



## GRADO EN INGENIERÍA DE ROBÓTICA SOFTWARE

Escuela de Ingeniería de Fuenlabrada

Curso académico 2023-2024

### Trabajo Fin de Grado

Programación de flujo de datos en  
multirobótica con ROS2 y Zenoh-Flow

**Tutor:** Julio Vega Pérez

**Autor:** Unai Sanz Conejo



Este trabajo se distribuye bajo los términos de la licencia internacional CC BY-NC-SA International License (Creative Commons AttributionNonCommercial-ShareAlike 4.0). Usted es libre de *(a) compartir*: copiar y redistribuir el material en cualquier medio o formato; y *(b) adaptar*: remezclar, transformar y crear a partir del material. El licenciador no puede revocar estas libertades mientras cumpla con los términos de la licencia:

- *Atribución.* Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciatante.
- *No comercial.* Usted no puede hacer uso del material con propósitos comerciales.
- *Compartir igual.* Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.

# Agradecimientos

---

Quisiera expresar mi más sincero agradecimiento a todas las personas que contribuyeron a la realización de este trabajo. En primer lugar, agradezco al equipo de ZettaScale por darme la oportunidad de realizar mis prácticas con ellos y de aprender incontables aspectos acerca de las telecomunicaciones y por su persistente ayuda. También agradezco a mi tutor de TFG por su orientación constante y su gran paciencia a lo largo de este proceso.

Además estoy profundamente agradecido a mis compañeros de clase por sus ideas y debates constructivos, que han enriquecido enormemente mi investigación.

No puedo dejar de agradecer a mis padres, hermana, amigos cercanos y pareja por sus ideas, consejos y apoyo moral e incondicional.

Por último, pero no menos importante, quiero expresar mi gratitud a todas las fuentes y recursos que consulté durante la elaboración de este trabajo, así como a cualquier institución o persona que haya contribuido de alguna manera, aunque indirecta, a este proyecto.

Sin el apoyo de todas estas personas y entidades, este trabajo no habría sido posible. Gracias de todo corazón.

*A mi abuelo;  
que estaría sumamente orgulloso de mi trabajo.*

Madrid, xx de xxxxxxx de 2024

*Unai Sanz*

# Resumen

---

Escribe aquí el resumen del trabajo. Un primer párrafo para dar contexto sobre la temática que rodea al trabajo.

Un segundo párrafo concretando el contexto del problema abordado.

En el tercer párrafo, comenta cómo has resuelto la problemática descrita en el anterior párrafo.

Por último, en este cuarto párrafo, describe cómo han ido los experimentos.

# Acrónimos

---

**DDS** *Data Distribution Service*

**GNU** *GNU's Not Unix (acrónimo recursivo)*

**JAXA** *Japan Aerospace Exploration Agency*

**LIDAR** *Light Detection and Ranging / Laser Imaging Detection and Ranging*

**LTS** *Long-Term Support*

**NASA** *National Aeronautics and Space Administration*

**PDCA** *Plan, Do, Check, Act*

**RMW** *ROS Middleware Interface*

**ROS2** *Robotic Operating System (ROS) 2*

**STEM** *Ciencia, Tecnología, Ingeniería y Matemáticas*

**UPC** *Universidad Politécnica de Cataluña*

**URJC** *Universidad Rey Juan Carlos*

# Índice general

---

<b>1. Introducción</b>	<b>1</b>
1.1. La evolución y el estado de la robótica . . . . .	1
1.2. La importancia de la robótica en la ciencia . . . . .	1
1.3. La robótica móvil en relación con la educación . . . . .	3
1.4. La educación en robótica . . . . .	4
1.5. La robótica de bajo coste . . . . .	6
1.6. La multirobótica . . . . .	8
1.7. Flujos de datos en robótica . . . . .	10
<b>2. Objetivos</b>	<b>13</b>
2.1. Descripción del problema . . . . .	13
2.2. Requisitos . . . . .	13
2.3. Metodología . . . . .	15
2.4. Plan de trabajo . . . . .	16
<b>3. Plataforma de desarrollo</b>	<b>18</b>
<b>4. Diseño</b>	<b>19</b>
4.1. Snippets . . . . .	19
4.2. Verbatim . . . . .	19
4.3. Ecuaciones . . . . .	20
4.4. Tablas o cuadros . . . . .	20
<b>5. Conclusiones</b>	<b>22</b>
5.1. Conclusiones . . . . .	22
5.2. Corrector ortográfico . . . . .	23
<b>Bibliografía</b>	<b>24</b>

# Índice de figuras

---

1.1.	Rover Perseverance y helicóptero Ingenuity de la NASA en Marte. . . . .	2
1.2.	Robots Spot (izq.) y Atlas (dcha.) de Boston Dynamics. . . . .	3
1.3.	Arduino. . . . .	5
1.4.	Código de Arduino en Scratch. . . . .	5
1.5.	Arquitectura de ROS y ROS2. . . . .	7
1.6.	Sora-Q, robot lunar <i>low-cost</i> desarrollado por Takara Tomy y la JAXA. .	8
1.7.	Múltiples robots durante una operacion de búsqueda y rescate. . . . .	9
1.8.	Robots educativos, modelos Turtlebot 2 (arriba) y Turtlebot 4 (abajo). .	10
1.9.	Flujo de datos de un robot para la detección y seguimiento de códigos QR. . . . .	12
1.10.	Comparación de un flujo de datos con los componentes de un robot. . .	12
2.1.	Esquema del desarrollo software iterativo. . . . .	15

# Listado de códigos

---

4.1. Función para buscar elementos 3D en la imagen . . . . .	19
4.2. Cómo usar un Slider . . . . .	20

# Listado de ecuaciones

---

4.1. Ejemplo de ecuación con fracciones . . . . .	20
4.2. Ejemplo de ecuación con array y letras y símbolos especiales . . . . .	20

# Índice de cuadros

---

4.1. Parámetros intrínsecos de la cámara . . . . .	21
--	----

---

# Capítulo 1

## Introducción

---

*El éxito es la suma de pequeños esfuerzos repetidos día tras día.*

Robert Collier

### 1.1. La evolución y el estado de la robótica

La robótica es un campo multidisciplinario que se concentra en el diseño, construcción, programación y operación de robots. Estos dispositivos electromecánicos, con frecuencia modelados antropomórficamente o zoomórficamente, están destinados a realizar tareas de manera autónoma o semiautónoma en una variedad de entornos, para lo que disponen de sensores que les proveen de información del medio que les rodea, de cierta capacidad de cómputo para tomar decisiones acerca de estos datos recolectados y de actuadores que les permiten interactuar con el mismo y llevar a cabo dichas decisiones, por lo que sus capacidades y limitaciones están determinadas por su *hardware*, y su inteligencia reside en su *software*.

Este gran campo de estudio e investigación ha experimentado un rápido crecimiento y expansión desde sus inicios en la década de 1950, abarcando una amplia gama de aplicaciones en la industria, la medicina, el entretenimiento y la exploración espacial, entre muchas otras, y se encuentra en constante evolución, desempeñando un papel cada vez más importante en nuestra sociedad moderna, gracias en gran medida a los incesantes avances en tecnologías como la inteligencia artificial, los sensores, los procesadores y los actuadores.

### 1.2. La importancia de la robótica en la ciencia

En particular, la robótica ha conformado un factor crucial en la exploración espacial, y esta ha impulsado la creación de nuevas tecnologías que han sido desarrolladas

exclusivamente para ciertas misiones espaciales pero que luego se han aplicado en la Tierra, mejorando la calidad de vida de las personas. Ejemplos destacados incluyen los sistemas de purificación de agua, los tejidos avanzados como la viscoelástica, los pañales y los dispositivos de imágenes médicas como la resonancia magnética, que han proporcionado acceso a las personas respectivamente a agua potable en regiones remotas, a colchones y almohadas que promueven un mejor descanso, a mayores facilidades en cuanto al cuidado de los niños y mayores y a diagnósticos médicos precisos sin radiación nociva, lo cual ha salvado innumerables vidas. De esta manera, la investigación espacial no solo expande nuestro conocimiento del universo, sino que también beneficia directamente a la humanidad en la Tierra.

En cuanto al papel de la robótica en este campo, podemos destacar ejemplos como los resistentes robots enviados a diferentes planetas y lunas del sistema solar en busca de datos científicos. Numerosas misiones científicas lo evidencian, como la misión *Mars 2020* de la NASA, cuyos robots se ven ilustrados en la Figura 1.1, tomada por uno de los robots, el rover Perseverance, que depositó con éxito al segundo robot sobre la superficie marciana, el helicóptero Ingenuity, que aparece más al fondo en la imagen, y que ayudó al rover a llevar a cabo su exploración y toma de muestras. Este helicóptero estaba diseñado para realizar 5 vuelos durante un mes a modo de demostrador tecnológico, pero su misión pudo alargarse hasta casi los 3 años, momento en el que termina debido a la rotura de una de sus palas, sumando entonces un total de 72 vuelos.



Figura 1.1: Rover Perseverance y helicóptero Ingenuity de la NASA en Marte.

Para el éxito de esta misión fue clave la investigación en múltiples campos de la robótica, como la robótica móvil y todos los campos que esta conlleva, como pueden ser la visión artificial, la navegación o la localización, áreas clave que no solo están

redefiniendo los límites de la tecnología, sino que también tienen un gran impacto en cómo usamos la tecnología en nuestra vida diaria, como ha sucedido por ejemplo, con las aspiradoras robóticas o la conducción autónoma, que inevitablemente ya forman parte de nuestra sociedad.

### 1.3. La robótica móvil en relación con la educación

La robótica móvil ha emergido como un campo multidisciplinario que fusiona la ingeniería, la inteligencia artificial y múltiples ramas de la robótica y la mecatrónica para crear sistemas capaces de moverse y operar en entornos dinámicos, aprovechando áreas como la robótica de campo, la creación de mapas, la localización y la navegación con ayuda de otros como la visión artificial o la manipulación de objetos.

Desde sus inicios, ha sido impulsada por los avances tecnológicos, permitiendo su aplicación en una amplia gama de campos, desde la exploración espacial y submarina, hasta la logística industrial y la atención médica, siendo ya parte indispensable de nuestras vidas y mejorando la calidad de las mismas. En este contexto, la investigación en robótica móvil se centra en desarrollar sistemas autónomos capaces de navegar de manera segura y eficiente en entornos conocidos o desconocidos, adaptarse a cambios imprevistos y realizar tareas complejas de manera autónoma.

Un ejemplo representativo de este tipo de robots se puede ver en la Figura 1.2, donde se pueden observar dos de los robots más desarrollados en el ámbito móvil, que han demostrado una gran versatilidad en una variedad de entornos para ejecutar una amplia variedad de tareas, desde abrir puertas, pasando por transportar cargas de peso o realizar trabajos manuales, hasta incluso seguir rutinas de deportivas variadas, que en muchos casos iguala o incluso supera la de los humanos.



Figura 1.2: Robots Spot (izq.) y Atlas (dcha.) de Boston Dynamics.

En concreto, la localización de los robots juega una papel importante en la robótica móvil debido a su indispensable necesidad a la hora de navegar por el entorno normalmente tomando puntos de referencia gracias a los sensores y creando un modelo probabilístico de la posición del robot basado en estos datos.

Por su parte, para el correcto funcionamiento de la navegación, es indispensable conocer la posición del robot durante el movimiento del mismo, para tener la capacidad de sortear obstáculos y poder desplazar el robot al objetivo.

La robótica móvil también forma parte de campos más grandes y complejos e incluso sienta las bases de algunos de ellos, como sucede con la multirobótica, que representa un paso adelante en la complejidad y la escala de los sistemas robóticos individuales, y permite realizar tareas que un solo robot no es capaz de hacer, o realizarlas mucho mas rápidamente o eficientemente. Como ejemplo de esta ventaja, son notables ciertos trabajos realizados sobre localización con múltiples robots, como en el artículo [Trawny et al., 2009], en el que se ponen a prueba las mismas técnicas utilizadas para un solo robot y evalúan su viabilidad en un entorno unidimensional. También se han realizado trabajos enfocados a entornos tridimensionales, como se observa en el artículo [Fox et al., 2000], en el cuál se logra una localización basada en sensores, teniendo en cuenta la posición de los demás robots y aumentando de este modo la precisión de la localización del propio robot en cuestión.

Además, este campo supone una gran oportunidad para el aprendizaje, ya que juega un papel crucial en el desarrollo de habilidades tecnológicas a la vez que en la motivación de los estudiantes, los cuales obtienen una gran sensación de realización y entusiasmo por aprender, al poder visualizar los resultados en movimiento de manera autónoma.

## 1.4. La educación en robótica

Debido a la creciente participación de los robots móviles en nuestras vidas diarias, como se ha hecho notar en el caso de las aspiradoras robóticas, la relevancia de la robótica en el ámbito educativo ha ido ganando terreno en los últimos años, tanto en la comunidad europea como en otros muchos países. Esto ha dado lugar a un aumento en la implementación de programas educativos que han incluido actividades prácticas de robótica en escuelas de educación primaria y secundaria, promoviendo la creatividad, el pensamiento crítico y las habilidades tecnológicas entre los estudiantes, además de fomentar el trabajo en equipo y la resolución de problemas complejos, preparándolos

para futuros grados o carreras relacionadas con la tecnología, cuya demandada aumenta incesablemente.

En el caso de España, desde la década de los 90, se han implementado programas piloto y competiciones robóticas, como el programa Robolot<sup>1</sup> (1992), desarrollado por la UPC, las Olimpiadas de Informática<sup>2</sup> (1993), que incorporaron desafíos relacionados con la programación de robots, así como la RoboCupJunior<sup>3</sup> (2000), ofreciendo a los estudiantes la oportunidad de diseñar, construir y programar robots para competir en diferentes categorías. Desde alrededor de 2014, dependiendo de la comunidad autónoma de España, se han introducido programas y asignaturas que incluyen la robótica como parte esencial del plan de estudios.

Al ser necesario un contexto simple y barato para la introducción a la robótica, en la educación se buscan herramientas como las placas Arduino (Figura 1.3), que presentan una amplia compatibilidad con distintos sensores y actuadores y brinda un entorno sencillo para aquellos que se están introduciendo en este campo y plataformas como *Scratch*<sup>4</sup> para simplificar la programación gracias a su interfaz de bloques y la asociación de ideas a colores, como se muestra en la Figura 1.4.



Figura 1.3: Arduino.

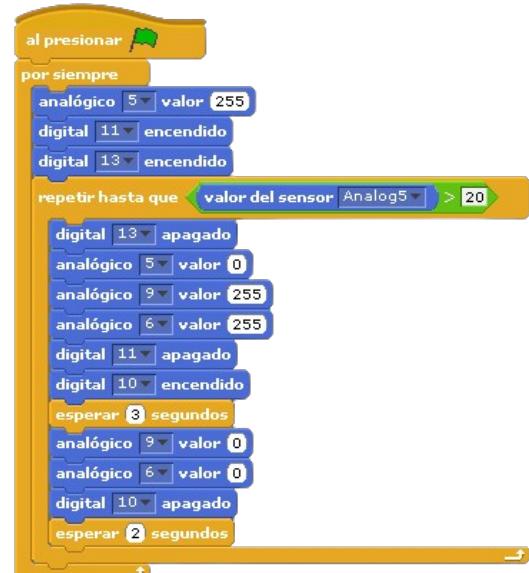


Figura 1.4: Código de Arduino en Scratch.

Las placas utilizadas en el ámbito educativo, mencionadas anteriormente, resultan ideales para estos propósitos debido a su coste y simplicidad, sin embargo, también

<sup>1</sup><https://www.robolot.online/>

<sup>2</sup><https://olimpiada-informatica.org/>

<sup>3</sup><https://junior.robocup.org/>

<sup>4</sup><https://scratch.mit.edu/>

imponen ciertas limitaciones en las capacidades del robot y en la posibilidad de añadir *hardware* externo más complejo y potente, como cámaras o LIDARs, o actuadores como motores más potentes que a menudo requieren de mayor alimentación eléctrica de la que estas placas pueden brindar. Esto genera trabas a la propia originalidad y aprendizaje de los estudiantes, restringiendo así su creatividad, innovación y potencial de creación, una vez se han obtenido unos conocimientos básicos.

Es entonces cuando se encuentra el desafío de dar el siguiente paso: la programación de robots utilizando ROS, el *middleware* estándar por excelencia en robótica. Este proceso implica un considerable escalón de aprendizaje, ya que no solo se debe dominar un lenguaje de programación más complejo, sino que también se debe comprender el entorno que rodea a esta plataforma, en el que se incluyen campos de la robótica como son las comunicaciones, la arquitectura *software*, la programación modular y orientada a objetos, algoritmos y estructuras de datos, entre otros muchos, y que suelen conllevar decenas de asignaturas con identidad propia en cualquier grado de universidad. La teoría de la existencia de una brecha educativa en este ámbito se ve respaldada por trabajos como la tesis doctoral de [Vega, 2018a], en cuyas secciones A.3 y A.4, se analiza esta misma perspectiva y se propone una solución respectivamente.

Dicha complejidad puede verse ilustrada en la Figura 1.5, en la que se muestra un esquema simplificado de la arquitectura de ambas versiones de este *middleware* que puede resultar abrumadora para aquellos que se están iniciando en este ámbito, ya que se muestra la creciente complejidad adquirida al pasar de ROS, que ya era suficientemente complejo, a ROS2, en el que ahora existen distintas implementaciones de *middleware* de comunicaciones, y otras capas como al de transporte, el sistema operativo y el *hardware*.

Por este motivo, resulta evidente la necesidad de un paso intermedio que pueda actuar como puente entre estos dos niveles de aprendizaje, el cual podría ser incorporado, por ejemplo, en el programa educativo del bachillerato tecnológico o de centros de educación secundaria. Este nivel intermedio facilitaría la transición entre las habilidades adquiridas en la enseñanza primaria o secundaria y los requisitos más avanzados de la universidad en el campo de la robótica.

## 1.5. La robótica de bajo coste

La robótica de bajo coste se refiere al desarrollo e implementación de sistemas robóticos como los descritos anteriormente utilizando componentes y recursos económi-

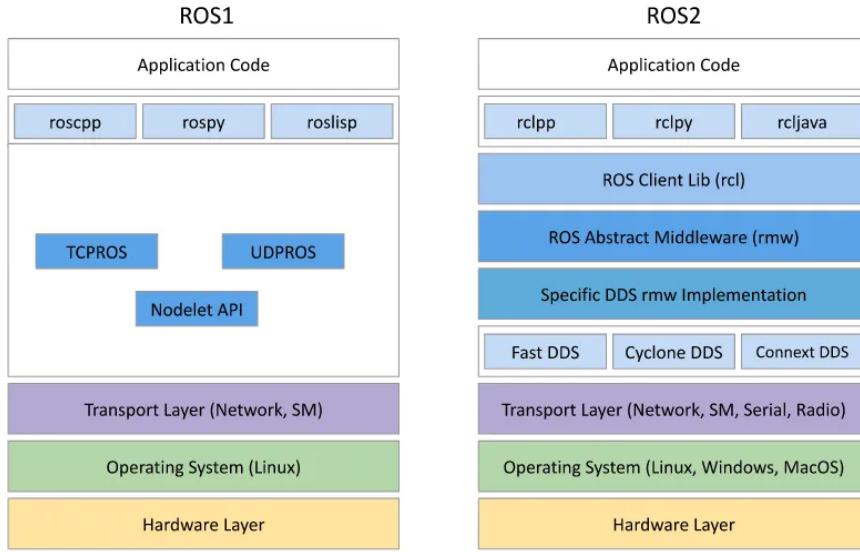


Figura 1.5: Arquitectura de ROS y ROS2.

cos, con el objetivo de hacer la tecnología robótica más accesible y asequible para una amplia gama de aplicaciones y usuarios. Este enfoque busca reducir los costes asociados con la construcción y operación de robots, empleando materiales económicos, hardware de bajo coste y técnicas de fabricación eficientes.

En el contexto de la robótica móvil, los sistemas de bajo coste pueden ofrecer soluciones viables para aplicaciones con presupuestos limitados o despliegues a gran escala, abarcando un papel crucial en áreas como la educación, la investigación académica, la asistencia social y la exploración de entornos remotos o peligrosos. Además de su utilidad práctica, la robótica de bajo coste también promueve la innovación y el desarrollo de nuevas tecnologías al proporcionar una plataforma accesible para la experimentación y la creatividad abierta a una amplia comunidad, así como a la educación.

Un ejemplo destacado de este tipo de robótica es el robot Sora-Q representado en la Figura 1.6, enviado a la Luna recientemente por la JAXA y desarrollado por una empresa de juguetería japonesa, y que tras completar su misión, fue comercializado por 150€, hito que ilustra cómo la tecnología robótica puede volverse accesible para un público más amplio, incluso después de su participación en misiones espaciales.

La educación en robótica suele basarse en este tipo de sistemas de bajo coste, ya que las instituciones educativas enfrentan limitaciones presupuestarias que dificultan la adquisición de sistemas más costosos, por su gran número de alumnos, a los que no podrían proveer de sistemas de este calibre de otra manera.

La robótica de bajo coste no solo ha democratizado el acceso a la tecnología



Figura 1.6: Sora-Q, robot lunar *low-cost* desarrollado por Takara Tomy y la JAXA.

robótica, sino que también ha revolucionado la forma en que se enseña la robótica en las escuelas. Este enfoque económico ha permitido a las instituciones educativas superar las limitaciones presupuestarias y proporcionar a un mayor número de estudiantes la oportunidad de involucrarse en actividades prácticas de robótica, allanando el camino para que los estudiantes se sumerjan en áreas más avanzadas de este área, como pueden ser la robótica móvil o campos estrechamente relacionados.

## 1.6. La multirobótica

La multirobótica es un campo de investigación que estudia y desarrolla sistemas robóticos compuestos por múltiples robots que trabajan en conjunto para realizar una variedad tareas complejas, que ha sido ampliamente estudiada en artículos como [Verma and Ranga, 2021], en los que se relatan todos los aspectos de la misma, poniendo en contexto este novedoso campo.

Estos sistemas pueden dividir sus tareas, como la exploración de entornos desconocidos o la búsqueda y rescate en áreas de difícil acceso, por lo que la colaboración entre ellos es de vital importancia, e implica aspectos como el establecimiento de comunicaciones para compartirse información y entender de un mejor modo el mundo y contexto que les rodea, la creación de mapas del entorno para poder localizarse y navegar por el mismo de manera controlada o la manipulación de objetos, muchas veces necesaria para completar el objetivo propuesto para estos sistemas. Todo ello puede verse descrito en al trabajo de [Parker, 2003], en el que se relatan con más detalle los avances logrados en estos aspectos.

La coordinación entre estos sistemas puede suponer la diferencia entre el éxito o el fracaso de su misión, por lo que también es de suma importancia, y por ello se han realizado múltiples trabajos acerca de este tema. En estos términos, los equipos de

robots pueden operar eficientemente asignando roles y responsabilidades como expone el artículo de [Alami et al., 1998], en el que se desarrolla un sistema de control con la menor centralización posible para estudiar la cooperación multirobot en el proyecto MARTHA<sup>5</sup>.

A pesar de intentar crear sistemas con la mayor descentralización posible, el trabajo anterior sigue siendo un sistema centralizado, donde un robot puede asumir roles específicos. La tendencia actual se inclina hacia sistemas directamente o casi totalmente descentralizados, donde la coordinación y la optimización son fundamentales, como es notable en el trabajo de [Sheng et al., 2006], en el que todos los robots actualizan su propio mapa local con la información de los demás, sin que ninguno de ellos adquiera un mayor protagonismo o importancia.

Gracias a este tipo de trabajos, se ha conseguido mejorar la eficiencia y la robustez de los sistemas robóticos, y la multirobótica se ha convertido en un campo importante de investigación. Los principios y problemas técnicos en este campo se exploran en diversos contextos, como se ilustra en la Figura 1.7, en la cual, la flota de robots está realizando una tarea de búsqueda y rescate, mediante la división de un área, probablemente desconocida, actualizando sus mapas como se ha explicado anteriormente.



Figura 1.7: Múltiples robots durante una operación de búsqueda y rescate.

En el ámbito educativo, la multirobótica ofrece una oportunidad única para involucrar a los estudiantes en actividades prácticas y colaborativas. Al trabajar con sistemas multirobot, los estudiantes no solo adquieren conocimientos sobre programación, control y mecánica de robots, sino que también exploran conceptos como son las telecomunicaciones entre robots, la coordinación, la planificación y asignación de tareas con o sin prioridades, la localización y navegación conjunta, como se ha explicado en los trabajos citados en la sección 1.3, así como la seguridad que se requiere para

---

<sup>5</sup>MARTHA: European ESPRIT III Project No 6668 “Multiple Autonomous Robots for Transport and Handling Applications”

evitar colisiones entre ellos, adquiriendo a su vez habilidades útiles para el trabajo en equipo. Además, la multirobótica proporciona un entorno de aprendizaje dinámico y estimulante que despierta aún más la curiosidad y la creatividad de los estudiantes, preparándolos para enfrentar los desafíos del mundo tecnológico en constante evolución.

Un ejemplo representativo de los robots educativos, en este caso del laboratorio de robótica de la URJC, aparece en la Figura 1.8, en la que se pueden diferenciar hasta dos modelos distintos de robots: Turtlebot 2 en la parte superior de la imagen, y Turtlebot 4, más modernos, en la parte inferior de la misma.



Figura 1.8: Robots educativos, modelos Turtlebot 2 (arriba) y Turtlebot 4 (abajo).

## 1.7. Flujos de datos en robótica

Las mencionadas telecomunicaciones entre robots son fundamentales en la multirobótica, ya que garantizan una comunicación rápida, optima, eficiente y ordenada entre los distintos dispositivos, necesaria para un correcto desempeño de la funcionalidad en cuestión. Sin embargo, este proceso puede enfrentarse a desafíos como la congestión de la red, y la consecuente pérdida de mensajes, que pueden ser críticos para el correcto funcionamiento de los robots. Por este motivo es crucial gestionar cuidadosamente la cantidad de robots y mensajes generados, intentando minimizarlos para optimizar el rendimiento del sistema y evitando de esta manera el problema conocido como “cuello de botella”.

Una de las empresas más importantes en este ámbito es ZettaScale Technology, que ha tomado mucha relevancia en los últimos años, ya que es responsable de una de las implementaciones de *middleware* de telecomunicaciones de DDS utilizado en

ROS2, llamado *CycloneDDS* y son los desarrolladores de un reciente protocolo de comunicaciones llamado *Zenoh*, que ya cuenta con importantes clientes como la NASA, debido a las prestaciones que este ofrece, superando en la mayoría de casos a los protocolos convecionales y que ya ha sido oficialmente seleccionado como el próximo RMW de ROS2<sup>6</sup>. Además de esto poseen un potente middleware para la programación de flujos de datos aún en desarrollo, llamado *Zenoh-Flow*, así como un puente llamado *zenoh-bridge-dds* que hace las veces de traductor entre los protocolos DDS y Zenoh, lo que permite la comunicación entre estos flujos de datos con nodos de ROS2, haciendo posible la creación de aplicaciones robóticas en forma de flujos de datos, utilizando nodos de ROS2 existentes, y fusionando de esta manera ambos campos.

Un flujo de datos consiste en un grafo dirigido de los datos que fluyen entre operaciones. Mantener un flujo de datos correcto es fundamental para solventar los problemas de telecomunicaciones mencionados en el desarrollo de sistemas de multirobótica. Además, simplifica el desarrollo del software al proporcionar una clara visión de la dirección, el origen y el destino de los datos en cada momento. Esto permite dividir el programa en partes claramente diferenciadas, normalmente llamadas nodos, modularizándolo y dando lugar a la división del problema último en varios problemas más simples y fáciles de atajar, creando un paradigma que modela el programa como un flujo de datos. Como resultado, el desarrollo se vuelve un proceso más sostenible y escalable, y por tanto, más fácil de llevar a cabo por los estudiantes.

Este proceso se puede ver representado en la Figura 1.9, donde se ilustra el flujo de datos de manera simplificada, de un robot para detectar y seguir códigos QR<sup>7</sup>, en la que se puede observar cómo los datos siguen un esquema de nodos dirigido, y en este caso unidireccional, pasando de su origen en el robot a su procesamiento en otra máquina y acabando en su posterior vuelta al robot en forma de órdenes de movimiento, pudiendo saber en todo momento en qué proceso se encuentran dichos datos.

Este proceso es equiparable a la forma de programación de un robot reactivo, ya que, como se observa en la Figura 1.10, en los flujos de datos existen tres tipos esenciales de nodos: unos en los que se originan los datos, otros donde se comutan y otros donde los datos llegan a su final; y que encuentran correspondencia en los nodos encargados de la percepción de un robot, del computo de estos datos obtenidos y de los nodos encargados de la actuación del robot, respectivamente. Puede encontrarse una exposición más detallada durante una ponencia en la conferencia ROSCON Madrid de

---

<sup>6</sup><https://discourse.ros.org/t/ros-2-alternative-middleware-report/33771>

<sup>7</sup>[https://github.com/USanz/follow\\_beacon](https://github.com/USanz/follow_beacon)

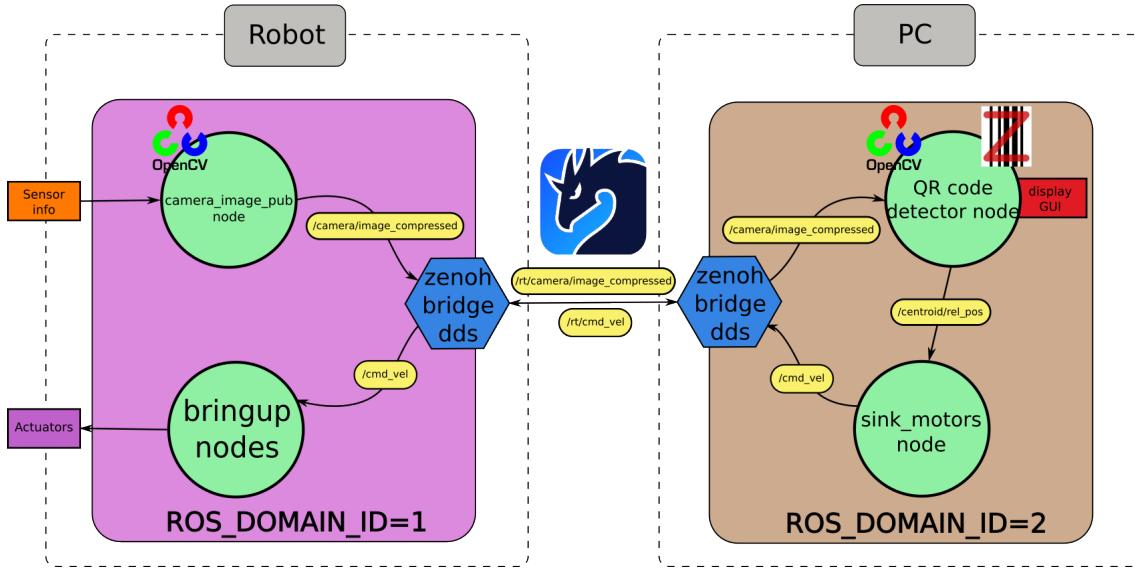


Figura 1.9: Flujo de datos de un robot para la detección y seguimiento de códigos QR.

2023<sup>8</sup>

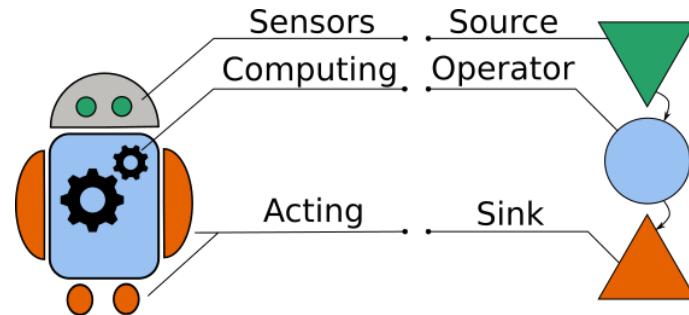


Figura 1.10: Comparación de un flujo de datos con los componentes de un robot.

Además, este tipo de programación de los robots de manera reactiva es el más simple y el primero que se suele aprender, por lo que genera una sinergia con la educación en este ámbito.

<sup>8</sup><https://www.youtube.com/watch?v=ZgFHCvEFUOI>

---

# Capítulo 2

# Objetivos

---

*Dame seis horas para talar un árbol y pasaré las primeras cuatro afilando el hacha*

Abraham Lincoln

Una vez establecido el marco contextual de este proyecto, se procederá a presentar una descripción del problema abordado, así como el proceso creativo e intelectual que ha guiado el desarrollo del mismo, que incluirá los requisitos del proyecto, la metodología empleada y el plan de trabajo detallado.

## 2.1. Descripción del problema

Este proyecto surge como respuesta a la escasa investigación sobre los flujos de datos en conjunto con ROS2, ofreciendo a su vez un entorno propicio para la creación sencilla de distintas aplicaciones robóticas basadas en dichos flujos de datos, que pueden replicar de manera versátil y sencilla, los comportamientos reativos de los robots, eludiendo la complejidad inherente de este *middleware* robótico y dando lugar, por tanto, a un nuevo entorno de programación más simple.

Además, tiene como objetivo secundario cerrar la brecha educativa entre la enseñanza secundaria y universitaria, expuesta en la Sección 1.4, en la que queda explicado el salto que existe en la educación en robótica entre la educación secundaria y la universitaria, debido a la complejidad del código y de las plataformas de desarrollo utilizadas.

## 2.2. Requisitos

Para solucionar los problemas descritos, este trabajo debe cumplir los siguientes requisitos:

1. Se utilizará *GNU/Linux*, con la distribución *Ubuntu 22.04 LTS* como sistema operativo en todos los *hardwares*.

2. Desarrollar alguna forma de programación de flujos de datos con ROS2.
3. El entorno de programación debe brindar la posibilidad de funcionar en conjunto con nodos de ROS2, permitiendo la comunicación con los mismos mediante *topics*.
4. Los *softwares* utilizados deben ser compatibles para funcionar correctamente en conjunto.
5. Las aplicaciones demostrativas desarrolladas deben ser fácilmente reproducibles y desplegables tanto en un entorno simulado como en un ambiente educativo real o de laboratorio.
6. El desarrollo del *software* debe ser lo suficientemente sencillo para poder ser llevado a cabo por alumnos preuniversitarios.
7. El *hardware* utilizado debe ser suficientemente económico para ser adquirido por organismos educativos.

Las competencias generales que se han cumplido, según la guía docente de la asignatura consisten en:

1. *CB2*. Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio. Esta competencia ha sido cumplida con la realización de la parte del software de este trabajo, en la que se han aplicado distintos conocimientos adquiridos durante el grado.
2. *CB4*. Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado. Esta competencia queda resuelta al detallar todo el complejo proceso consecuente a este trabajo de manera clara y comprensible en la memoria que esta siendo leída.
3. *CB5*. Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía. Esta competencia queda cumplida al haber adquirido los conocimientos suficientes para el desarrollo de este trabajo de manera completamente autónoma, a base de distintas pruebas y consultas en Internet.

La competencia específica *CE28* de la asignatura detalla lo siguiente: Desarrollo de las capacidades adecuadas para realizar un ejercicio original individual (o excepcionalmente colectivo), presentarlo y defenderlo ante un tribunal universitario, consistente

en un proyecto en el ámbito de las tecnologías específicas del campo de la Robótica de naturaleza profesional en el que se sinteticen e integren las competencias adquiridas en las enseñanzas. Esta última queda cumplida con la creación del presente trabajo, de la consecuente memoria y documentación y de su posterior defensa ante un tribunal.

### 2.3. Metodología

La metodología utilizada sigue pautas de investigación sobre el estado del arte previo al trabajo, y posteriormente sobre el *software* utilizado, siempre evaluando de antemano la compatibilidad con el *hardware* disponible, así como realizando pruebas pertinentes sobre su correcto funcionamiento en los distintos entornos, incluyendo la simulación y el laboratorio.

En relación con el desarrollo del *software* demostrativo se sigió un ciclo de desarrollo *software* iterativo, que consiste en la planificación del *software*, el desarrollo del mismo, su consecuente revisión mediante pruebas y su corrección, todo ello de manera periódica, generando en cada una de las iteraciones un resultado ejecutable mejor que el anterior, hasta conseguir al final una versión completamente funcional, como se ve reflejado en el esquema de la Figura 2.1. Este proceso de desarrollo puede verse alineado con los principios de mejora continua del ciclo de desarrollo PDCA.

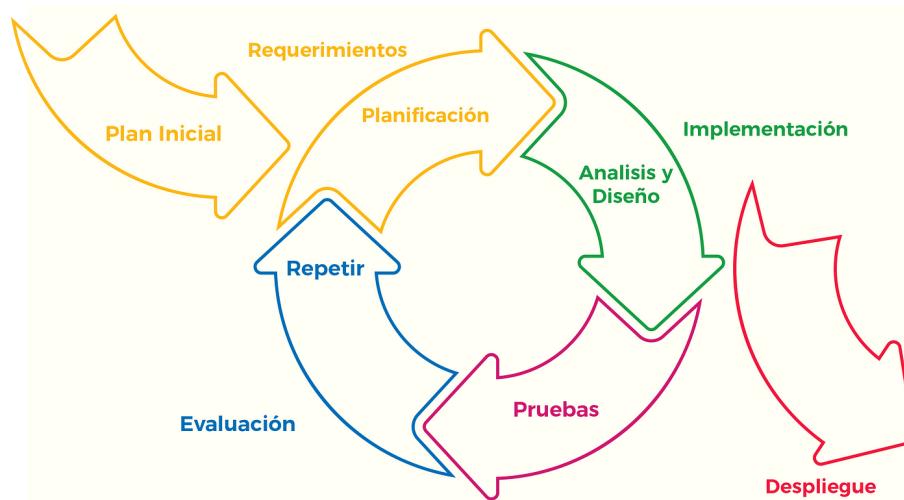


Figura 2.1: Esquema del desarrollo software iterativo.

Este desarrollo implicó pruebas periódicas en simulación con el fin de identificar errores y perfeccionar los valores de los parámetros, para posteriormente evaluar su funcionamiento en un entorno real, como el del laboratorio.

## 2.4. Plan de trabajo

El desarrollo del proyecto ha comprendido varias etapas, incluyendo la investigación del *software* a utilizar, la investigación del estado del arte, la implementación de una arquitectura *software* funcional en los distintos entornos y el desarrollo del *software* demostrativo o de ejemplo, comprendiendo un periodo de tiempo superior a un año, comenzando en Febrero de 2023 y finalizando en Mayo de 2024.

1. *Investigación del software a utilizar.* Periodo de febrero a mayo de 2023 durante las prácticas de empresa, en las que estuve aprendiendo el funcionamiento de softwares como Zenoh, Zenoh-Flow, Zenoh-bridge-DDS, CycloneDDS, y acerca de las telecomunicaciones entre robots, 35 horas semanales durante 4 meses.
2. *Investigación del estado del arte.* Periodo de junio a agosto de 2023, en el que se investigó acerca de los trabajos previos relacionados, y sobre la viabilidad y compatibilidad del proyecto.
3. *Implementación de una arquitectura software funcional en simulación.* Periodo de junio a agosto de 2023, en el que se consiguió un correcto funcionamiento del *software* en simulación.
4. *Desarrollo de software demostrativo.* Periodo de agosto a noviembre de 2023 en el que se migró el *software* desarrollado durante las prácticas a versiones posteriores.
5. *Implementación de una arquitectura software funcional en un entorno real.* Periodo de noviembre de 2023 a enero de 2024 en el que se consiguió un correcto funcionamiento del *software* en el laboratorio.
6. *Pruebas del software desarrollado en el laboratorio.* Periodo de noviembre de 2023 a abril de 2024 en el que se realizaron las pruebas y cambios necesarios para un correcto funcionamiento del *software* demostrativo en el entorno real del laboratorio.

Durante los periodos de desarrollo de este proyecto fuera de las prácticas de empresa, se dedicaban aproximadamente de 30 a 40 horas semanales entre las que se incluyen reuniones con el tutor, que generalmente se llevaban a cabo semanalmente aunque ocasionalmente cada dos semanas. Este proceso continuo dió lugar a más de un año de esfuerzo.

El proceso de trabajo ha sido realizado mediante contribuciones a un repositorio en GitHub<sup>1</sup>, así como su posterior explicación y desarrollo en el apartado de la Wiki<sup>2</sup> del mismo, haciendo las veces de bitácora, donde quedan reflejados todos los contratiempos, soluciones y pruebas realizados.

---

<sup>1</sup><https://github.com/RoboticsURJC/tfg-unai>

<sup>2</sup><https://github.com/RoboticsURJC/tfg-unai/wiki>

---

## **Capítulo 3**

# **Plataforma de desarrollo**

---

*Quizás algún fragmento de libro inspirador...*

*Autor, Título*

Escribe aquí un párrafo explicando brevemente lo que vas a contar en este capítulo. En este capítulo, explica qué has usado a nivel hardware y software para poder desarrollar tu trabajo: librerías, sistemas operativos, plataformas, entornos de desarrollo, etc.

---

# Capítulo 4

## Diseño

---

*Quizás algún fragmento de libro inspirador...*

Autor, Título

Escribe aquí un párrafo explicando brevemente lo que vas a contar en este capítulo. En este capítulo (y quizás alguno más) es donde, por fin, describes detalladamente qué has hecho y qué experimentos has llevado a cabo para validar tus desarrollos.

### 4.1. Snippets

Puede resultar interesante, para clarificar la descripción, mostrar fragmentos de código (o *snippets*) ilustrativos. En el Código 4.1 vemos un ejemplo escrito en C++.

---

```
void Memory::hypothesizeParallelograms () {
    for(it1 = this->controller->segmentMemory.begin(); it1++) {
        squareFound = false; it2 = it1; it2++;
        while ((it2 != this->controller->segmentMemory.end()) && (!squareFound))
            {
                if (geometry::haveACommonVertex((*it1), (*it2), &square)) {
                    dist1 = geometry::distanceBetweenPoints3D ((*it1).start, (*it1).end);
                    dist2 = geometry::distanceBetweenPoints3D ((*it2).start, (*it2).end);
                }
            // [...]
```

---

Código 4.1: Función para buscar elementos 3D en la imagen

En el Código 4.2 vemos un ejemplo escrito en Python.

### 4.2. Verbatim

Para mencionar identificadores usados en el código —como nombres de funciones o variables— en el texto, usa el entorno literal o verbatim `hypothesizeParallelograms()`. También se puede usar este entorno para varias líneas, como se ve a continuación:

---

```

def mostrarValores():
    print (w1.get(), w2.get())

master = Tk()
w1 = Scale(master, from_=0, to=42)
w1.pack()
w2 = Scale(master, from_=0, to=200, orient=HORIZONTAL)
w2.pack()
Button(master, text='Show', command=mostrarValores).pack()

mainloop()

```

---

Código 4.2: Cómo usar un Slider

```

void Memory::hypothesizeParallelograms () {
    // add your code here
}

```

### 4.3. Ecuaciones

Si necesitas insertar alguna ecuación, puedes hacerlo. Al igual que las figuras, no te olvides de referenciarlas. A continuación se exponen algunas ecuaciones de ejemplo: Ecuación 4.1 y Ecuación 4.2.

$$H = 1 - \frac{\sum_{i=0}^N \frac{(\frac{d_{js} + d_{je}}{2})}{N}}{M} \quad (4.1)$$

Ecuación 4.1: Ejemplo de ecuación con fracciones

$$v(\text{entrada}) = \begin{cases} 0 & \text{if } \epsilon_t < 0,1 \\ K_p \cdot (T_t - T) & \text{if } 0,1 \leq \epsilon_t < M_t \\ K_p \cdot M_t & \text{if } M_t < \epsilon_t \end{cases} \quad (4.2)$$

Ecuación 4.2: Ejemplo de ecuación con array y letras y símbolos especiales

### 4.4. Tablas o cuadros

Si necesitas insertar una tabla, hazlo dignamente usando las propias tablas de L<sup>A</sup>T<sub>E</sub>X, no usando pantallazos e insertándolas como figuras... En el Cuadro 4.1 vemos un ejemplo.

Parámetros	Valores
Tipo de sensor	Sony IMX219PQ[7] CMOS 8-Mpx
Tamaño del sensor	3.674 x 2.760 mm (1/4"format)
Número de pixels	3280 x 2464 (active pixels)
Tamaño de pixel	1.12 x 1.12 um
Lente	f=3.04 mm, f/2.0
Ángulo de visión	62.2 x 48.8 degrees
Lente SLR equivalente	29 mm

Cuadro 4.1: Parámetros intrínsecos de la cámara

---

# **Capítulo 5**

# **Conclusiones**

---

*Quizás algún fragmento de libro inspirador...*

*Autor, Título*

Escribe aquí un párrafo explicando brevemente lo que vas a contar en este capítulo, que básicamente será una recapitulación de los problemas que has abordado, las soluciones que has prouesto, así como los experimentos llevados a cabo para validarlos. Y con esto, cierras la memoria.

## **5.1. Conclusiones**

Enumera los objetivos y cómo los has cumplido.

Enumera también los requisitos implícitos en la consecución de esos objetivos, y cómo se han satisfecho.

No olvides dedicar un par de párrafos para hacer un balance global de qué has conseguido, y por qué es un avance respecto a lo que tenías inicialmente. Haz mención expresa de alguna limitación o peculiaridad de tu sistema y por qué es así. Y también, qué has aprendido desarrollando este trabajo.

Por último, añade otro par de párrafos de líneas futuras; esto es, cómo se puede continuar tu trabajo para abarcar una solución más amplia, o qué otras ramas de la investigación podrían seguirse partiendo de este trabajo, o cómo se podría mejorar para conseguir una aplicación real de este desarrollo (si es que no se ha llegado a conseguir).

## 5.2. Corrector ortográfico

Una vez tengas todo, no olvides pasar el corrector ortográfico de L<sup>A</sup>T<sub>E</sub>Xa todos tus ficheros *.tex*. En Windows, el propio editor TeXworks incluye el corrector. En Linux, usa aspell ejecutando el siguiente comando en tu terminal:

```
aspell --lang=es --mode=tex check capitulo1.tex
```

# Bibliografía

---

- [Alami et al., 1998] Alami, R., Fleury, S., Herrb, M., Ingrand, F., and Robert, F. (1998). Multi-robot cooperation in the martha project. *IEEE Robotics Automation Magazine*, 5(1):36–47.
- [Arai and Parker, 2003] Arai, T. and Parker, L. (2003). Editorial: Advances in multi-robot systems.
- [Chaimowicz et al., 2001] Chaimowicz, L., Sugar, T., Kumar, V., and Campos, M. (2001). An architecture for tightly coupled multi-robot cooperation. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 3, pages 2992–2997 vol.3.
- [Fox et al., 2000] Fox, D., Burgard, W., Kruppa, H., and Thrun, S. (2000). A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344.
- [Parker, 2003] Parker, L. E. (2003). Current research in multirobot systems. *Artificial Life and Robotics*, 7(1):1–5.
- [Sheng et al., 2006] Sheng, W., Yang, Q., Tan, J., and Xi, N. (2006). Distributed multi-robot coordination in area exploration. *Robotics and Autonomous Systems*, 54(12):945–955.
- [Trawny et al., 2009] Trawny, N., Roumeliotis, S. I., and Giannakis, G. B. (2009). Cooperative multi-robot localization under communication constraints. In *2009 IEEE International Conference on Robotics and Automation*, pages 4394–4400.
- [Vega, 2008] Vega, J. (2008). Navegación y autolocalización de un robot guía de visitantes. Master thesis on computer science, Rey Juan Carlos University.
- [Vega, 2015] Vega, J. (2015). De la tiza al robot. Technical report.

- [Vega, 2018a] Vega, J. (2018a). *Educational framework using robots with vision for constructivist teaching Robotics to pre-university students*. Doctoral thesis on computer science and artificial intelligence, University of Alicante.
- [Vega, 2018b] Vega, J. (2018b). JdeRobot-Kids framework for teaching robotics and vision algorithms. In *II jornada de investigación doctoral*. University of Alicante.
- [Vega, 2019] Vega, J. (2019). El profesor Julio Vega, finalista del concurso 'Ciencia en Acción 2019'. URJC, on-line newspaper interview.
- [Vega and Cañas, 2019] Vega, J. and Cañas, J. (2019). PyBoKids: An innovative python-based educational framework using real and simulated Arduino robots. *Electronics*, 8:899–915.
- [Vega et al., 2012] Vega, J., Perdices, E., and Cañas, J. (2012). *Attentive visual memory for robot localization*, pages 408–438. IGI Global, USA. Text not available. This book is protected by copyright.
- [Verma and Ranga, 2021] Verma, J. K. and Ranga, V. (2021). Multi-robot coordination analysis, taxonomy, challenges and future scope. *Journal of Intelligent & Robotic Systems*, 102(1):10.