; pero a una distancia de -0.5 m en el diagrama fase de la Figura
 9 (4.1), una vez que el robot 1 se ubicó en cada uno de los puntos encontrados con A^* .

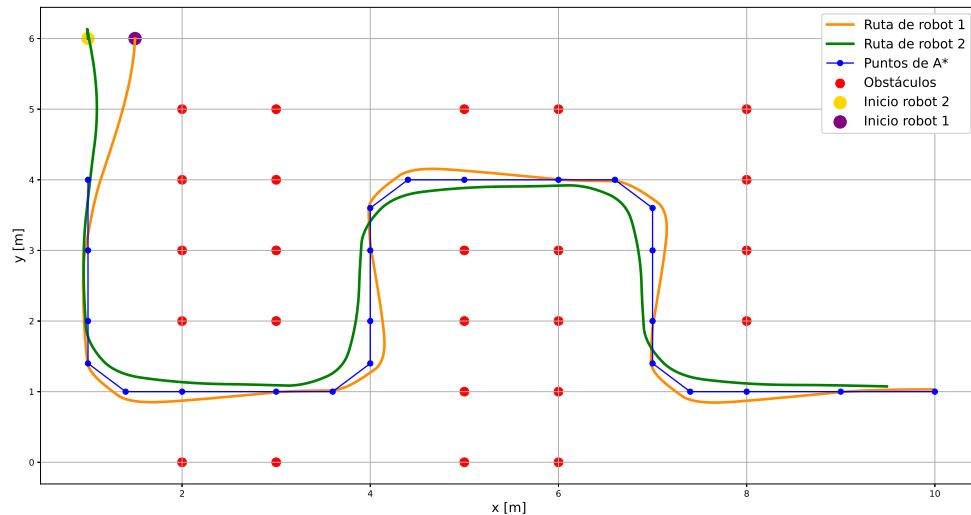


Figura 4.1: Ruta de A^* con puntos intermedios para dos robots uniciclo acoplados.

10 Con el objetivo de presentar la navegación coordinada de los dos robots. Se implementó el
 11 entorno de navegación de la Figura (4.1) con un visualizador programado en Matlab. Se presenta
 12 en la Figura (4.2) , cuatro capturas de esta navegación.

13 Las Figuras (4.3a) y (4.3b) presentan las velocidades lineales y angulares del robot 1 y 2
 14 respectivamente, donde se puede observar un comportamiento similar de estas señales entre
 15 ambos robots; ya que tienden a converger a cero una vez que se ubicaron en el punto deseado.
 16 A pesar de que el robot 2 tiene un desfase al seguir los puntos de la ruta. Este tiende a recorrerla
 17 siguiendo la pose del robot 1. Se puede observar que el robot 2 requiere menos esfuerzo en las
 18 entradas de control; dado que de acuerdo a sus ganancias, éste puede tener un desplazamiento
 19 más curvo en los cambios de dirección y el desfase de distancia para recorrer la ruta le permite
 20 demandar menos velocidad lineal.

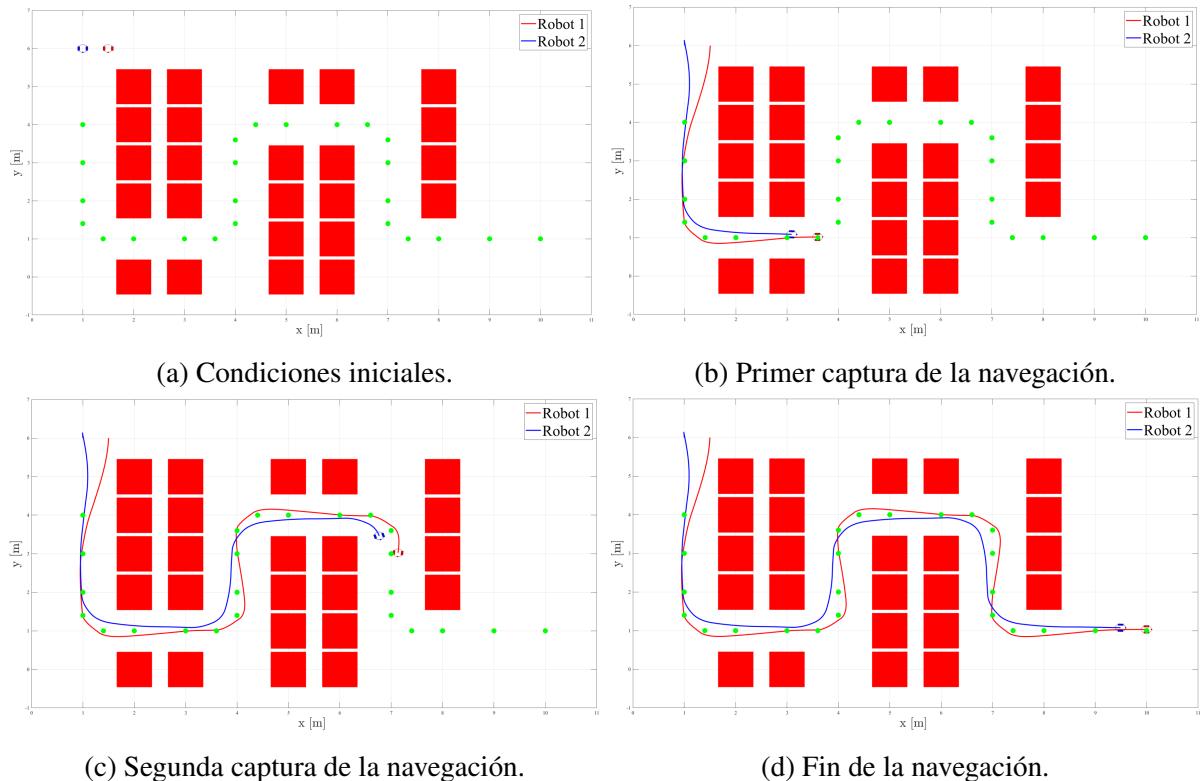


Figura 4.2: Capturas del visualizador para la navegación de dos robots.

La Figura (4.4a) muestra el comportamiento de los errores e_x y e_y de ambos robots. Es importante recalcar en esta figura cómo el error e_x (el cual está asociado con el acoplamiento u_{x_i} de (4.4)) converge a cero. Esto quiere decir que los robots convergen a los puntos de la ruta durante la navegación; pero de forma desfasada para mantener la formación y no colisionar entre ellos. Por otro lado, aunque los errores en e_y de ambos robots no guardan similitud, ambos intentan siempre llegar a cero. Esto se debe a que el error e_y no está acotado y, en los cambios de dirección, el robot 1 puede haber girado cuando el robot 2 todavía no. Sin embargo, en la Figura (4.4b) se observa el error en orientación, el cual procura siempre converger a cero en cada cambio de dirección y presenta un comportamiento similar para ambos robots.

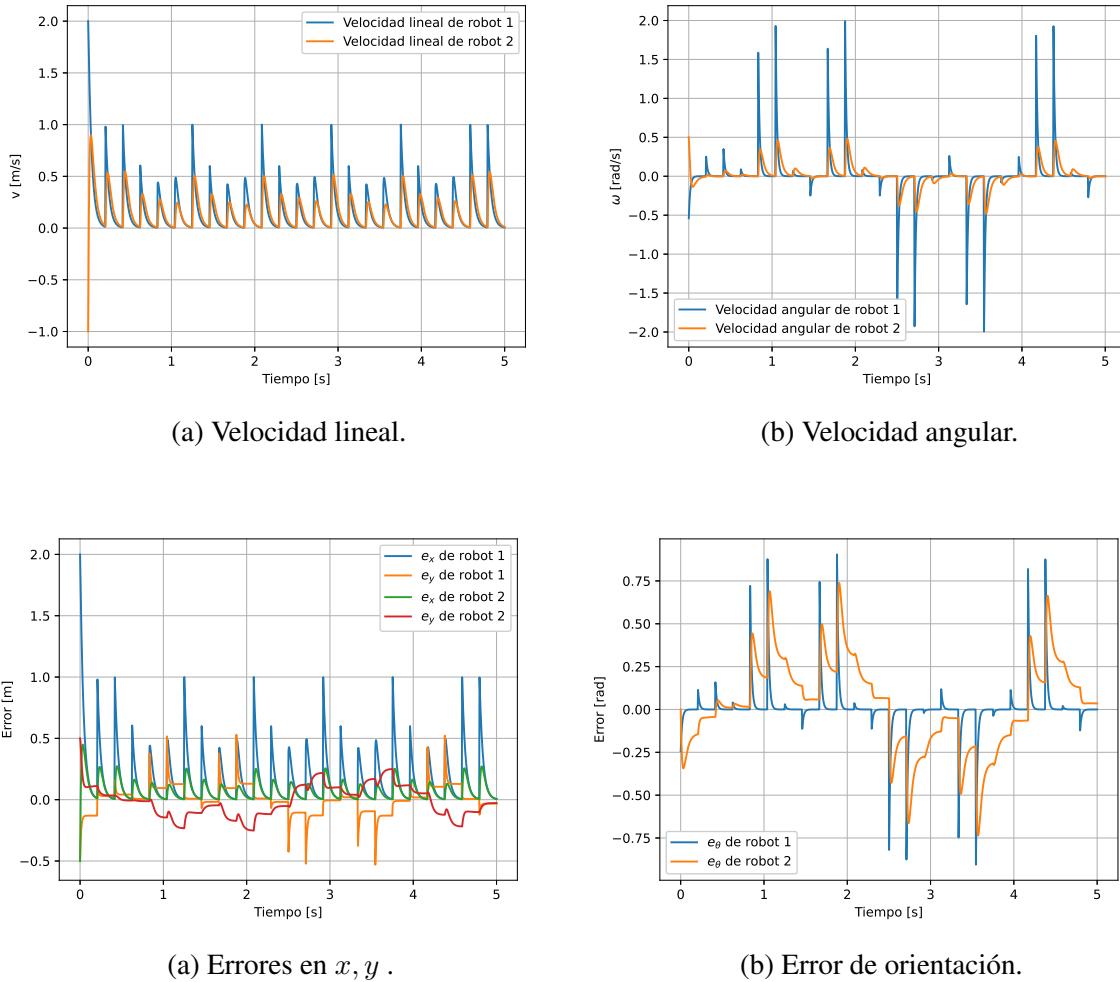


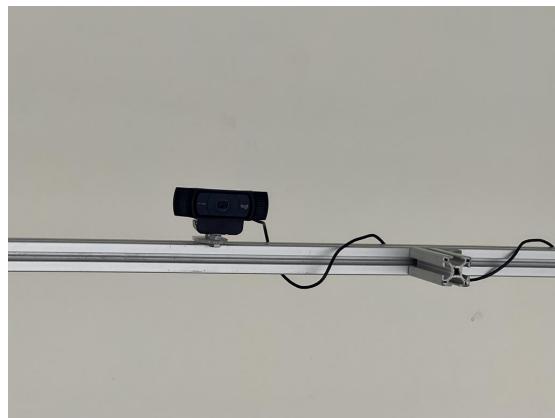
Figura 4.4: Control y errores para múltiples puntos de A^* para dos agentes. La Figura (4.4a) indica la convergencia al origen de los errores e_x conforme se asigna un nuevo punto de la ruta y, la oscilación de los errores e_y con respecto al origen, debido a los cambios de orientación.

4.2. Entorno experimental

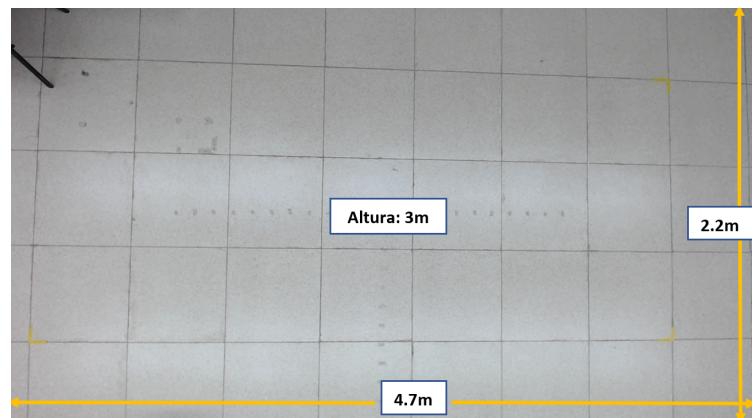
Para el set experimental se propuso un entorno estructurado virtual. Con ayuda de las bibliotecas de OpenCV se dibujaron en el marco de la cámara los obstáculos, la cuadrícula de los nodos del grafo y los puntos que constituyen la ruta encontrada con A^* . Por otro lado, para el sistema robótico se construyeron 4 robots móviles diferenciales, donde cada uno tiene un microcontrolador Arduino en el que se programó un controlador PID para regulación de velocidad de rueda izquierda y derecha.

4.2.1. Entorno de navegación

Para el entorno de navegación se armó una estructura con perfiles de aluminio que se colocó a 3 metros del suelo del laboratorio. En ésta, se atornilló una cámara Logitech C920 como se observa en la Figura (4.5a). El campo de visión de esta cámara colocada a esta distancia es de $10.34 m^2$. Lo cual es un rectángulo de 2.2 m de ancho y 4.7 m de largo, como se observa en la Figura (4.5b).



(a) Cámara C920.



(b) Campo de visión de la cámara.

Figura 4.5: Sistema de visión para entorno experimental.

Para exportar y trazar los entornos de navegación, como el presentado en la Figura (4.1), al marco de referencia de la cámara; se ocuparon las bibliotecas OpenCV-python y OpenCV-contrib-python, como se presenta en la Figura (4.6). La cuadrícula de color azul representa el grafo y cada cuadrado un nodo de 35.3 cm de lado, que puede ser un espacio libre para la búsqueda de caminos. También, se colocaron en color rojo todos los nodos marcados como obstáculos; es decir, nodos por los cuales no es posible que el robot navegue. Por último, se presenta en color verde, cada uno de los puntos que A^* encontró para el camino libre de obstáculos.

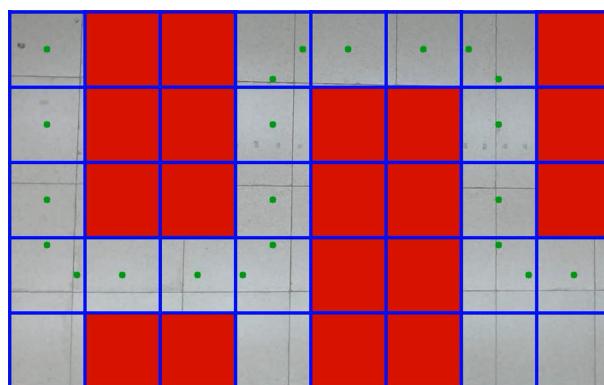
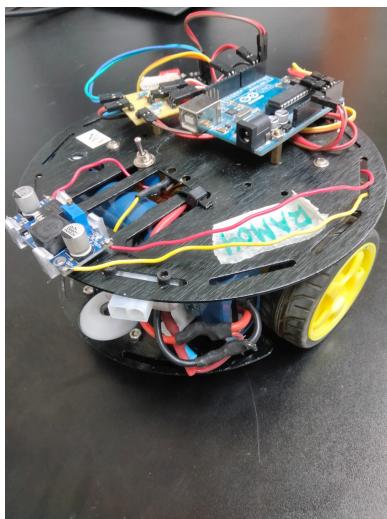


Figura 4.6: Entorno virtual de navegación.

1 4.2.2. Sistema robótico

2 Se construyeron cuatro robots móviles diferenciales. Cada uno consta de dos actuadores
3 para velocidades de rueda izquierda y derecha, además de una rueda loca. Cada robot tiene
4 un microcontrolador Atmega328 en una placa Arduino, junto con un módulo Bluetooth para
5 establecer la comunicación serial. En la Figura (4.7a) se presenta en vista isométrica el primer
6 robot armado. Posteriormente, se construyeron otros tres robots con el mismo hardware y la
7 misma programación. Los cuatro robots para los experimentos se muestran en la Figura (4.7b).



(a) Primer robot.



(b) 4 robots.

Figura 4.7: Robots para parte experimental.

8 Para llevar a cabo la parte experimental existen dos sistemas de control. Uno es el control
9 de alto nivel, presentado en las ecuaciones (4.2) y (4.3). Éste es el encargado de la navegación
10 y coordinación de los robots en la ruta previamente encontrada con A^* . Por otro lado, existe un
11 control de bajo nivel para cada uno de los robots, el cual es un control PID de regulación para
12 velocidades de rueda izquierda y derecha; encargado de establecer las velocidades deseadas en
13 cada instante para que los robots puedan llegar a cada uno de los puntos deseados.

14 Se presenta en la Figura (4.8) un diagrama de funcionamiento del lazo cerrado del control
15 de bajo nivel para cada uno de los robots. El funcionamiento consiste en que el PID de cada
16 una de las ruedas reciba un error de velocidad y lo transforme en una señal por modulación de
17 ancho de pulso (PWM). Estas señales se mandan a los registros asociados con los pines que
18 habilitan las entradas del puente H. Esto lleva a que se activen los motores y al mismo
19 tiempo los contadores de pulsos por vuelta asociados a los encoders. Posteriormente, se realiza
20 una conversión de estos datos para obtener la velocidad actual del robot en rad/s y por ende,
21 el error de velocidad con respecto a la velocidad de referencia o deseada; cerrando con esto el

1 lazo de control.

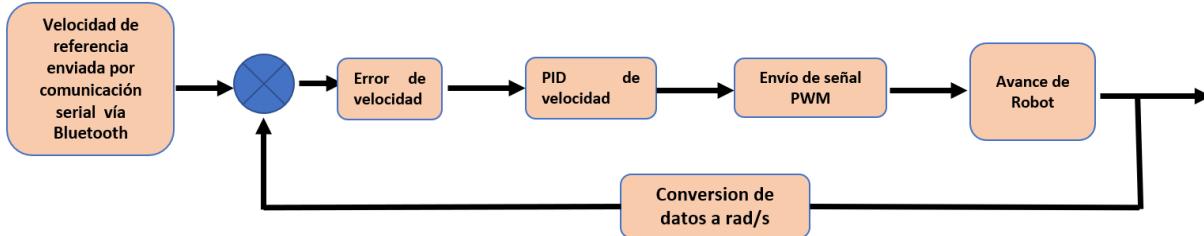


Figura 4.8: Funcionamiento del control de bajo nivel.

2 4.2.3. Retroalimentación con marcadores ArUco

3 Para conocer la posición de cada uno de los robots con el sistema de visión, se utilizó la
4 biblioteca ArUco. La cual es una biblioteca de código abierto basada en OpenCV; que permite
5 detectar marcadores como los mostrados en la Figura (4.9). Cada marcador tiene un patrón
6 binario único que se codifica para obtener la posición y orientación relativa de la cámara con
7 respecto a cada uno de ellos [Bravo, 2018].



Figura 4.9: Ejemplo de marcadores ArUco.

8 Se elaboró un programa que permite obtener la posición de más de un marcador de forma
9 simultánea. Junto con esto, se agregó el marco de referencia de cada uno de ellos, como se
10 observa en la Figura (4.10a). Posteriormente, se diseñó una base en la parte superior de cada
11 robot y se le designó un marcador específico a cada uno (Figura 4.10b).

12 La biblioteca ArUco se utilizó para tres cosas:

- 13 1. Detectar los identificadores de todos los marcadores ArUco que aparecen en la cámara.
- 14 2. Obtener las coordenadas de las cuatro esquinas de todos los marcadores ArUco.
- 15 3. Dibujar el marco de referencia en cada uno de los marcadores ArUco detectados.

16 Aunque la biblioteca ArUco brinda la detección e información de los marcadores. Para
17 obtener la posición del centro de cada uno de ellos. Es necesario hacer una conversión de coor-
18 denadas a partir de los datos de sus esquinas, con las ecuaciones de (4.18).

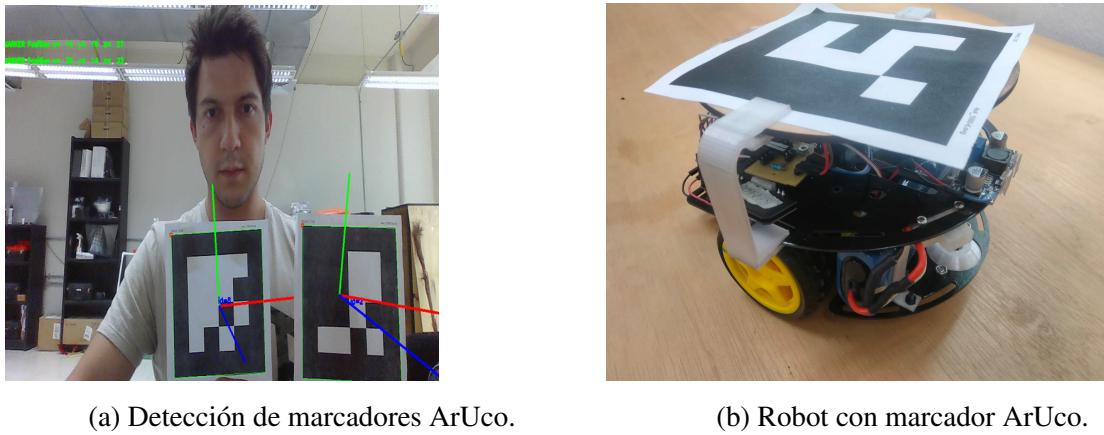


Figura 4.10: Implementación de marcadores ArUco

$$x[\text{cm}] = d[\text{cm}] \left(\frac{x[\text{px}] - Co_x}{f_x} \right); \quad (4.18)$$

$$y[\text{cm}] = d[\text{cm}] \left(\frac{Co_y - y[\text{px}]}{f_y} \right);$$

1 donde:

- 2 ■ $x[\text{cm}], y[\text{cm}]$: son la proyección de la distancia relativa al centro de la imagen en el eje x
3 y eje y .
- 4 ■ $d[\text{cm}]$: es la distancia de profundidad del objeto con respecto a la cámara.
- 5 ■ Co_x, Co_y : son las coordenadas del centro de la imagen de la cámara a lo largo del eje x y
6 eje y .
- 7 ■ f_x, f_y : son las distancias focales en el eje x y en el eje y respectivamente. Este parámetro
8 hace referencia a la relación píxel-centímetro
- 9 ■ $x[\text{px}], y[\text{px}]$: es la coordenada del centro del objeto detectado en la imagen (en píxeles).

10 Esto quiere decir que, a partir de conocer las cuatro esquinas del marcador ArUco, es po-
11 sible encontrar el punto medio entre dos pares de esquinas para establecer una coordenada x_c
12 y, posteriormente, encontrar un punto medio entre otro par de esquinas que genere un vector
13 perpendicular al primer par de puntos; encontrando un punto y_c . De modo que, la coordenada
14 (x_c, y_c) indica el centro del ArUco [Andriyanov et al., 2022]; ver Figura (4.11).

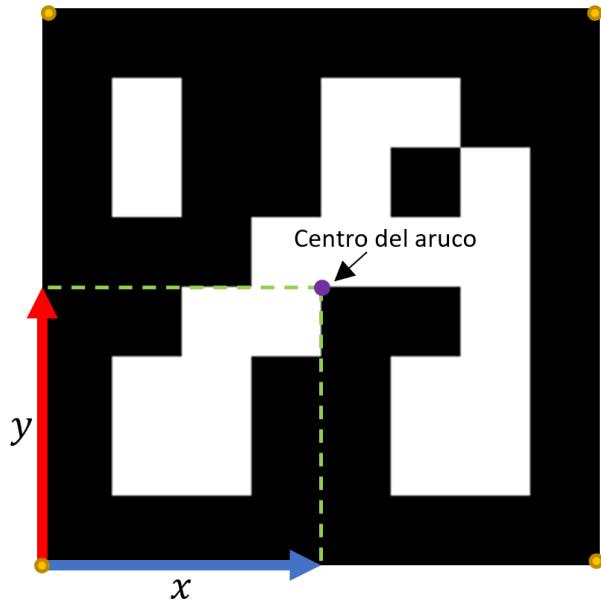


Figura 4.11: Marcador ArUco con vectores para ubicar su centro.

1 La construcción del robot, la detección del marcador ArUco y la exportación del entorno
 2 virtual de navegación a la cámara de la Figura (4.5a). Se presentan en la Figura (4.12), donde el
 3 origen del marco de referencia inercial se colocó en la esquina inferior izquierda.

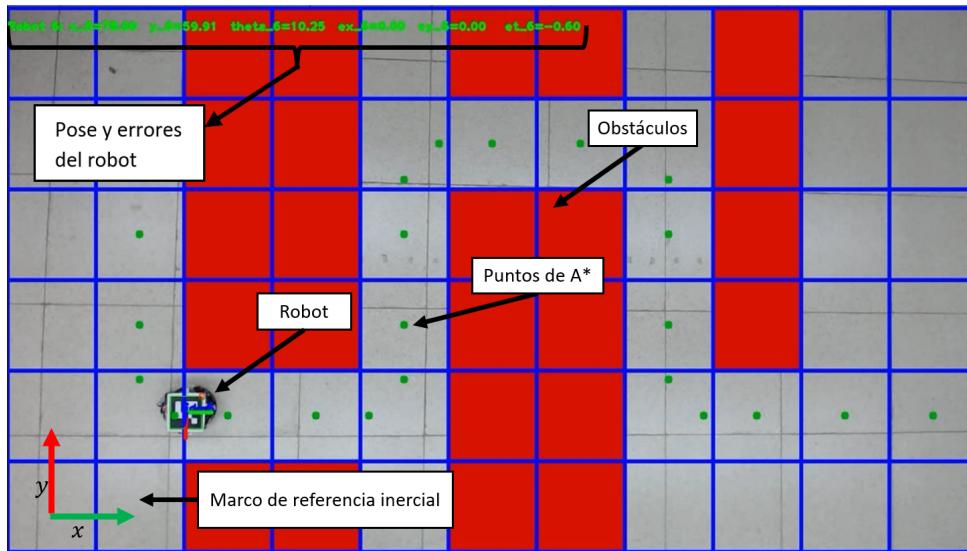


Figura 4.12: Entorno de navegación con robot móvil.

4 Para obtener la pose del robot en cada instante se implementaron las ecuaciones de (4.18);
 5 debido a que el robot tiene una altura de 12.8 cm y la cámara se colocó a una altura de 3 m del
 6 suelo, se requiere restar la altura del robot a la altura total para obtener la profundidad de su

1 detección. Entonces: $d = 300 \text{ cm} - 12.8 \text{ cm} = 287.2 \text{ cm}$. Por otro lado, las dimensiones de la
 2 cámara es de $1280 \text{ px} \times 720 \text{ px}$. Esto quiere decir que para desplazar el origen hacia la esquina
 3 inferior izquierda de la cámara se propuso $Co_x = 0$ y $Co_y = 720 \text{ px}$ y, con los respectivos
 4 parámetros de calibración de distancia focal f_x y f_y . Las ecuaciones para obtener la pose del
 5 robot de la Figura (4.12) son:

$$x = 287.2 \text{ cm} \left(\frac{0 \text{ px} - x_c}{915} \right); \quad (4.19)$$

$$y = 287.2 \text{ cm} \left(\frac{720 \text{ px} - y_c}{930} \right);$$

6 donde la orientación se obtiene por medio de (x_c, y_c) de la forma:

$$\theta = \arctan \left(\frac{y_c}{x_c} \right). \quad (4.20)$$

7 Por último, se tiene el lazo cerrado de todo el sistema de control para la navegación coordi-
 8 nada de la ruta encontrada con A^* , como se observa en la Figura (4.13). Para esto, se establece
 9 la pose deseada del robot hacia cada uno de los puntos de A^* . Se realiza un mapeo de errores
 10 hacia el marco de referencia local de cada robot. Con base en estos errores, se obtienen las
 11 entradas de control (velocidad lineal y angular) de cada robot. Posteriormente, se realiza una
 12 conversión a velocidades de rueda izquierda y derecha y se ingresan estas señales a una función
 13 de saturación para guardar la integridad de los actuadores. Para los robots construidos se pre-
 14 senta en (4.21) la función de saturación propuesta, donde $a = 6 \text{ rad/s}$ es el valor máximo de
 15 velocidad permitida, V_{lr} son las velocidades de rueda izquierda y derecha del robot y $V_{lr_{sat}}$ son
 16 las señales de velocidad V_{lr} con saturación.

$$V_{lr_{sat}} = \begin{cases} a & \text{si } V_{lr} > a \\ V_{lr} & \text{si } a \geq V_{lr} \geq -a \\ -a & \text{si } -a > V_{lr}. \end{cases} \quad (4.21)$$

17 Las señales $V_{lr_{sat}}$ se mandan por comunicación serial a los robots. Donde cada uno tiene su
 18 propio control de bajo nivel (Figura 4.8). Este control se encarga de regular la velocidad del
 19 robot, hasta llegar a la posición deseada, en donde en cada instante de tiempo (1 ms) se obtiene
 20 la posición del robot con el marcador ArUco.

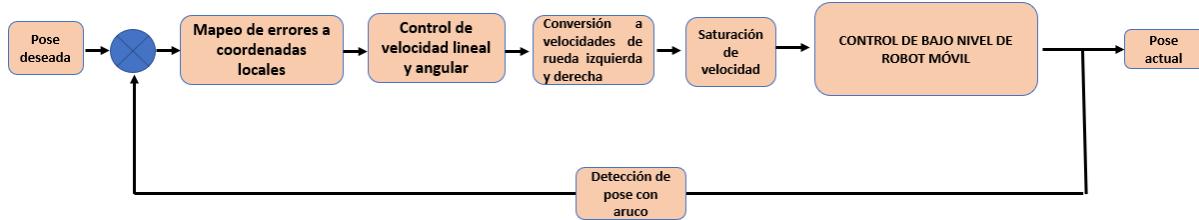
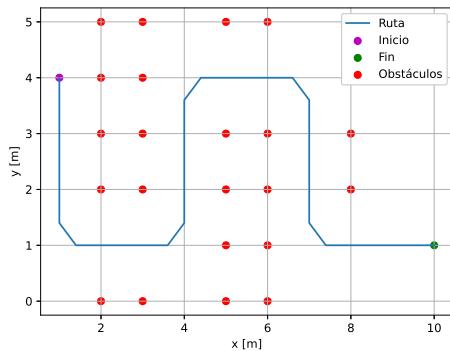
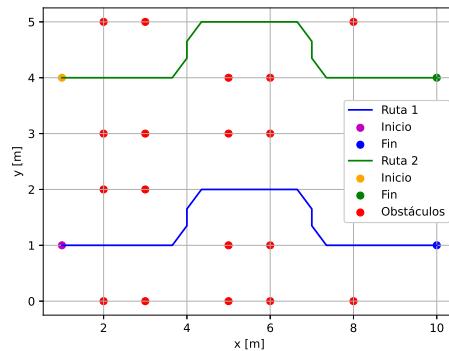


Figura 4.13: Funcionamiento de control de alto nivel.

4.3. Experimentos

La parte experimental se trabajó con dos entornos de navegación. El primero se presenta en la Figura (4.14a); donde la cantidad de obstáculos permitió a A^* encontrar sólo un camino. Con base en este entorno, se realizaron experimentos para coordinar 2 y 3 robots en configuración líder-seguidor. Por otro lado, en el entorno de la Figura (4.14b) se redujo el número de obstáculos, permitiendo a A^* encontrar dos caminos libres de obstáculos. Por lo cual, para este entorno, se coordinaron 4 robots, asignando a cada par una configuración de líder-seguidor en una de las dos rutas.

(a) Ruta de un camino de A^* .(b) Múltiples caminos con A^* .Figura 4.14: Rutas obtenidas con A^* para experimentos.

Se estudiaron dos topologías de comunicación entre cada par de robots conectados. En la Figura (4.15a) se presenta una comunicación unidireccional (CU), donde sólo el robot 2 (que navega con un desfase sobre la ruta de A^* para mantener la formación) recibe información del robot 1. Por otro lado, la Figura (4.15b) muestra una comunicación bidireccional (CB). Esto quiere decir que el desempeño en la navegación del robot 1, influyen los datos de la pose y errores del robot 2.

A partir de los 4 robots construidos, los dos entornos que se programaron y las dos topolo-

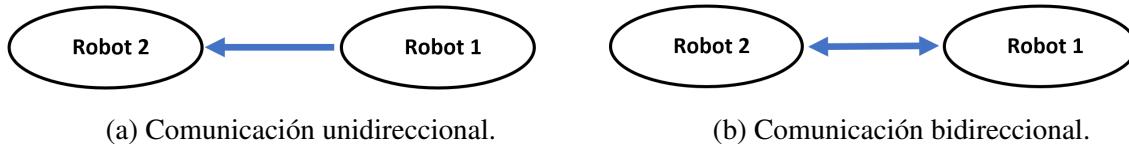


Figura 4.15: Topologías de comunicación entre cada par de robots.

Entornos	No. de robots	CU	CB
Entorno 1	2	Experimento 1	Experimento 2
Entorno 1	3	Experimento 3	Experimento 4
Entorno 2	4	Experimento 5	Experimento 6

Cuadro 4.1: Experimentos que se llevaron a cabo.

- 1 gías de comunicación. Se realizaron un total de 6 experimentos, como se presenta en el cuadro
 2 4.1. Los videos de estos experimentos pueden ser consultados en:
 3 <https://drive.google.com/drive/folders/1wMgJq8WwvHArfsmLuuK3dGsd0hbUVY6P?usp=sharing>
- 4 Los seis experimentos se realizaron asignando las leyes de control (4.2) y (4.3) a cada uno
 5 de los robots. Sin embargo, a nivel experimental fue necesario definir una cota cartesiana deno-
 6 minada $cota_{xy}$ y una de orientación denominada $cota_\theta$. Ambas cotas se asignaron a cada uno de
 7 los puntos encontrados con A^* . La cota $cota_{xy}$ indica un radio de convergencia; esto quiere decir que
 8 los errores de los robots se van a mandar a cero una vez que hayan entrado al espacio marcado
 9 por esta cota. Por otro lado, $cota_\theta$ indica un radio alrededor de cada punto en el cual una vez
 10 que ingrese el robot, este permanecerá con la orientación que se haya calculado en el último
 11 instante. El objetivo de estas dos cotas es que sirvan como indicador para que los robots sepan
 12 cuándo cambiar al siguiente punto de la ruta y continúen con la navegación.
- 13 Para los experimentos se propuso que ambas cotas fueran iguales; para asegurar que el robot
 14 se detenga en el punto deseado con la orientación correcta. Aunado a esto, la necesidad de las
 15 cotas surge del hecho de que en los robots ocupados, incluso en valores de errores e_x y e_y
 16 muy cercanos a cero, el control continua enviando velocidades suficientes para mover el robot,
 17 causando que sea complicado asignarle el siguiente punto de la ruta. Sin embargo, implementar
 18 las cotas a cada uno de los puntos erradicó este problema y facilitó programar la navegación.
 19 La Figura (4.16) presenta un diagrama de los puntos de la ruta con las cotas.
- 20 En todos los experimentos se dejó la misma tasa de actualización de datos. De acuerdo
 21 con el diagrama de la Figura (4.13). Los datos de envío por comunicación serial del control de
 22 alto nivel a cada uno de los robots es de 1 ms y, la actualización de la señal de control de los
 23 controladores PID de bajo nivel de cada robot es de 50 ms. La actualización del control de bajo

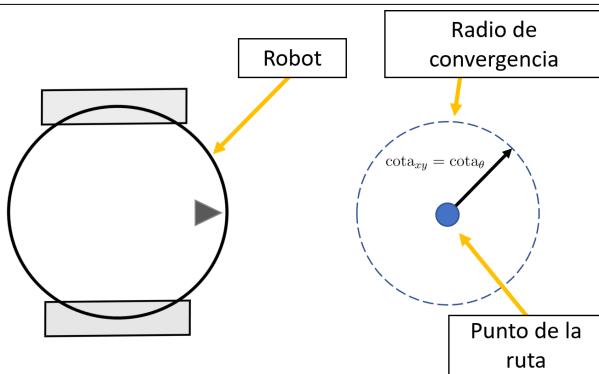


Figura 4.16: Diagrama de convergencia del robot a uno de los puntos de la ruta de A^* .

1 nivel es más lento debido a que se requiere dejar más tiempo para que este control regule a la
 2 velocidad indicada (las velocidades de rueda izquierda y derecha de los robots en cada uno de
 3 los experimentos pueden ser consultadas en el Apéndice B).

4.3.1. Dos robots

5 Para dos robots se elaboró el entorno experimental de la Figura (4.17). Los robots se acomo-
 6 daron en el plano con las mismas condiciones iniciales y una orientación de $-\pi/2$ con respecto
 7 al marco de referencia inercial. Sin embargo, en estos experimentos, no sólo se varió el tipo de
 8 comunicación (CU o CB) sino también las ganancias de velocidad lineal, velocidad angular y
 9 de acoplamiento.

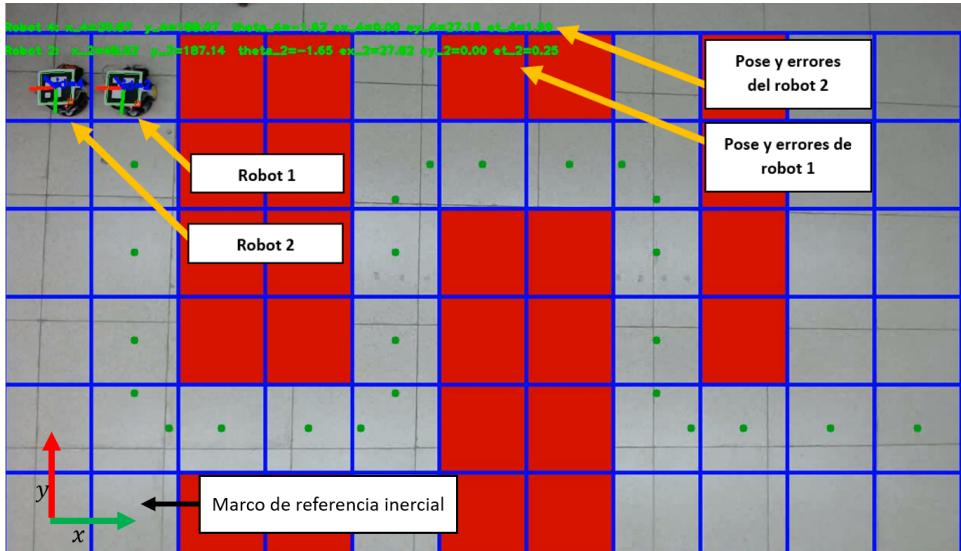


Figura 4.17: Entorno para experimentos 1 y 2.

1 4.3.1.1. Experimento 1

2 Este experimento se llevó a cabo con una comunicación unidireccional (CU) como se muestra en la Figura (4.18a). Los robots tardaron 45.63 segundos en completar el recorrido. Las
 3 ganancias con las que se presentó el mejor desempeño fueron:
 4

$$k_x = 1, k_y = 0.0001, k_\theta = 1; \quad (4.22)$$

5 las cuales permitieron definir la heurística de (4.23) para las ganancias de acoplamiento:

$$C_x = 1.2k_x, C_y = 0.1ky. \quad (4.23)$$

6 Se propusieron ambas cotas ($cota_{xy}$ y $cota_\theta$) de 10 cm. Por otro lado, se le asignó un desfase
 7 al robot 2 de $\Delta_{21} = -25$ cm con respecto a la ruta.

8 La Figura (4.18b) presenta la ruta recorrida por ambos robots. Se observa en este experimento,
 9 que el robot 1 tuvo dificultades en los cambios de dirección, ya que no recibe información
 10 del robot 2. Esto conlleva a que este robot presente dificultades para regular su velocidad.

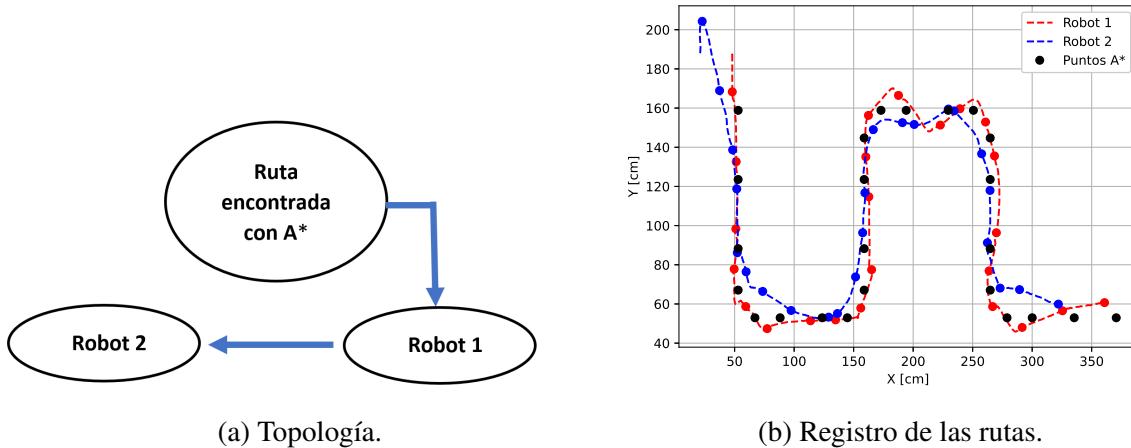
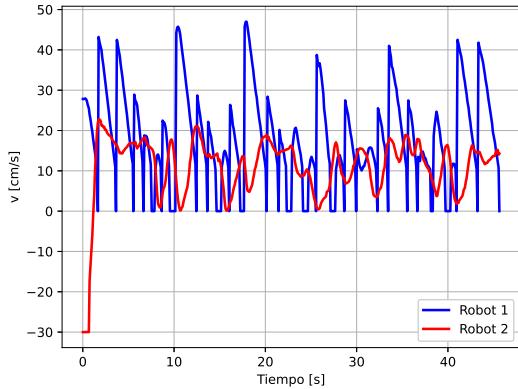


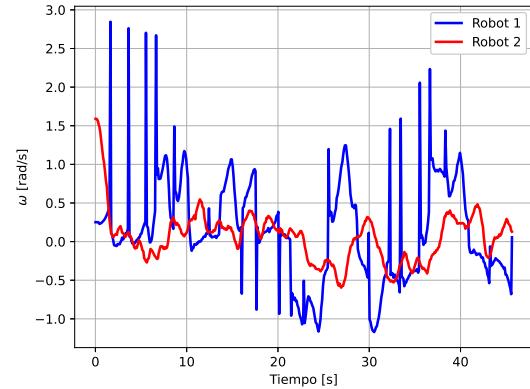
Figura 4.18: Topología y rutas recorridas del experimento 1.

11 La Figura (4.19a) presenta las velocidades lineales de ambos robots, donde se observa que
 12 el robot 1 demanda más velocidad para llegar a los puntos de A^* , en comparación con el robot
 13 2. Este último debe de llegar a sus puntos de regulación dados por la pose del robot 1; pero
 14 manteniendo su desfase con respecto a la ruta de A^* . Otro aspecto importante es el hecho de
 15 que, al inicio de su recorrido, el robot 2 tiene velocidad lineal negativa, ya que se alinea en
 16 la parte trasera del robot 1 para comenzar a seguirlo en la navegación. En la Figura (4.19b) la

- 1 señal de control angular oscila alrededor de cero; pero el robot 1 mantiene de igual forma mayor
- 2 demanda de velocidad angular para los cambios de orientación, ya que es el que está recibiendo
- 3 directamente los datos para seguir el camino planeado.



(a) Velocidad lineal.



(b) Velocidad angular.

Figura 4.19: Entradas de control del experimento 1.

- 4 La Figura (4.20a) muestra los errores en e_x . En ambos robots este error trata de converger
- 5 a cero cada vez que se ubican en un nuevo punto de la ruta. Por otra parte, se observa un
- 6 comportamiento similar en los errores de orientación de la Figura (4.20c) y la Figura (4.19b),
- 7 ya que el error de orientación está directamente asociado a esta entrada de control angular. Sin
- 8 embargo, los errores en e_y de la Figura (4.20b) oscilan alrededor de cero y es un error que al no
- 9 estar acotado, incrementa de forma abrupta en los cambios de orientación.

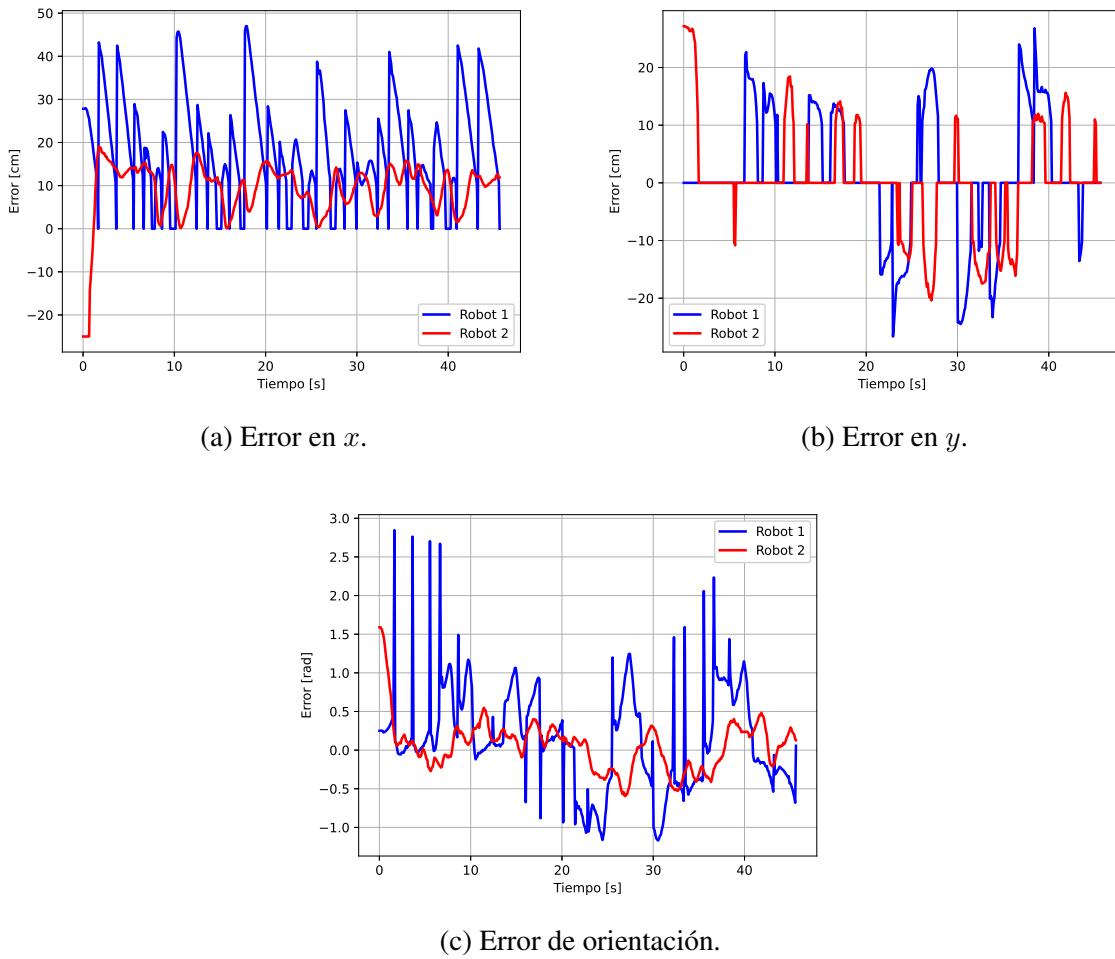


Figura 4.20: Errores de navegación del experimento 1.

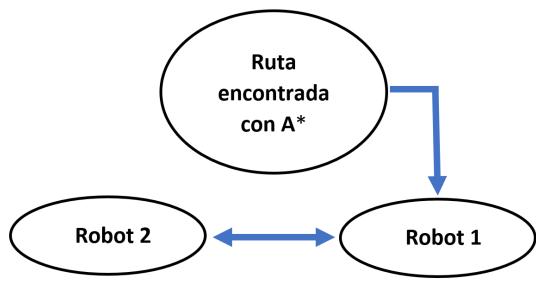
4.3.1.2. Experimento 2

Este experimento se llevó a cabo con una comunicación bidireccional (CB). Los robots se acomodaron como se muestra en la Figura (4.18a), y tardaron 54.05 segundos en completar el recorrido. El mejor desempeño se presentó con las ganancias:

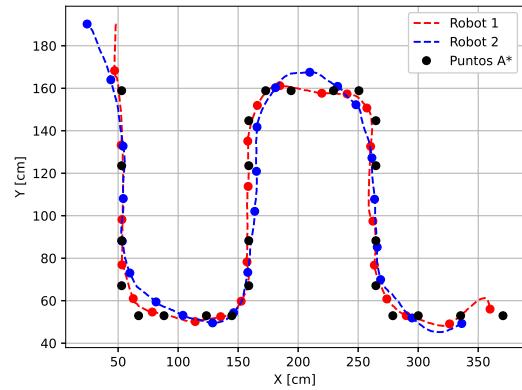
$$k_x = 0.3, k_y = 0.0001, k_{\theta_1} = 1, k_{\theta_2} = 0.5. \quad (4.24)$$

Las ganancias de acoplamiento y las cotas se dejaron igual que en el experimento 1. Sin embargo, por el tipo de topología, se presentó un mejor resultado proponiendo un desfase positivo de 20 cm al robot 1 con respecto a la ruta. Es decir, $\Delta_{12} = 20$ cm en vez de un desfase negativo de -25 cm al robot 2 ($\Delta_{21} = -25$ cm) como se realizó en el experimento 1. Es posible reducir esta distancia, ya que existe mayor comunicación entre ellos y por ende resulta más sencillo para el control evitar las colisiones.

La Figura (4.21a) presenta la comunicación bidireccional entre el robot 1 y el robot 2. Por otro lado, la Figura (4.21b) presenta la ruta recorrida por ambos robots. La cual, a pesar de ser la misma que en el experimento 1, el resultado es más preciso en comparación al registrado en la Figura (4.18b); gracias a la comunicación bidireccional, la cual logra que el robot 1 disminuya su velocidad en caso de que comience a alejarse del robot 2. Esto trae como resultado que ambos robots tengan un mejor desempeño al seguir el camino encontrado, sobre todo en los cambios de dirección.



(a) Topología.



(b) Registro de las rutas.

Figura 4.21: Topología y rutas recorridas del experimento 2.

En la Figura (4.22a) se observa como el robot 1 requiere más velocidad tanto lineal como angular para seguir la ruta, que el robot 2 en tratar de llegar a la pose del robot 1. Por otro lado, a diferencia de la Figura (4.19a) mostrada en el experimento 1. En el experimento 2, el robot 1 se detiene en ciertos instantes de tiempo, para mantener la distancia previamente definida con el robot 2. De igual forma, en la Figura (4.22b) se observa mayor demanda de velocidad en el robot 1 que en el robot 2. En las ganancias propuestas en (4.24), el robot 2 tiene una ganancia angular menor, tal que $k_{\theta 1} > k_{\theta 2}$, lo cual permite al robot 2 dar curvas más abiertas en las vuelta durante la navegación. Y por ende, esto le demande menos velocidad angular.

La Figura (4.23a) indica el error e_x de ambos robots. Algo importante a destacar es cómo este error se aproxima a cero conforme los robots se posicionan en cada uno de los puntos de la ruta, ya que a pesar de que el desfase ahora se le asignó al robot 1; el comportamiento de este error indica que los robots mantuvieron su distancia durante la navegación. La Figura (4.23b) presenta el error no acotado e_y , el cual incrementa de forma abrupta su valor ante un cambio de dirección; sin embargo, este error siempre oscila con respecto a cero. Lo mismo ocurre en la Figura (4.23c), donde se mantiene un comportamiento muy similar a la entrada de velocidad angular de la Figura (4.20c) y, de igual forma, el robot 1 requiere más velocidad lineal y angular para seguir el camino encontrado con A^* , que el robot 2.

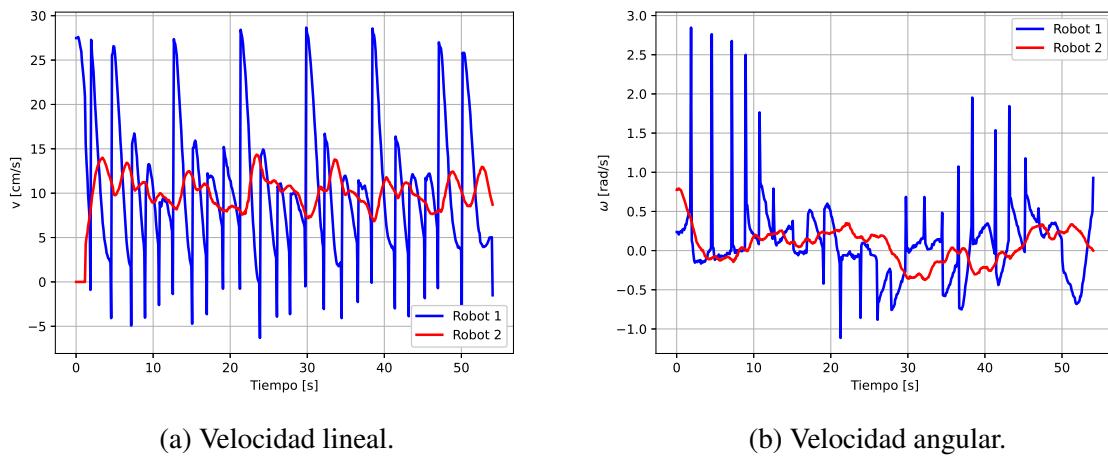


Figura 4.22: Entradas de control del experimento 2.

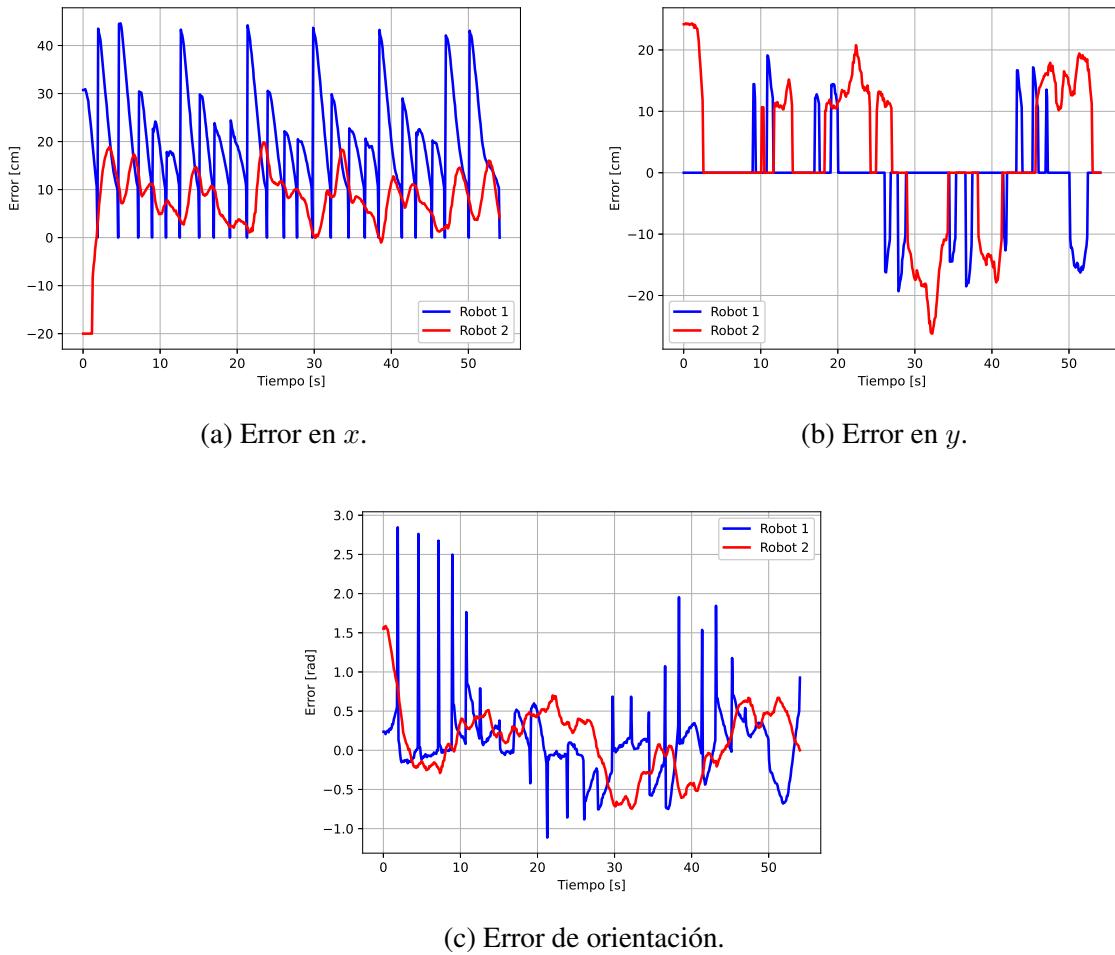


Figura 4.23: Errores de navegación del experimento 2.

4.3.2. Tres robots

Los experimentos con tres robots se llevaron a cabo a partir de ubicarlos en el plano, con una orientación para los tres de $-\pi/2$ con respecto al marco de referencia inercial, como se muestra en la Figura (4.24). Además de variar la comunicación (CU o CB), se variaron las ganancias del controlador, para obtener un mejor desempeño de acuerdo a la topología propuesta.

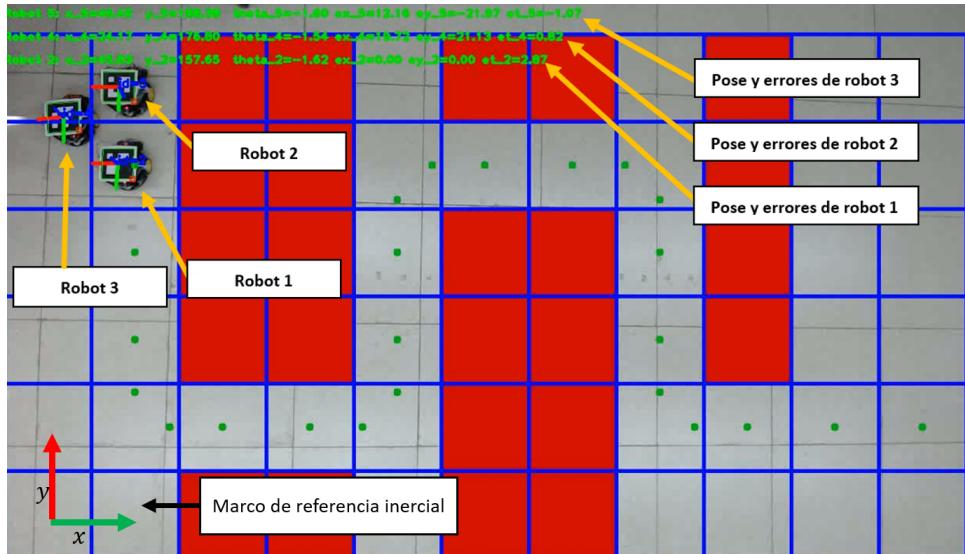


Figura 4.24: Entorno para experimentos 3 y 4.

4.3.2.1. Experimento 3

Al igual que el experimento 1, en el experimento 3, se llevó a cabo una comunicación unidireccional (CU) como se muestra en la Figura (4.25a). Aquí el robot 3 no tiene comunicación con el robot 1; pero si con el robot 2, el cual sí tiene comunicación con el robot 1. Los robots tardaron 44.26 segundos en completar el recorrido. Las ganancias del control con las que se presentó el mejor desempeño fueron las mismas que en (4.22). Se dejó la heurística de (4.23).

Se propusieron las cotas de 10 cm. Se mantuvo el mismo desfase entre los tres robots con respecto a la ruta recorrida. Esto quiere decir que la distancia entre el robot 3 y robot 2 es la misma que la del robot 2 y el robot 1. Por lo cual, los robots 2 y 3, tienen un desfase $\Delta_{32} = \Delta_{21} = -25$ cm. Esta es la distancia entre centro y centro de los marcadores ArUco.

La Figura (4.25b) presenta la ruta recorrida por los tres robots. En este experimento el robot 1 tuvo dificultades para seguir el camino, ya que no pudo pasar de forma precisa sobre todos los puntos de la ruta; puesto que la CU causa que no tenga conocimiento del robot 2, entonces para este caso el acoplamiento no le ayudó a regular su velocidad. Sin embargo, el robot 2 y 3 presentaron un mejor comportamiento en el seguimiento del camino en comparación al robot 1.

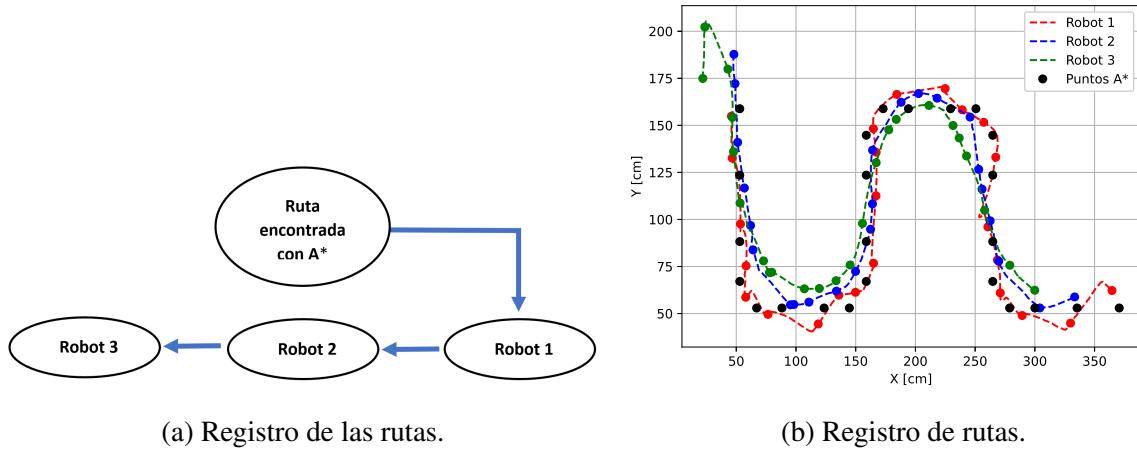


Figura 4.25: Topología y rutas recorridas del experimento 3.

1 La Figura (4.26a) presenta la velocidad lineal de los tres robots. Se observa que el robot
 2 1 es el que demanda mayor velocidad lineal en seguir la ruta; a diferencia de los robots 2 y
 3. Sin embargo, al inicio de la gráfica, se observa un pico de velocidad en el robot 3. Como
 4 se observa en la Figura (4.24), el robot 3 es el último de la formación. Esto quiere decir que
 5 para alinearse con respecto a la parte trasera del robot 2 requiere desacelerar. Por otro lado, en la
 6 Figura (4.26b) la velocidad angular, que demanda el robot 1 para seguir la ruta, es mucho mayor
 7 a la que requieren los robots 2 y 3. Estos dos robots demandan una velocidad angular similar;
 8 sin embargo, al inicio el robot 3 demanda más velocidad angular, ya que busca alinearse en la
 9 formación.

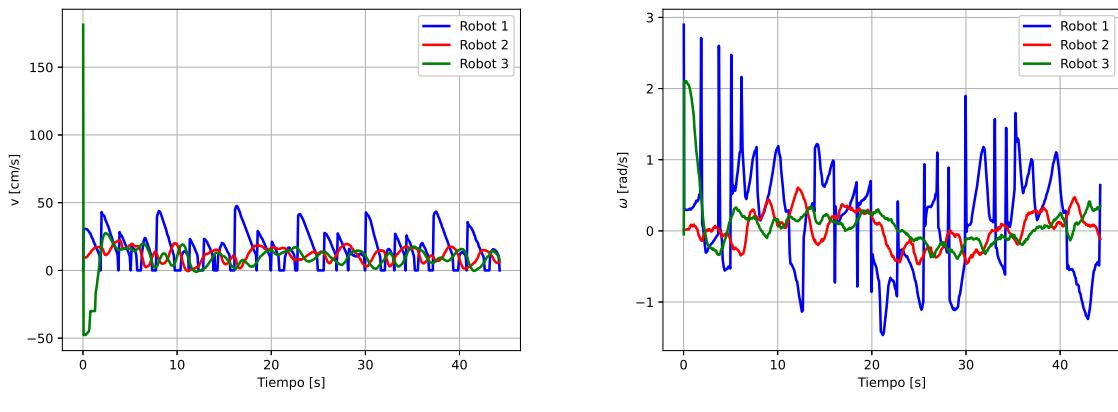
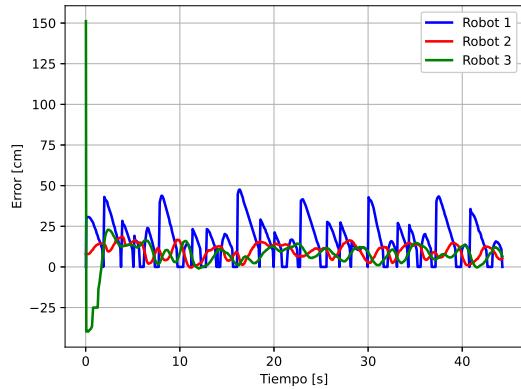
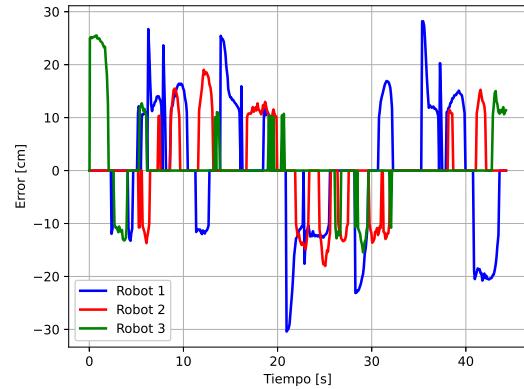


Figura 4.26: Entradas de control del experimento 3.

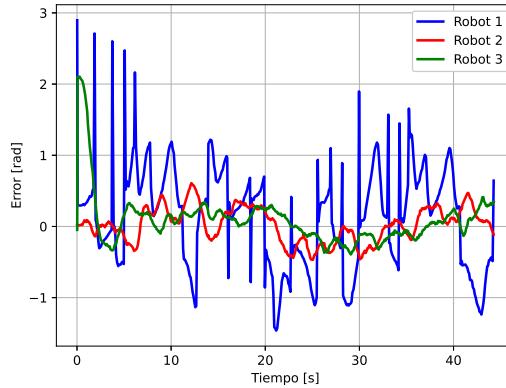
La Figura (4.27a) expone los errores e_x de los tres robots. Se observa que todos los robots aproximan este error a cero conforme van alcanzando los puntos de la ruta, tomando en cuenta los desfases que se les asignaron. Esto quiere decir, en otras palabras, que lograron navegar manteniendo su formación. Aunado a esto, se presenta en la Figura (4.27b) el error e_y , y cómo éste tiene una oscilación bajo valores similares para los tres robots, a excepción de picos de error en determinados pasos de tiempo para el robot 1. Por último, en la Figura (4.27c) se muestra el error de orientación, el cual debido a la entrada de control angular; este error guarda un comportamiento similar a esta entrada de control expuesta en la Figura (4.26b).



(a) Error en x .



(b) Error en y .



(c) Error de orientación.

Figura 4.27: Errores de navegación del experimento 3.

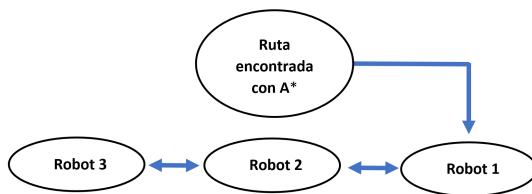
1 4.3.2.2. Experimento 4

2 El experimento 4 se llevó a cabo con una comunicación bidireccional, como se presenta
 3 en la Figura (4.28a). El robot 3 no tiene comunicación con el robot 1 y el robot 2 mantiene
 4 comunicación tanto con el robot 3 y el robot 1. Los robots tardaron 54.56 segundos en completar
 5 la ruta. Para el controlador se propusieron las ganancias mostradas en (4.24), donde se dejó un
 6 valor de 0.5 en la ganancia de velocidad angular de los robots 2 y 3. Sin embargo, debido a la
 7 cantidad de interconexiones de comunicación en este experimento, fue necesario modificar el
 8 valor de las ganancias de acoplamiento, proponiendo las siguientes:

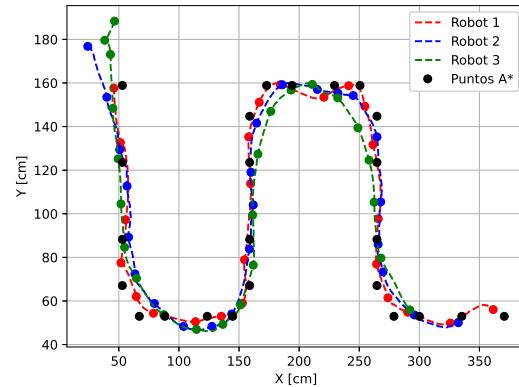
$$C_x = 0.8k_x, C_y = 0.1ky; \quad (4.25)$$

9 de la misma forma, ambas cotas se propusieron de 10 cm. Junto con esto, se dejó un desfase
 10 positivo en el seguimiento de la ruta entre el robot 1 y el robot 2 de $\Delta_{12} = 20$ cm; al igual que
 11 un desfase positivo de $\Delta_{23} = 20$ cm en el seguimiento de la ruta entre el robot 2 y el robot 3.

12 La Figura (4.28b) presenta el camino recorrido por los tres robots. Se observa una navega-
 13 ción más precisa con respecto a las rutas de la Figura (4.25b) en el experimento 3. Debido a que
 14 la CB, entre cada par de robots adyacentes, permite que el robot 2 desacelere al robot 1, en caso
 15 de que este comience a separarse de la formación y, al mismo tiempo, el robot 2 logre que el
 16 robot 3 incremente su velocidad, en caso de que comience a alejarse de la formación.



(a) Topología.



(b) Registro de rutas.

Figura 4.28: Topología y rutas recorridas del experimento 4.

17 Algo importante a destacar es que debido a la CB fue necesario invertir las condiciones
 18 iniciales de los robots 2 y 3. Es decir, en la Figura (4.24) donde se encuentra el robot 2 se colocó
 19 el robot 3 y donde se ubica el robot 3 se colocó el robot 2 como se muestra en la Figura (4.29).
 20 Fue necesario este cambio ya que el hecho de que el robot 2 tenga comunicación con ambos

- ¹ robots, estos colisionan si se dejan como en la Figura (4.24) al iniciar la navegación. Entonces
² para este experimento fue necesario dejar al robot 3 en la parte trasera de la formación.

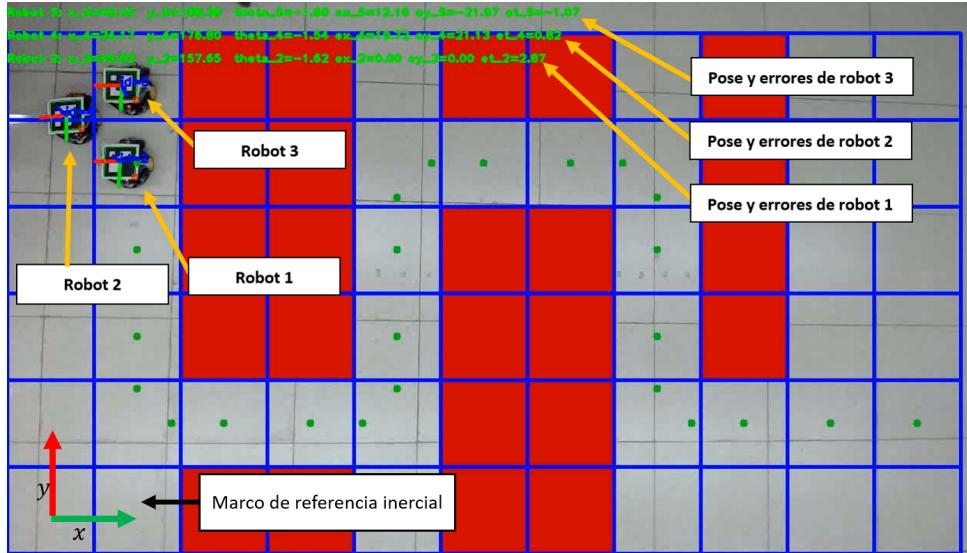
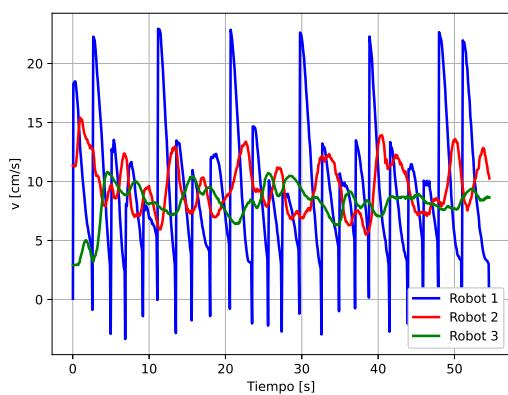
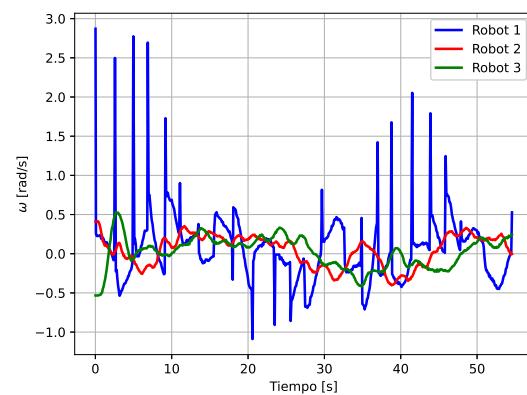


Figura 4.29: Reacomodo de robots para experimento 4.

- ³ La Figura (4.30a) presenta las velocidades lineales de los tres robots. La velocidad lineal
⁴ requerida para el robot que recibe directamente los datos de la ruta de A^* (robot 1) es más
⁵ elevada; en comparación a la requerida para los otros dos robots, los cuales mantienen una
⁶ demanda similar de velocidad lineal a lo largo de la ruta. Lo mismo ocurre con la velocidad
⁷ angular presentada en la Figura (4.30b).



(a) Velocidad lineal.



(b) Velocidad angular.

Figura 4.30: Entradas de control del experimento 4.

- 1 La Figura (4.31a) presenta los errores e_x de los robots. Se observa en estos errores una
 2 oscilación alrededor de cero, lo cual indica que lograron mantener el desfase entre ellos que
 3 permitió conservar la formación durante la navegación. Aunado a esto, se observa que el error
 4 del robot 3 es más elevado que el error del robot 2, ya que la CB causa pequeños sobre tiros
 5 entre estos dos cuando se encuentran muy cerca (menos de 2 cm).
- 6 La Figura (4.31b) presenta los errores e_y , los cuales se mantienen en cero pero se incremen-
 7 tan de forma abrupta en los cambios de orientación. Por otro lado, la Figura (4.31c) muestra
 8 los errores de orientación de los tres robots; los cuales, al igual que en los otros experimentos,
 9 mantienen un comportamiento similar a la velocidad angular de cada uno de los robots, a causa
 10 de la parte proporcional establecida en el diseño del control.

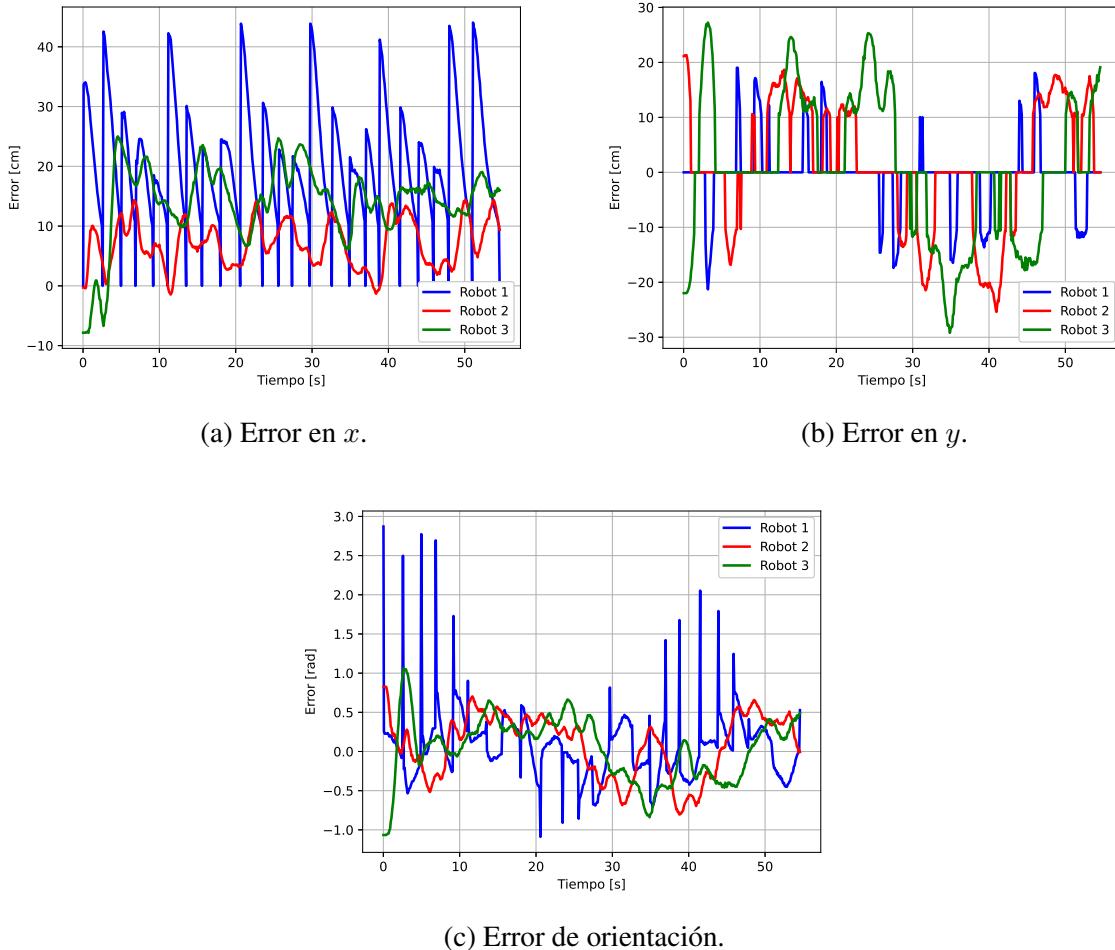


Figura 4.31: Errores de navegación del experimento 4.

4.3.3. Cuatro robots

Los experimentos con cuatro robots se realizaron a partir de formarlos como se presenta en la Figura (4.32). A diferencia de los experimentos anteriores. Para estos experimentos (5 y 6), fue necesario implementar A^* para la búsqueda de más de una ruta; con el objetivo de asignar un par de robots en cada ruta. Sin embargo, al igual que con 2 y 3 robots; se varió la comunicación (CU o CB) además de las ganancias del controlador. Aunado a esto, al tener dos rutas, se abrevió la ruta 1 como R1 y la ruta 2 como R2.

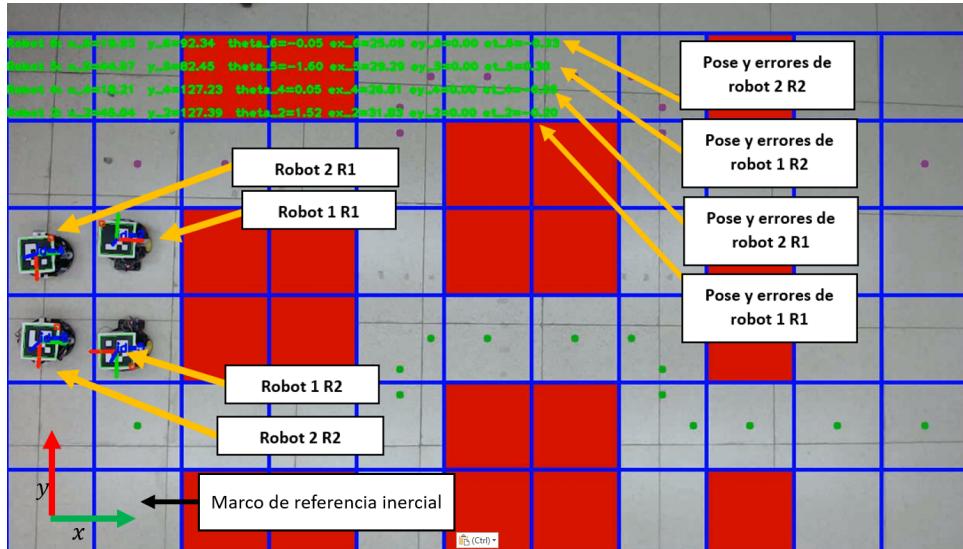
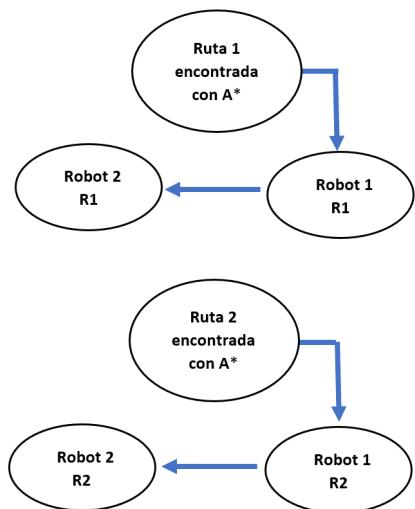


Figura 4.32: Entorno para experimentos 5 y 6.

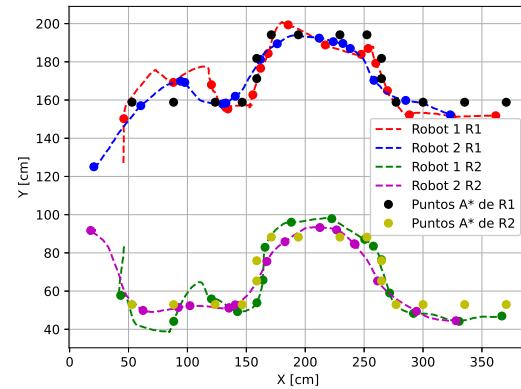
4.3.3.1. Experimento 5

El experimento 5 se llevó a cabo con una comunicación unidireccional como se observa en la Figura (4.33a). El robot 1 R1 sólo mantiene comunicación con el robot 2 R1. Por otro lado, el robot 1 R2 sólo mantiene comunicación con el robot 2 R2. El experimento tardó 32.85 segundos, para que cada robot recorriera su respectiva ruta. Para cada par de robots de una ruta se propusieron las mismas ganancias de (4.22). De igual forma, se dejaron las ganancias de acoplamiento de (4.23). También se dejaron los mismos valores en las cotas. Al robot 2 R1 y robot 2 R2 se les agregó un desfase de $\Delta = -25$ cm con respecto a los puntos de sus rutas.

La Figura (4.33a) presenta la ruta recorrida por los cuatro robots. Los caminos encontrados son de la misma longitud, debido a la simetría del entorno. Por esta razón, los robots tardaron el mismo tiempo en llegar a su meta. Aunque ambas formaciones tienen un comportamiento similar en su navegación, los robots 1 R1 y 1 R2 no tienen conocimiento del otro robot de su ruta, debido a la CU. Esta falta de comunicación bidireccional conlleva a que estos dos robots tengan dificultades en regular su velocidad en los cambios de dirección.

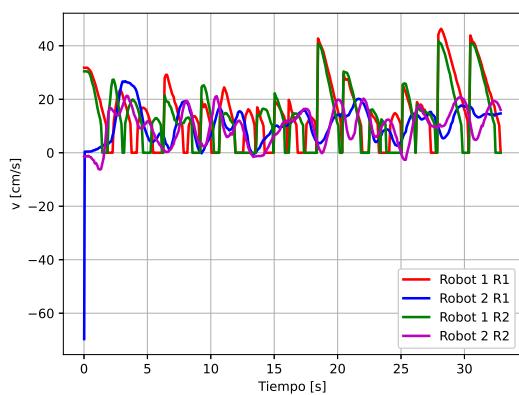


(a) Topología.

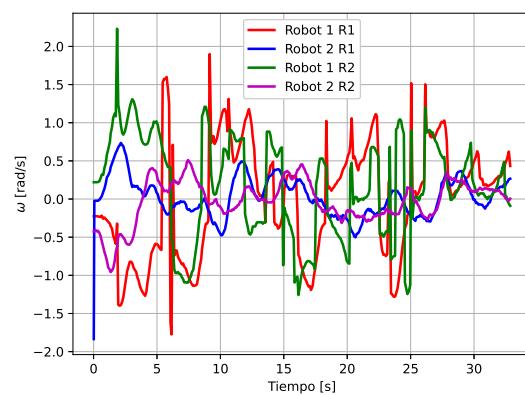


(b) Registro de rutas.

Figura 4.33: Topología y rutas recorridas del experimento 5.



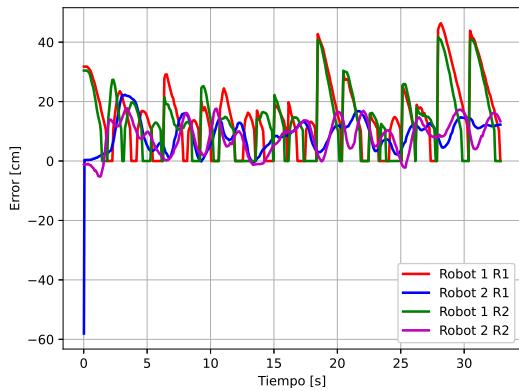
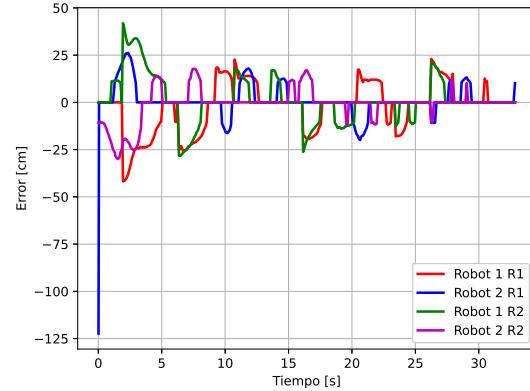
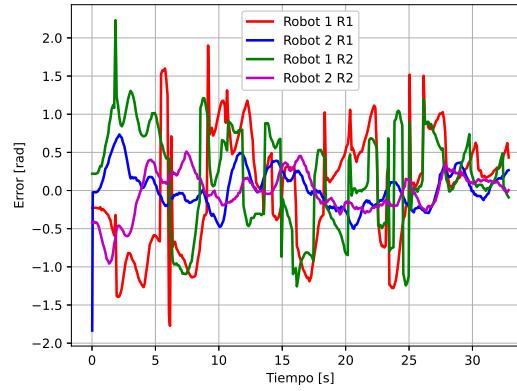
(a) Velocidad lineal.



(b) Velocidad angular.

Figura 4.34: Entradas de control del experimento 5.

La Figura (4.34a) muestra la velocidad lineal de los 4 robots. Al inicio de la navegación los robots 2 R1 y 2 R2 tienen un pico de velocidad lineal negativa, ya que buscan alinearse con el otro robot de su respectiva ruta. Junto con esto, se observa que a diferencia del entorno propuesto en los experimentos 1 a 4. Las rutas encontradas en la Figura (4.32), al no tener tramos rectos tan prolongados; demandan menos velocidad lineal en la navegación. Por otro lado, la Figura (4.34b) presenta la velocidad angular de los 4 robots. Aunque estas velocidades oscilan alrededor de cero; las rutas encontradas exigen más velocidad angular que con la que se trabajó en los experimentos 1 a 4, debido a que incrementó la cercanía entre las esquinas a lo largo del recorrido, lo que conlleva a un incremento en la frecuencia con la cual el robot requiere dar una vuelta.

(a) Error en x .(b) Error en y .

(c) Error de orientación.

Figura 4.35: Errores de navegación del experimento 5.

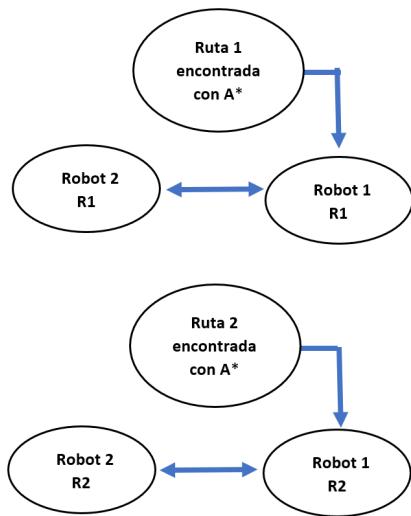
La Figura (4.35a) presenta los errores e_x de los cuatro robots. Se observa que los errores de todos los robots se aproximan a cero conforme se posicionan en los puntos de su respectiva ruta. Esto quiere decir que cada par de robots logró mantener su formación durante el recorrido.

La Figura (4.35b) muestra los errores e_y . Los cuales, oscilan alrededor de cero; sin embargo, los robots 2 R1 y 2 R2 comienzan con un valor muy grande al inicio, ya que en las condiciones iniciales de la Figura (4.32), los robots 1 R1 y 1 R2 comienzan con un desfase de 90^0 en orientación con respecto al otro robot de su ruta. Por último, la Figura (4.35c) expone los errores en orientación; los cuales, como se ha visto en los experimentos anteriores, guardan una estrecha relación con las velocidades angulares debido al control.

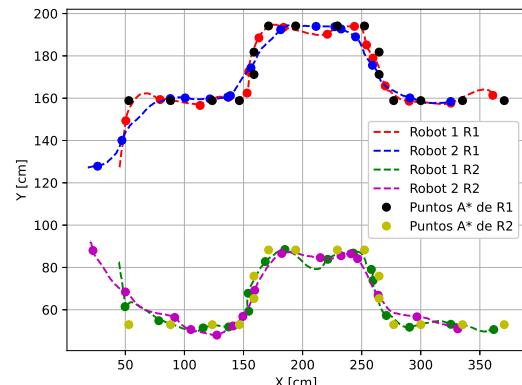
4.3.3.2. Experimento 6

El experimento 6 se realizó con una comunicación bidireccional entre cada par de robots como se observa en la Figura (4.36a). Al igual que en el experimento 5, el robot 1 R1 sólo mantiene comunicación con el robot 2 R1 y el robot 1 R2 sólo se comunica con el robot 2 R2. Este experimento tardó 42.09 en completarse. Las ganancias utilizadas fueron las mismas que en (4.24); pero se dejaron las ganancias de acoplamiento de (4.25), y se dejaron los mismos valores de las cotas. A los robots 1 R1 y 1 R2 se les agregó un desfase positivo de $\Delta = 25$ cm con respecto a los puntos de su ruta; para mantener la formación con el otro robot al cual se le asignó la misma ruta.

La Figura (4.36b) presenta las rutas recorridas de los cuatro robots. Se observa un comportamiento más preciso en el recorrido de las rutas en comparación con lo presentado en la Figura (4.33b) en el experimento 5. Esto es debido a la CB, ya que al igual que en los experimentos 2 y 4, ayuda a los robots posicionados en la parte frontal de la formación a regular de mejor manera su velocidad en los cambios de orientación.



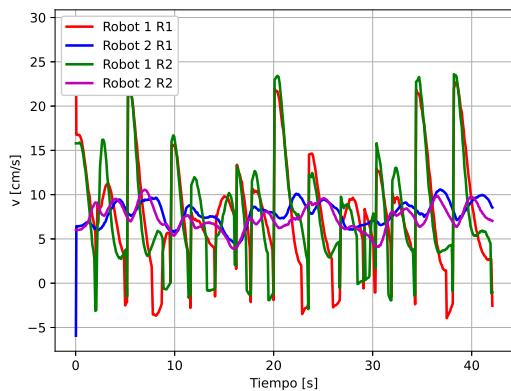
(a) Topología.



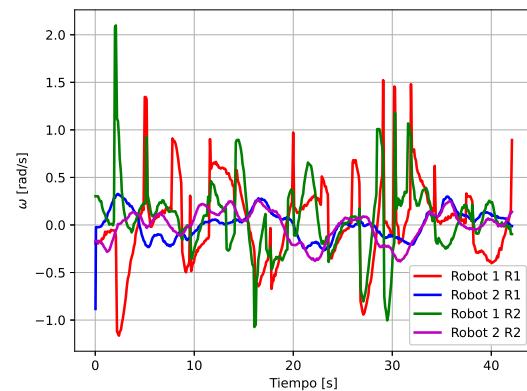
(b) Registro de rutas.

Figura 4.36: Topología y rutas recorridas del experimento 6.

La Figura (4.37a) muestra las velocidades lineales de los 4 robots. A diferencia de las velocidades presentadas en el experimento 5 en la Figura (4.34a). En el experimento 6, estas velocidades oscilan en valores negativos, ya que la CB causa que en determinados momentos los robots 1 R1 y 1 R2 se desplacen hacia atrás para mantener su distancia cuando comienzan a alejarse del otro robot de su ruta. Por otro lado, la Figura (4.37b) presenta las velocidades angulares de los robots, y cómo éstas son más elevadas en los robots 1 R1 y 1 R2, ya que reciben directamente la información de su ruta encontrada con A^* .



(a) Velocidad lineal.



(b) Velocidad angular.

Figura 4.37: Entradas de control del experimento 6.

La Figura (4.38a) presenta el error e_x de todos los robots. Se observa, que todos los errores se aproximan a cero a lo largo de la navegación. Esto quiere decir que, al igual que en los experimentos anteriores, todos los robots fueron capaces de mantener su formación en su respectiva ruta. Por otro lado, el error e_y de la Figura (4.38b) tiene un comportamiento similar al del experimento 5 en la Figura (4.35b), ya que los robots 1 R1 y 1 R2 de igual manera, comienzan con un desfase de 90° con respecto al otro robot de su ruta. Por último, el error de orientación presenta un comportamiento similar a la velocidad angular debido a la ganancia proporcional que se propuso para este error en el diseño del control.

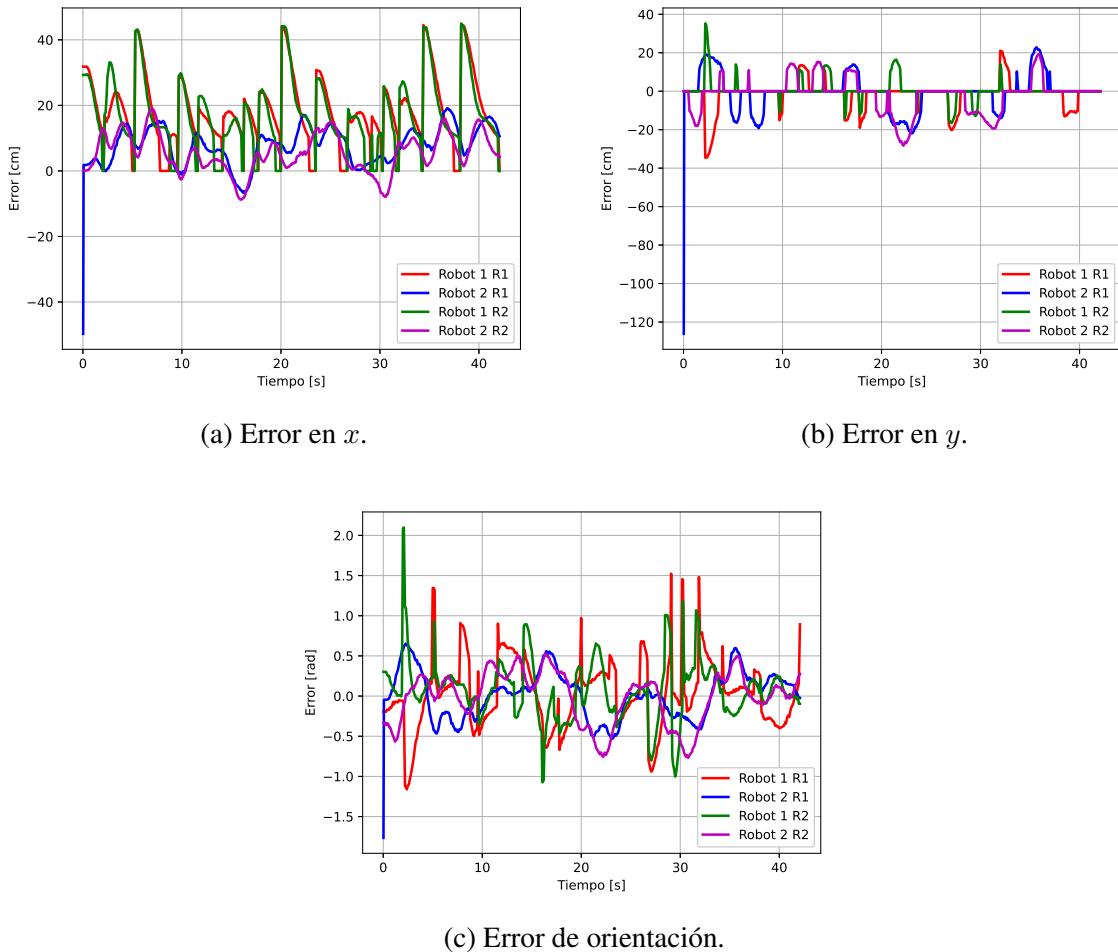


Figura 4.38: Errores de navegación del experimento 6.

¹ Capítulo 5

² Conclusiones y trabajo futuro

³ 5.1. Conclusiones

⁴ Este documento reporta una estrategia de navegación coordinada de robots móviles diferenciales en un ambiente estructurado virtual. El desarrollo de esta estrategia se fundamentó en integrar un algoritmo de control de regulación basado en consenso junto con el algoritmo A^* como método de búsqueda de rutas. La estrategia se trabajó en simulación con dos agentes y posteriormente se llevó a cabo de forma experimental variando la cantidad de robots, las interconexiones entre ellos y el entorno de navegación.

¹⁰ La estrategia de navegación desarrollada disminuyó la complejidad algorítmica cuando se requiere añadir más robots al entorno, ya que el diseño del control acoplado permitió asignar a más de un robot a una misma ruta encontrada. Esto evitó el procesamiento de búsqueda de caminos libres de obstáculos independientes para cada uno. Por otro lado, variaciones en la ubicación y cantidad de obstáculos en el entorno no afectaron a la estrategia ya que el algoritmo A^* se encarga de encontrar, siempre que exista, el camino libre de obstáculos más corto al punto de destino. Junto con esto, a pesar de que A^* ha sido ampliamente utilizado en proyectos de navegación; las rutas obtenidas no contemplan la restricción no holónoma de los robots, por lo cual, fue necesario aplicar a cada camino un posprocesamiento de recorte de esquinas, el cual permitió a los robots navegar con mejor desempeño.

²⁰ El control de regulación logró que los robots fueran capaces de seguir de forma satisfactoria los puntos de las rutas obtenidas con A^* . Esto se debió, en gran medida, al término no lineal que permitió disminuir sus velocidades angulares en los cambios de dirección. Aunado a esto, los experimentos demostraron un mejor comportamiento en la navegación cuando existe una comunicación bidireccional entre cada par de robots adyacentes. El uso de esta comunicación redujo la posibilidad de colisiones entre los robots, ya que éstos tienden a alejarse mutuamente para mantener la formación. Por otro lado, en la comunicación unidireccional, sólo el robot que reci-

1 be la información es el que busca mantener la formación. Esta característica de la comunicación
2 unidireccional, conlleva a la aparición de incrementos en las velocidades lineales y angulares de
3 los robots, debido a una menor cantidad de interconexiones entre ellos; dificultando mantener
4 la formación durante la navegación en comparación con la comunicación bidireccional.

5 **5.2. Trabajo futuro**

6 Como continuación de este trabajo de tesis, se proponen tres vertientes por las cuales es
7 posible mejorar la estrategia de navegación propuesta:

- 8 1. Implementar métodos de aproximación polinómica que permitan convertir la secuencia
9 de puntos de las rutas encontradas en trayectorias y con esto, convertir el control de re-
10 gulación, a uno de seguimiento de trayectorias que logre mejorar el desempeño de la
11 navegación.
- 12 2. Agregar obstáculos dinámicos al entorno de navegación, con el objetivo de añadir una
13 replaneación de las rutas en tiempo real. Asimismo, se deben agregar las restricciones
14 para la evasión de los obstáculos virtuales.
- 15 3. Proponer otro tipo de acoplamiento que permita a los robots mantener de forma más per-
16 sistente la formación designada y que, al mismo tiempo, permita establecer formaciones
17 más complejas entre ellos.

Apéndices

¹

¹ Apéndice A

² Publicaciones producidas durante la tesis

³ El desarrollo de la estrategia de navegación de un robot móvil con algoritmo A^* ha sido
⁴ presentada en el XXIV Congreso Mexicano de Robótica COMRob 2022 y ha sido publicada en
⁵ la revista Pádi Boletín Científico De Ciencias Básicas E Ingenierías Del ICBI
⁶ [Baizabal-Montañez et al., 2022]. Ver Figura (A.1a) y Figura (A.1b).



Reconocimiento

Por la ponencia titulada: "Navegación autónoma de robot móvil diferencial con algoritmo A^* algoritmo A estrella", en el marco del XXIV Congreso Mexicano de Robótica COMRob 2022, celebrado del 09 al 11 de noviembre del año en curso, en las instalaciones del Instituto de Ciencias Básicas e Ingeniería.

Mineral de la Reforma, Hgo., a 11 de noviembre de 2022.

Atentamente
"Amor, Orden y Progreso"

Dr. Omar Arturo Domínguez Ramírez
Presidente del XXIV COMRob 2022

ICBI ICBI-DI1384/2022

Dr. Jesús Alberto Sandoval Galarza
Presidente de la AMRob 2020-2023

Dr. J. Patricio Orta Salive
Presidente del Programa Científico



Navegación autónoma de robot móvil diferencial con algoritmo A^*
Differential mobile robot autonomous navigation with algorithm A^*

E. Baizabal-Montañez, N. González-Yanes, A. Morales-Díaz, K. Ordaz-Hernández

*Grupo de Robótica y Manufactura Avanzada, Centro de Investigación y de Estudios Avanzados del IPN-Saltillo, Industrial, Zona Industrial, 25960 Ramos Arizpe, Coahuila, México.

Resumen

Con la finalidad de que un robot móvil diferencial pueda navegar en entornos con obstáculos, en el presente trabajo se aplica el algoritmo A^* , uno de los algoritmos más populares para encontrar el camino más corto entre dos nodos de un grafo. Debido a la restricción no-holónoma que poseen este tipo de robots, la trayectoria obtenida con el algoritmo A^* se post-processa, para que posteriormente sea ejecutada por el robot mediante un controlador no-lineal aquí propuesto. También se presenta la estabilidad del controlador haciendo uso del método de Lyapunov. Finalmente el desempeño del robot es probado en experimentos.

Palabras Clave: Robótica Móvil, Planeación de Trajetorias, Optimización, Control No Lineal.

Abstract

In order for a differential mobile robot to navigate in environments with obstacles, this paper applies the A^* algorithm, one of the most popular algorithms to find the shortest path between two nodes of a graph. Due to the nonholonomic constraint of this type of robots, the path obtained with A algorithm is post-processed, to be subsequently executed by the robot by means of a nonlinear controller proposed here. The stability of this controller is also presented using the Lyapunov method. Finally, the performance of the robot is tested in experiments.

Keywords: Mobile Robotics, Path Planning, Optimization, Non-linear Control.

1. Introducción

El creciente interés en los robots móviles se debe a las numerosas aplicaciones que tienen en el área médica (Xiaofei y Gang, 2018), industrial (Yuki y Shuki, 2022), agrícola (Luis y Herreid, 2022) y doméstica (Tatsuya y Yuki, 2022); en éstas áreas se requiere que los robots puedan planificar rutas de manejo en entornos observados desde su posición inicial hacia un objetivo. Para que los robots puedan realizar sus tareas en distintas zonas del entorno, es necesario que lo conozcan y encuentren otros caminos entre 2 puntos a través de un entorno complejo (y Hernández), posteriormente seleccionar el más corto y seguro.

El problema de la planeación y optimización de trayectorias en entornos complejos se ha abordado por métodos de búsqueda en grafos como lo son A^* , Dijkstra y D^* , aunque también existen otros métodos basados en algoritmos genéticos y cam-

pos potenciales.

En el presente trabajo se propone una estrategia de planeación de ruta basada en grafos, capaz de comparar el costo de cada camino encontrado y seleccionar el más corto (S. Guanwan y Cahyadi, 2019) en un entorno estructurado mediante el algoritmo A^* . Para que un robot diferencial siga el camino seleccionado se propone una estrategia de ajuste de ruta y un controlador no lineal.

2. Estrategia de planificación de ruta

La búsqueda de caminos es un problema que usualmente se resuelve por medio de algoritmos de búsqueda en grafos; como lo son Dijkstra y A^* . En este tipo de problemas se conoce el estado inicial, el estado final (meta) y un conjunto de reglas mediante las cuales se pueden realizar desplazamientos para que

*Autor para correspondencia: edgar.baizabal@ciencias.mx

Correos electrónicos: edgar.baizabal@ciencias.mx (Edgar Baizabal Montañez), nerub.gonzalez@ciencias.mx (Nerubia J. González-Yanes), ameli.morales@ciencias.mx (Amelia Morales-Díaz), kathy.ordaz@ciencias.mx (Kathy Ordaz Hernández).

Historial del manuscrito: recibido el 27/10/2022, última versión revisada recibida el 28/10/2022, aceptado el 07/11/2022, publicado el 11/11/2022. DOI: <https://doi.org/10.29057/cib.v10i5.10145>

(a) Presentación en COMRob 2022.

(b) Publicación en revista Pádi.

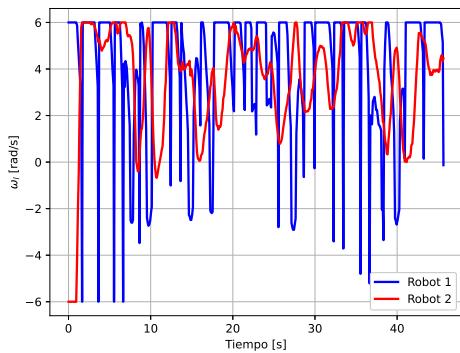
1 Apéndice B

2 **Velocidades experimentales de las ruedas 3 de los robots**

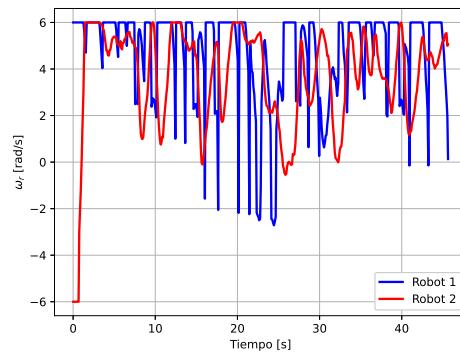
4 Los resultados experimentales del capítulo 4 mostraron las velocidades lineales y angulares
5 de los robots durante la navegación. Para mandar estas velocidades a cada robot, deben ser
6 previamente mapeadas a velocidades de rueda izquierda y derecha de acuerdo a la relación
7 presentada en la ecuación (3.10) del capítulo 3. Una vez mapeadas, se ingresan a una función
8 de saturación; para mantener la integridad de los actuadores. En el caso de los robots utilizados,
9 se implementó la función de saturación presentada en (4.21). Finalmente, estas velocidades son
10 enviadas por comunicación serial al control de bajo nivel de cada robot, como se presenta en el
11 diagrama de la Figura (4.8). A continuación se muestra el registro de estas velocidades en los
12 seis experimentos realizados.

13 **Experimento 1**

14 Las velocidades de rueda izquierda y derecha de la Figura (B.1), se obtuvieron a partir de
las velocidades lineales y angulares de la Figura (4.19).



(a) Velocidades de rueda izquierda.

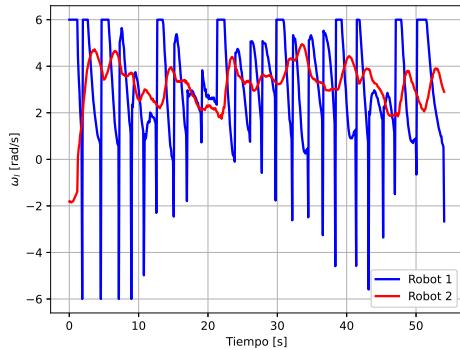


(b) Velocidades de rueda derecha.

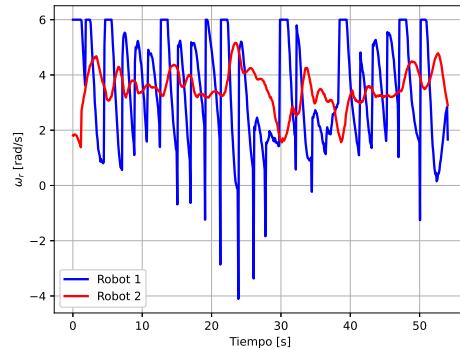
Figura B.1: Velocidades de rueda izquierda y derecha de los robots en el experimento 1.

1 Experimento 2

2 Las velocidades de rueda izquierda y derecha de la Figura (B.2), se obtuvieron a partir de las velocidades lineales y angulares de la Figura (4.22).



(a) Velocidades de rueda izquierda.



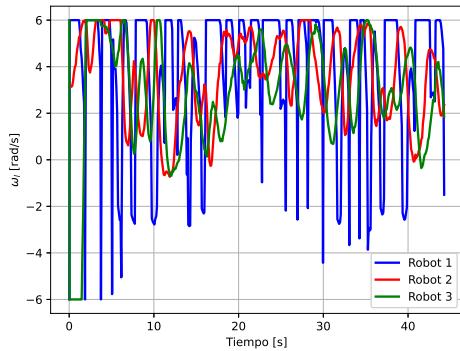
(b) Velocidades de rueda derecha.

Figura B.2: Velocidades de rueda izquierda y derecha de los robots en el experimento 2.

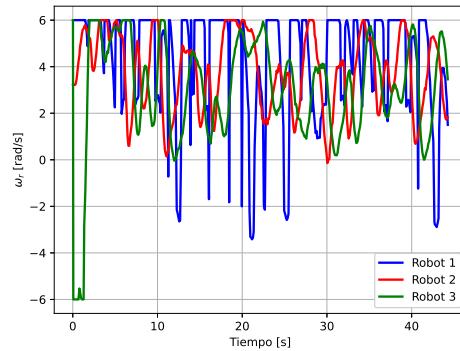
3

4 Experimento 3

5 Las velocidades de rueda izquierda y derecha de la Figura (B.3), se obtuvieron a partir de las velocidades lineales y angulares de la Figura (4.26).



(a) Velocidades de rueda izquierda.



(b) Velocidades de rueda derecha.

Figura B.3: Velocidades de rueda izquierda y derecha de los robots en el experimento 3.

6

1 Experimento 4

- 2 Las velocidades de rueda izquierda y derecha de la Figura (B.4), se obtuvieron a partir de las velocidades lineales y angulares de la Figura (4.30).

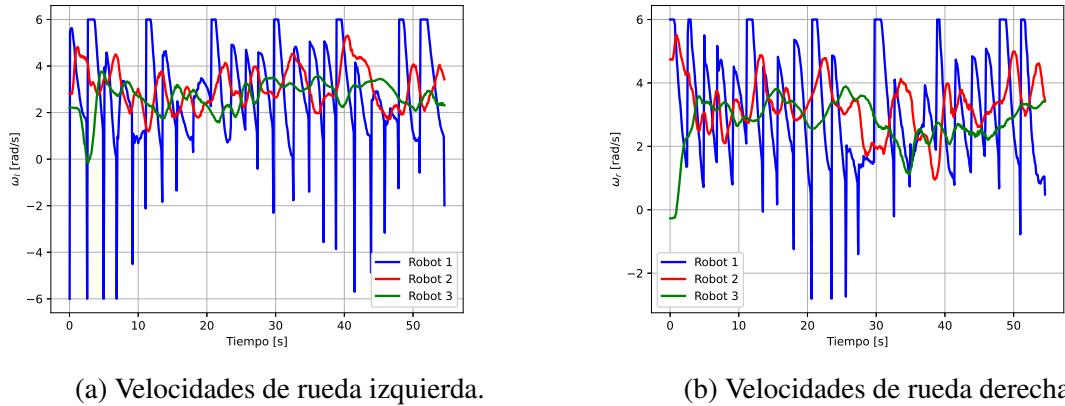


Figura B.4: Velocidades de rueda izquierda y derecha de los robots en el experimento 4.

3

4 Experimento 5

- 5 Las velocidades de rueda izquierda y derecha de la Figura (B.5), se obtuvieron a partir de las velocidades lineales y angulares de la Figura (4.34).

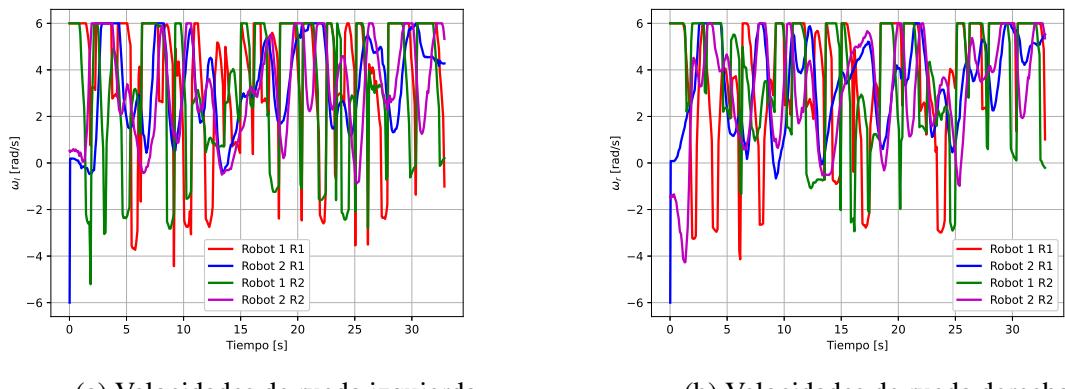
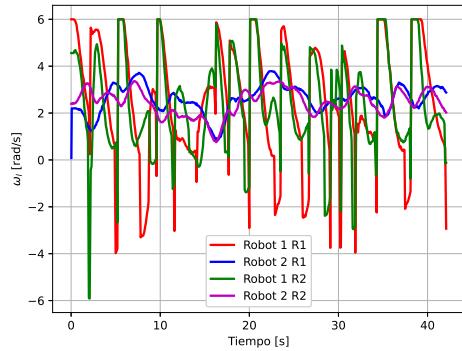


Figura B.5: Velocidades de rueda izquierda y derecha de los robots en el experimento 5.

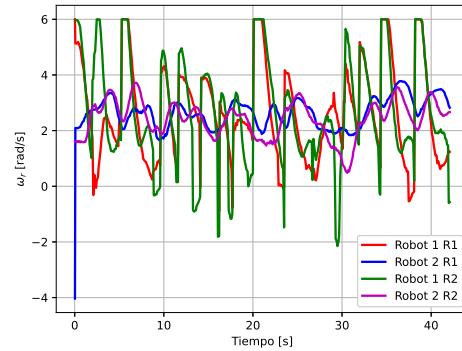
6

Experimento 6

- Las velocidades de rueda izquierda y derecha de la Figura (B.6), se obtuvieron a partir de las velocidades lineales y angulares de la Figura (4.37).



(a) Velocidades de rueda izquierda.



(b) Velocidades de rueda derecha.

Figura B.6: Velocidades de rueda izquierda y derecha de los robots en el experimento 6.

¹ Bibliografía

- ² [Alyasin et al., 2019] Alyasin, A., Abbas, E., and Hasan, S. (2019). An efficient optimal path
³ finding for mobile robot based on dijkstra method. *4th Scientific International Conference –*
⁴ *Najaf – IRAQ.*
- ⁵ [Andriyanov et al., 2022] Andriyanov, N., Khasanshin, I., Utkin, D., Gataullin, T., Ignar, S.,
⁶ Shumaev, V., and Soloviev, V. (2022). Intelligent system for estimation of the spatial position
⁷ of apples based on yolov3 and real sense depth camera d415. *Symmetry.*
- ⁸ [Baizabal-Montañez et al., 2022] Baizabal-Montañez, E., Gonzalez-Yances, N. J., Morales-
⁹ Díaz, A., and Ordaz-Hernández, K. (2022). Navegación autónoma de robot móvil diferen-
¹⁰ cial con algoritmo A*. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI,*
¹¹ 10(Especial5):67–72.
- ¹² [Bravo, 2018] Bravo, R. J. (2018). Sistema de seguimiento de objetos usando opencv, aruco y
¹³ filtro de kalman extendido. Master's thesis, Universidad de Sevilla.
- ¹⁴ [Chudý et al., 2020] Chudý, J., Popov, N., and Surynek, P. (2020). Emulating centralized con-
¹⁵ trol in multi-agent pathfinding using decentralized swarm of reflex-based robots. *2020 IEEE*
¹⁶ *International Conference on Systems, Man, and Cybernetics (SMC).*
- ¹⁷ [D'Andrea, 2012] D'Andrea, R. (2012). Guest editorial a revolution in the warehouse: A retros-
¹⁸ pectiveon kiva systems and the grand challenges ahead. *IEEE Transactions on automation*
¹⁹ *science and engineering.*
- ²⁰ [Duo et al., 2021] Duo, Q., Junhua, H., Xiaolong, L., Jiaqiang, Z., and Zhihao, Z. (2021). Re-
²¹ search on consensus of multi-agent systems with and without input saturation constraints.
²² *Journal of Systems Engineering and Electronics.*
- ²³ [Grenouilleau et al., 2019] Grenouilleau, F., van Hoeve, W.-J., and Hooker, J. N. (2019). A
²⁴ multi-label A star algorithm for multi-agent pathfinding. *Association for the Advancement*
²⁵ *of Artificial Intelligence.*
- ²⁶ [Gunawan et al., 2019] Gunawan, S., Pratama, G., Cahyad, A., Winduratna, B., Yuwono, Y.,
²⁷ and Wahyunggoro, O. (2019). Smoothed A-star algorithm for nonholonomic mobile robot
²⁸ path planning. *International Conference on Information and Communications Technology*
²⁹ *(ICOIACT).*

- 1 [Gutiérrez et al., 2017] Gutiérrez, H., Morales, A., and Nijmeijer, H. (2017). Synchronization
2 control for a swarm of unicycle robots: Analysis of different controller topologies. *Asian
3 Journal of Control.*
- 4 [Hönig et al., 2019] Hönig, W., Kiesel, S., Tinka, A., Durham, J. W., and Ayanian, N. (2019).
5 Persistent and robust execution of mapf schedules in warehouses. *IEEE ROBOTICS AND
6 AUTOMATION LETTERS.*
- 7 [Karur et al., 2021] Karur, K., Sharma, N., Dharmatti, C., and Siegel, J. (2021). A survey of
8 path planning algorithms for mobile robots.
- 9 [Kent D. Lee, 2015] Kent D. Lee, S. H. (2015). *Data Structures and Algorithms with Python.*
10 Springer International Publishing.
- 11 [Khalil, 1996] Khalil, H. K. (1996). *Nonlinear Systems.* PRENTICE HALL.
- 12 [Koung et al., 2020] Koung, D., Fantoni, I., Kermorgant, O., and Belouaer, L. (2020).
13 Consensus-based formation control and obstacle avoidance for nonholonomic multi-robot
14 system. *International Conference on Control, Automation, Robotics and Vision (ICARCV).*
- 15 [Kuang et al., 2022] Kuang, H., Li, Y., Zhang, Y., and Feng, Y. (2022). Improved A-star algo-
16 rithm based on topological maps for indoor mobile robot path planning. *IEEE ITOEC.*
- 17 [Lester, 2014] Lester, P. (2014). A* pathfinding for beginners.
- 18 [Li and Liu, 2016] Li, J.-t. and Liu, H.-j. (2016). Design optimization of amazon robotics.
19 *Automation, Control and Intelligent Systems.*
- 20 [Liu et al., 2019] Liu, Y., Chen, M., and Huang, H. (2019). Multi-agent pathfinding based on
21 improved cooperative A* in kiva system. *International Conference on Control, Automation
and Robotics.*
- 23 [Malu and Majumdar, 2014] Malu, S. K. and Majumdar, J. (2014). Kinematics, localization
24 and control of differential drive mobile robot. *Global Journal of researches in engineering:
H Robotics and Nano-Tech.*
- 26 [Ning et al., 2020] Ning, B., Han, Q.-L., and Lu, Q. (2020). Fixed-time leader-following con-
27 sensus for multiple wheeled mobile robots. *IEEE Transactions on cybernetics.*
- 28 [Oriolo et al., 2002] Oriolo, G., Luca, A. D., and Vendittelli, M. (2002). Wmr control via dy-
29 namic feedback linearization:design, implementation, and experimental validation. *IEEE
30 TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY.*
- 31 [Randal Beard, 2007] Randal Beard, W. R. (2007). *Distributed Consensus in Multi-vehicle
32 Cooperative Control.* Springer London.
- 33 [Rivest et al., 2009] Rivest, R. L., Cormen, T. H. D. C., Leiserson, C. E. M., and Stein, C. C. U.
34 (2009). *Introduction to Algorithms.* MIT Press Ltd.

- 1 [Roy, 2019] Roy, B. (2019). A star search algorithm. *Towards Data Science*.
- 2 [Savina et al., 2019] Savina, A. M., Lucia, B. J. G., Romané, G. F. M., and Julian, M. L. C.
3 (2019). Inteligencia artificial (búsqueda A estrella).
- 4 [Sharon et al., 2021] Sharon, G., Stern, R., Felner, A., and Sturtevant, N. (2021). Conflict-
5 based search for optimal multi-agent path finding. *Proceedings of the Twenty-Sixth AAAI
6 Conference on Artificial Intelligence*.
- 7 [Siciliano et al., 2010] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2010). *Robotics:
8 Modelling, Planning and Control*. SPRINGER NATURE.
- 9 [Siegwart et al., 2011] Siegwart, R. A. S. L., of Robotics, I. R. P., Nourbakhsh, C. M. U., and
10 of Zurich) Scaramuzza, D. U. (2011). *Introduction to Autonomous Mobile Robots*. MIT
11 Press Ltd.
- 12 [Silver, 2005] Silver, D. (2005). Cooperative pathfinding. *American Association for Artificial
13 Intelligence*.
- 14 [Takeyas, 2005] Takeyas, B. L. (2005). Inteligencia artificial.
- 15 [Wang, 2021] Wang, B. (2021). Path planning of mobile robot based on A star algorithm.
16 *IEEE International Conference on Electronic Technology, Communication and Information
17 (ICETCI)*.
- 18 [Zhong and Lv, 2021] Zhong, Y. and Lv, Y. (2021). Appointed-weight privacy-preserving
19 consensus of multi-agent systems. *International Conference on Neuromorphic Computing
20 (ICNC)*.