



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 licenciante.
- *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 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 y debates constructivos, 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 xxxxxx 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

ROS2 *Robotic Operating System (ROS) 2*

RMW *ROS Middleware Interface*

DDS *Data Distribution Service*

STEM *Ciencia, Tecnología, Ingeniería y Matemáticas*

Índice general

Índice de figuras

Listado de códigos

Listado de ecuaciones

Índice de cuadros

Capítulo 1

Introducción

La creatividad es la inteligencia divirtiéndose

Albert Einstein

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, interactuando con los mismos de manera que reciben información mediante sensores y aplican cambios en el medio gracias a sus actuadores, todo ello manera inteligente debido al software integrado y a su capacidad de cómputo. Desde sus inicios, ha sido impulsada por avances significativos en tecnología, incluyendo sensores y actuadores cada vez más sofisticados, poder de procesamiento mejorado y algoritmos de control más eficientes. Estos avances han permitido la aplicación de robots móviles en una amplia gama de áreas, 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 vida de las personas. En este contexto, la investigación en robótica móvil se centra en desarrollar sistemas autónomos capaces de navegar de manera segura y eficiente en entornos desconocidos, adaptarse a cambios imprevistos y realizar tareas complejas de manera autónoma. Un buen ejemplo de este tipo de robots se puede ver en la Figura ??, en la que aparecen dos robots enviados a Marte por la NASA en 2020.

La robótica de bajo coste se refiere al desarrollo y la implementación de sistemas robóticos como los descritos anteriormente utilizando componentes y recursos económicos, con el objetivo de hacer la tecnología robótica más accesible y asequible para una amplia gama de aplicaciones y cualquier tipo de usuario. Este enfoque busca reducir los costos asociados con la construcción y operación de robots, utilizando 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

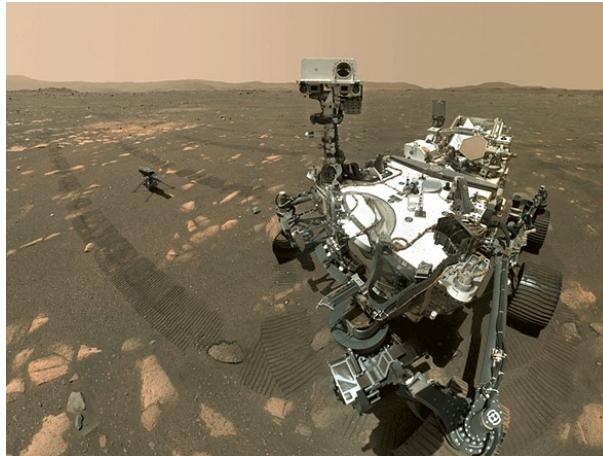


Figura 1.1: Rover Perseverance en Marte con helicóptero Ingenuity de la NASA.

viabiles para aplicaciones en las que el presupuesto es limitado o se requiere un despliegue a gran escala. Estos robots pueden desempeñar 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.



Figura 1.2: Sora-Q, robot de bajo coste enviado a la Luna, por la JAXA y comercializado por 150€.

En España, como en el resto de la comunidad europea, el enfoque educativo ha visto cada vez más claro la relevancia de incluir la robótica en los programas educativos, ya que es un instrumento valioso para promover la creatividad, el pensamiento crítico y las habilidades tecnológicas de los alumnos. En las escuelas primarias y secundarias, se ha visto un incremento en los últimos años en la implementación de programas educativos que incluyen actividades prácticas de robótica. Los estudiantes no solo

aprenden conceptos básicos de programación y robótica, sino que también aprenden a trabajar en equipo y a resolver problemas complejos. Además, la robótica educativa se ha convertido en un recurso valioso para aumentar la participación y el interés de los estudiantes en los campos STEM, preparándolos para futuras carreras relacionadas con la tecnología.

Figura 1.3: Arduino.

Dicha educación robótica suele basarse en la robótica móvil, esto se debe al hecho de que ver el resultado en movimiento genera en los estudiantes una sensación de motivación, ilusión por aprender y una realización satisfactoria, lo cual ayuda en gran medida al su aprendizaje. También suele basarse en robótica de bajo coste, debido a la baja capacidad de adquisición de los centros, por su gran número de alumnos, a los que no podrían proveer de sistemas de este calibre de otra manera. Comúnmente se utilizan placas como Arduino o similares, que resultan ideales para propósitos educativos, sin embargo, estas plataformas también imponen ciertas limitaciones en la capacidad del robot y en la posibilidad de añadir hardware periférico más complejo, y por tanto en la propia creatividad y aprendizaje de los estudiantes, restringiendo así su capacidad y potencial de creación. A medida que sus conocimientos avanzan, muchos estudiantes se enfrentan al desafío de dar el siguiente paso: la programación de robots con ROS, la plataforma estándar por excelencia, que presenta un considerable escalón de aprendizaje, ya que para su correcto uso no solo se debe aprender a programar en un lenguaje complejo, sino que también se debe aprender el entorno que rodea a este *middleware* robótico, conformado por las telecomunicaciones, la arquitectura software, la programación modular u orientada a objetos, algoritmos y estructuras de datos, y un largo etcétera, que suele conllevar decenas de asignaturas individuales en cualquier grado de robótica en una universidad. Por lo tanto, resulta evidente la necesidad de un paso intermedio que pueda servir como puente entre estos dos niveles de aprendizaje, como podría ser el caso del bachillerato tecnológico en centros de educación secundaria. Este nivel intermedio facilitaría la transición entre las habilidades adquiridas en la educación secundaria y los requisitos más avanzados de la universidad en el campo de la robótica.

La multirobotica es un campo de investigación y desarrollo que estudia y desarrolla sistemas robóticos compuestos por múltiples robots que trabajan juntos para realizar tareas complejas. Estos sistemas pueden realizar una variedad de tareas, incluida

la exploración de entornos desconocidos y la búsqueda y rescate en áreas de difícil acceso. Al aprovechar la capacidad de múltiples robots para trabajar en conjunto y complementarse entre sí, la multirobotica tiene como objetivo mejorar la eficiencia y la robustez de los sistemas robóticos. En esta disciplina se analizan los principios fundamentales, los problemas técnicos y las aplicaciones prácticas de la multirobotica en varios contextos.

En el ámbito educativo, la multirobotica 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, la localización y navegación conjunta y así como la seguridad que se requiere para evitar colisiones entre ellos. Además, la multirobotica 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.

Las mencionadas telecomunicaciones entre robots son fundamentales en la multirobotica, ya que garantizan una comunicación rápida, óptima, eficiente y ordenada entre los distintos dispositivos, permitiendo de esta manera 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, minimizándolos para optimizar el rendimiento del sistema y evitando de esta manera un cuello de botella.

Mantener un flujo de datos correcto es fundamental para solventar los problemas de telecomunicaciones mencionados en el desarrollo de sistemas de multirobotica. 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. Como resultado, el desarrollo se vuelve un proceso más sostenible y escalable, y por tanto, más fácil de llevar a cabo por los estudiantes.

1.1. Problemas de ROS en relación con el aprendizaje

1.2. Segunda sección

No olvides incluir imágenes y referenciarlas, como la Figura ??.

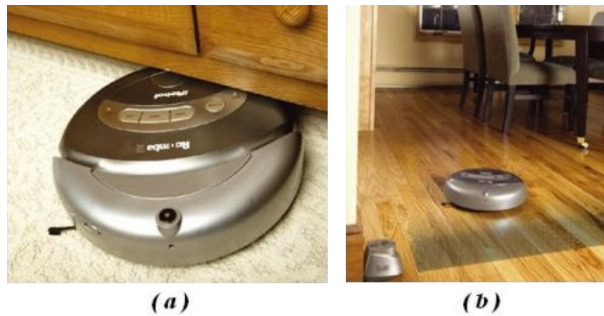


Figura 1.4: Robot aspirador Roomba de iRobot.

Ni tampoco olvides de poner las URLs como notas al pie. Por ejemplo, si hablo de la Robocup¹.

1.2.1. Números

En lugar de tener secciones interminables, como la Sección ??, divídelas en subsecciones.

Para hablar de números, mételes en el entorno *math* de L^AT_EX, por ejemplo, $1,5Kg$. También puedes usar el símbolo del Euro como aquí: 1.500€.

1.2.2. Listas

Cuando describas una colección, usa `itemize` para ítems o `enumerate` para enumerados. Por ejemplo:

- *Entorno de simulación.* Hemos usado dos entornos de simulación: uno en 3D y otro en 2D.
- *Entornos reales.* Dentro del campus, hemos realizado experimentos en Biblioteca y en el edificio de Gestión.

¹<http://www.robocup.org>

1. Primer elemento de la colección.
2. Segundo elemento de la colección.

Referencias bibliográficas Cita, sobre todo en este capítulo, referencias bibliográficas que respalden tu argumento. Para citarlas basta con poner la instrucción `\cite` con el identificador de la cita. Por ejemplo: libros como [?], artículos como [?], URLs como [?], tesis como [?], congresos como [?], u otros trabajos fin de grado como [?].

Las referencias, con todo su contenido, están recogidas en el fichero `bibliografia.bib`. El contenido de estas referencias está en formato `BibTex`. Este formato se puede obtener en muchas ocasiones directamente, desde plataformas como `Google Scholar` u otros repositorios de recursos científicos.

Existen numerosos estilos para reflejar una referencia bibliográfica. El estilo establecido por defecto en este documento es `APA`, que es uno de los estilos más comunes, pero lo puedes modificar en el archivo `memoria.tex`; concretamente, cambiando el campo `apalike` a otro en la instrucción `\bibliographystyle{apalike}`.

Y, para terminar este capítulo, resume brevemente qué vas a contar en los siguientes.

Capítulo 2

Objetivos

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 lo ideal es explicar cuáles han sido los objetivos que te has fijado conseguir con tu trabajo, qué requisitos ha de respetar el resultado final, y cómo lo has llevado a cabo; esto es, cuál ha sido tu plan de trabajo.

2.1. Descripción del problema

Cuenta aquí el objetivo u objetivos generales y, a continuación, concrétales mediante objetivos específicos.

2.2. Requisitos

Describe los requisitos que ha de cumplir tu trabajo.

2.3. Metodología

Qué paradigma de desarrollo software has seguido para alcanzar tus objetivos.

2.4. Plan de trabajo

Qué agenda has seguido. Si has ido manteniendo reuniones semanales, cumplimentando objetivos parciales, si has ido afinando poco a poco un producto final completo, etc.

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 ?? 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 ?? 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

```
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

`hypothesizeParallelograms()`. También se puede usar este entorno para varias líneas, como se ve a continuación:

```
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 ?? y Ecuación ??.

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

Ecuación 4.1: Ejemplo de ecuación con fracciones

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

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

4.4. Tablas o cuadros

Si necesitas insertar una tabla, hazlo digneamente usando las propias tablas de \LaTeX , no usando pantallazos e insertándolas como figuras... En el Cuadro ?? 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 μm
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^AT_EXa 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
```