



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*

Índice general

1. Introducción	1
1.1. Problemas de ROS en relación con el aprendizaje	1
1.2. Segunda sección	2
1.2.1. Números	3
1.2.2. Listas	3
2. Objetivos	5
2.1. Descripción del problema	5
2.2. Requisitos	5
2.3. Metodología	5
2.4. Plan de trabajo	5
3. Plataforma de desarrollo	6
4. Diseño	7
4.1. Snippets	7
4.2. Verbatim	7
4.3. Ecuaciones	8
4.4. Tablas o cuadros	8
5. Conclusiones	10
5.1. Conclusiones	10
5.2. Corrector ortográfico	11

Índice de figuras

1.1. Robot aspirador Roomba de iRobot.	2
--	---

Listado de códigos

4.1. Función para buscar elementos 3D en la imagen	7
4.2. Cómo usar un Slider	8

Listado de ecuaciones

4.1. Ejemplo de ecuación con fracciones	8
4.2. Ejemplo de ecuación con array y letras y símbolos especiales	8

Índice de cuadros

4.1. Parámetros intrínsecos de la cámara	9
--	---

Capítulo 1

Introducción

La creatividad es la inteligencia divirtiéndose

Albert Einstein

La educación en robótica se basa en la robótica de bajo coste, normalmente en placas como Arduino o similares, las cuales son ideales para este uso, y a su vez limitan la capacidad del robot en cuestión y el hardware que se puede usar y consecuentemente limitan la creatividad y el aprendizaje de los niños. Una vez se llega a un cierto nivel de conocimientos, el siguiente paso suele ser la programación de robots con ROS2, donde existe un gran escalón de aprendizaje. Este trabajo busca simplificar el desarrollo del software en ROS2 y así reducir dicho escalón, para hacer más fácil este desarrollo, dando la posibilidad de crear aplicaciones robóticas más complejas para robots más completos y que permanecen dentro de la categoría de robots de bajo coste y por tanto siguen siendo asequibles para instituciones como colegios o institutos.

1.1. Problemas de ROS en relación con el aprendizaje

ROS es el estándar en robótica para la programación de robots, pero tiene un problema y es el gran escalón de aprendizaje que existe cuando se pasa de una placa simple como arduino, a robots más complejos con máquinas integradas como las placas Raspberry Pi o un portátil directamente. Esto conlleva a una gran diferencia entre la robótica que se enseña en los colegios e institutos a la que se enseña en universidades, y es debido precisamente a la complejidad de código y enseñanza de ROS, para los cuales, se requiere incluso de varias asignaturas. Por eso en este trabajo se pretende incorporar un paso intermedio en este gran escalón.

La propia naturaleza de este *middleware* robótico nos obliga a programar nodos

que se ejecutan iterativamente en bucle, sin necesidad de generar una topología de red concreta para saber de donde vienen o a donde van los datos, lo que puede ser un poco complicado de entender a primera vista para los niños.

Además de este problema, existe otro relacionado con la congestión de red: los nodos de ROS2 se comunican a través de DDS, un protocolo de comunicaciones que genera una gran cantidad de mensajes de *Discovery*, lo que conlleva consecuentemente a la generación de congestión de la red, y dificulta de esta manera la programación de aplicaciones multiroboticas, que son un posible siguiente paso en la enseñanza de la robótica, para entender las comunicaciones entre los distintos robots.

Este trabajo pretende solucionar tanto el problema del escalón de aprendizaje, suponiendo un paso intermedio en la enseñanza de la robótica, y el problema de la congestión de red generala por DDS, suponiendo una posible solución a la misma.

El problema de la congestión se soluciona usando otro protocolo llamado Zenoh, con mejores prestaciones que DDS, y la simplicidad del código de ROS2, se ha conseguido gracias al uso de un *framework* llamado Zenoh-Flow, que funciona sobre el protocolo mencionado y el cual le da nombre. Este *framework* está pensado para la programación de flujos de datos,

Al poner las dos líneas del anterior párrafo, este aparecerá separado del anterior. Si no las pongo, los párrafos aparecerán pegados. Sigue el criterio que consideres más oportuno.

1.2. Segunda sección

No olvides incluir imágenes y referenciarlas, como la Figura 1.1.

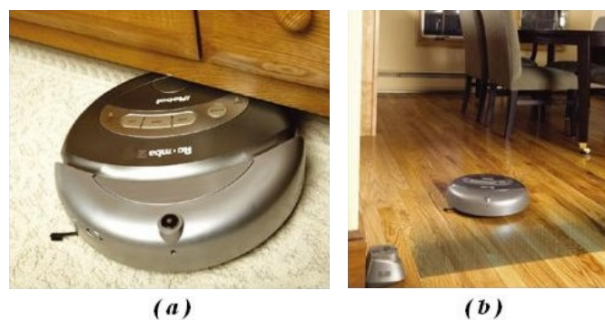


Figura 1.1: Robot aspirador Roomba de iRobot.

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 1.1, 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. 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}`.

¹<http://www.robocup.org>

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 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

```
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 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

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

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 L^AT_EX, 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 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
```

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.