

Flujo de datos en multirobótica

Unai Sanz Conejo

u.sanz.2019@alumnos.urjc.es



Universidad
Rey Juan Carlos

Trabajo fin de grado

26 de junio de 2024

1. Muchas gracias, a continuación presentaré mi trabajo de fin de grado, que como se ha explicado trata sobre la programación de flujos de datos, sobretodo orientada a entornos educativos, y como se pueden utilizar en conjunto con ROS2, que, por si no lo conocéis es el middleware de programación en robótica por excelencia.



(CC) Unai Sanz Conejo

*Este trabajo se entrega bajo licencia CC BY-NC-SA.
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.*

Contenidos

1 Introducción

2 Objetivos

3 Plataforma de desarrollo

4 Arquitectura

5 Desarrollo

6 Pruebas y experimentos

7 Conclusiones

1. La presentación esta dividida en varias partes, en las que introduciré la robótica dando un contexto general, y explicaré las bases y aspectos importantes de este proyecto, así como los experimentos y conclusiones realizadas.

Introducción

1. Comencemos con la introducción, en la que expondré brevemente el contexto en la robótica, y sobre todo en los ámbitos más relativos a este trabajo.

Robótica móvil

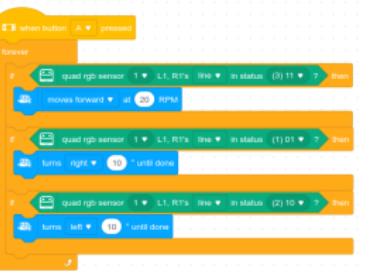
1. La robótica móvil se compone de robots capaces de moverse por el entorno que les rodea, recopilando información del mismo para tomar sus decisiones
2. Se suelen aplicar en diferentes ámbitos, como en aplicaciones de búsqueda y rescate, vigilancia, servicios y ocio (robots camareros), industria militar, etc



Robótica educativa y de bajo coste



a)



b)



c)



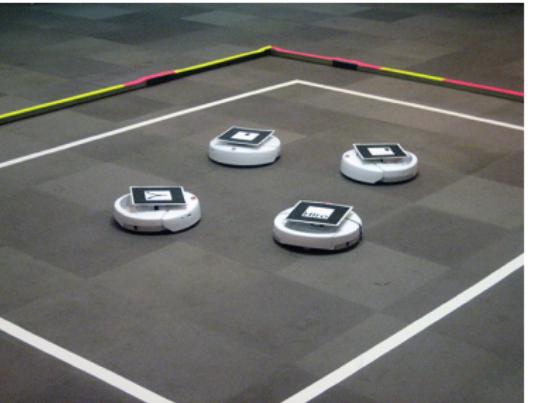
d)



e)

1. En relación con el anterior campo de la robótica, la educación cobra un papel importante dado el crecimiento de este ámbito, por lo que cada vez es más importante su enseñanza
2. Para ello la investigación se ha encargado de desarrollar la robótica de bajo coste, que se nutre de placas más económicas para entidades como institutos, y que se llevan utilizando en el itinerario formativo en España desde 2015
3. Algunos de estos ejemplos... (imágenes)

Robótica colaborativa

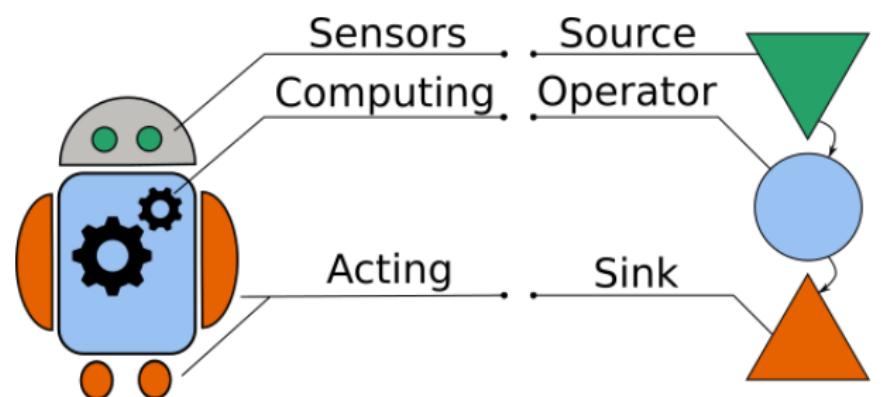


1. Debido en gran medida a este crecimiento de la robótica con ello ha surgido la robótica colaborativa y la multrobótica, las cuales me gusta diferenciar en cuestión de si ayudan al humano en su tarea o por el contrario la realizan de manera autónoma.
2. En este ámbito se encuentra el presente trabajo debido a la utilización de múltiples robots para demostrar su desempeño.

Flujos de datos en robótica

Comparación Robots con Flujos de datos

- **Sensores** = nodo **Source**.
- **Cómputo** = nodo **Operator**.
- **Actuadores** = nodo **Sink**.



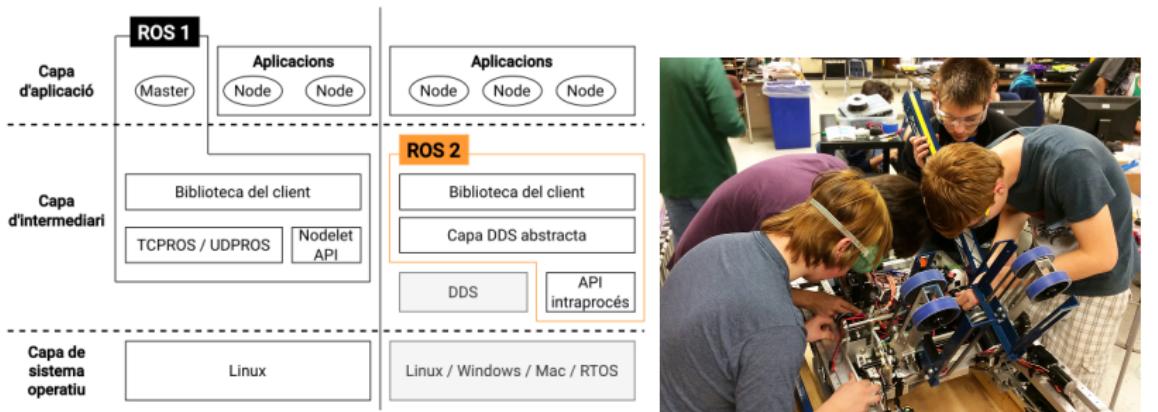
1. Los flujos de datos suelen componerse de 3 nodos, el primero en el que se originan los datos, al que llamaremos sink, el segundo donde se computan, al que llamaremos operator, y el último en el que los datos terminan su paso por el flujo, y aplicado a la robótica, muchas veces se acaban convirtiendo en decisiones o acciones tomadas por el robot.
2. Como se puede observar en la imagen, la similitud entre los propios componentes del robot y los de un flujo de datos es amplia, haciendo que su aplicación sea sencilla en este ámbito, sobretodo asemejando un comportamiento reactivo del robot, en el que se actúa en función de los datos recibidos, y que suele ser la forma más básica de programación de los robots durante su aprendizaje.
3. El hecho de aplicar los flujos de datos como forma de programación lo convierte en un paradigma de programación en sí mismo.

Objetivos

Descripción del problema

Problemas

- Escalón de aprendizaje en robótica.
 - Dificultad de [ROS](#).
- Congestion de red
 - Generada por [DDS](#).



1. Un gran problema de la robótica actualmente surge debido al envío de gran cantidad de mensajes entre los robots, especialmente en multirobótica, que en muchos casos, se ve incrementado por el envío de mensajes de Discovery con la utilización de DDS como protocolo de comunicaciones.
2. En relación con la educación, los alumnos preuniversitarios, que en su mayoría provienen de un sistema mucho más básico, de placas sencillas, entornos de programación visuales y hardware básico, encuentran un escalón de aprendizaje debido al uso de ROS y su gran dificultad, propiciado por el entorno que lo rodea, que requiere conceptos e ideas nuevas para su comprensión por parte del alumnado.
3. Es por este motivo que la solución propuesta se basa en la sencillez de entendimiento y programación del código así como en la reducción de la congestión de red, lo que además brinda un entorno de programación más simple, a la vez que compatible con el uso de nodos existentes de ROS, adquiriendo sus ventajas y paliando sus desventajas.

Requisitos

Requisitos

- Ubuntu 22.04 LTS en todas las máquinas.
- Compatibilidad entre herramientas software.
- Facilidad de reproducción y despliegue.
- Sencillez de las aplicaciones robóticas desarrolladas.
- Hardware económico.

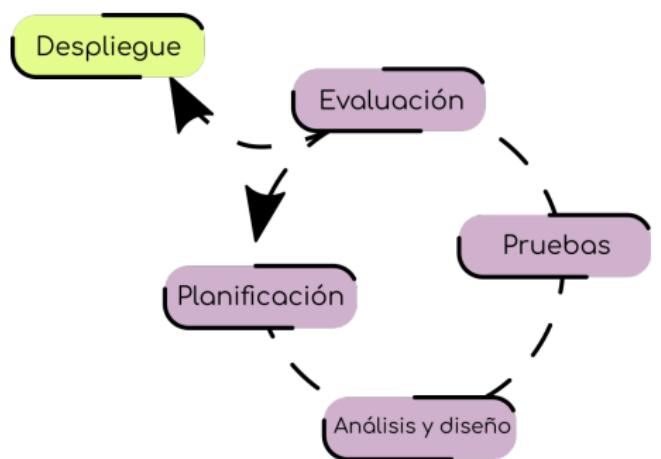
Objetivos

- Desarrollo de forma de programación de flujos de datos junto a ROS2.
- Desarrollo de una aplicación robótica que lo demuestre.
- Sencillez de código y hardware económico.
- Solución a la brecha de aprendizaje.
- Reducción de congestión.

1. De esta manera los requisitos propuestos pasan por la facilidad de reproducción y despliegue en un entorno de laboratorio, la sencillez de programación, la compatibilidad entre softwares, y un la posibilidad de aplicación sobre un hardware económico brindado por la escuela o instituto objetivo.

Metodología

- Ciclo de desarrollo software **iterativo**.
 - Actualizaciones en [YouTube](#) y [GitHub](#)
<https://github.com/RoboticsURJC/tfg-unai>.
- **Reuniones** semanales.
- Desarrollo de la **memoria**.



1. Para el cumplimiento de estos requisitos así como el desarrollo de la aplicación demostrativa, se utilizó un método iterativo de desarrollo software, que pasa por la mejora continua del programa hasta su versión final.
2. Así mismo se desarrolló una memoria, y documentación del proyecto tanto en GitHub como en YouTube, así como se mantuvieron reuniones semanales con el tutor.

Plataforma de desarrollo

Hardware



1. En cuanto a la plataforma de desarrollo hardware, se utilizaron los robots del laboratorio de robótica de la URJC (T2 y T4 Lite), así como el router allí desplegado y un portátil HP con capacidad suficiente (16GB RAM) para la ejecución de todos los nodos.
2. Además dicho portatil compone el ordenador de abordo del robot T2 utilizado, mientras que los T4 utilizan homónimamente una Raspberry Pi 4B.

Software



a)



b)



c)

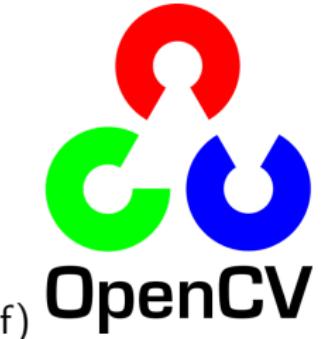


GAZEBO

d)



e)



f) OpenCV

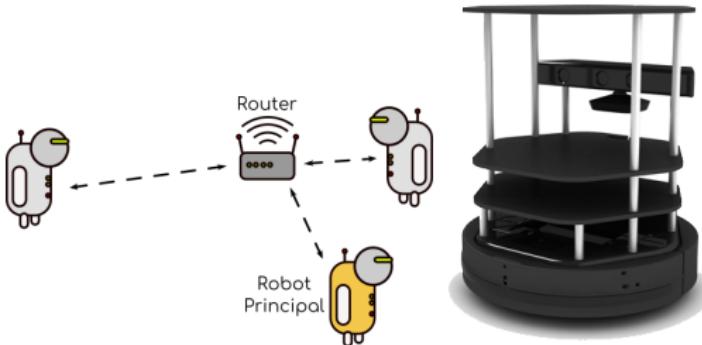
1. En cuanto al software utilizado podemos nombrar ROS2 Humble, el stack de navegación (Nav2), Zenoh y softwares relativos como Zenoh-Flow, Zenoh-bridge-DDS, además de OpenCV para la detección, Gazebo para la simulación y Python como lenguaje de programación para el desarrollo de la aplicación demostrativa.

Arquitectura

Topología hardware

Características

- Red **sin necesidad de acceso a Internet**.
- Topología **centralizada** alrededor del portátil.
- Máquina más potente que correrá todo el software.

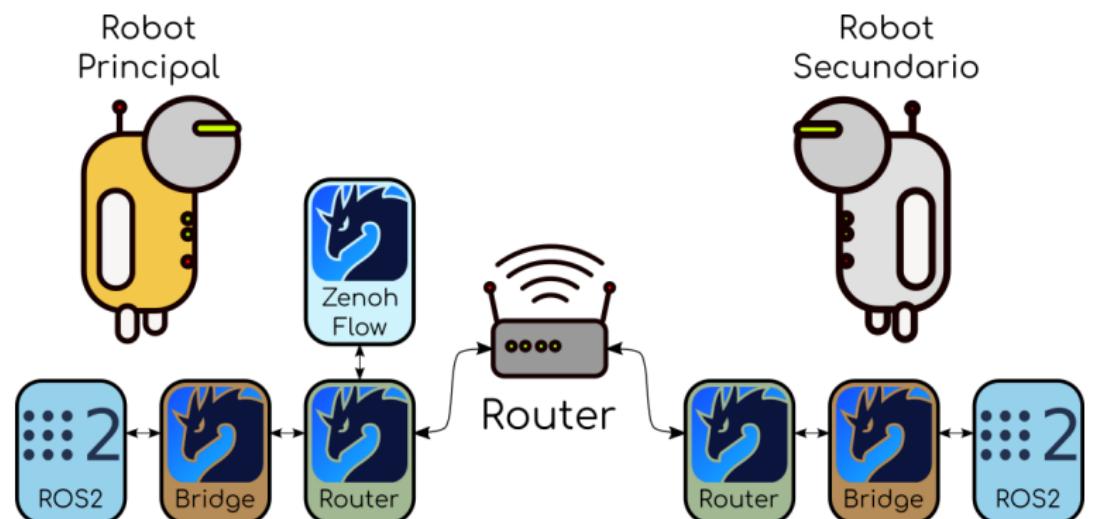


1. La distribución del hardware en la red es equitativo, estando ambos robots y ordenadores de a bordo en la misma sala, permitiendo su comunicación mediante el router como intermediario y sin necesidad de acceso a internet.
2. Además, como ya hemos explicado previamente, el portátil correrá la mayoría del software, por lo que topológicamente hablando, es el principal.

Topología software

Características

- **División** en nodos.
- **Modularización** del código.



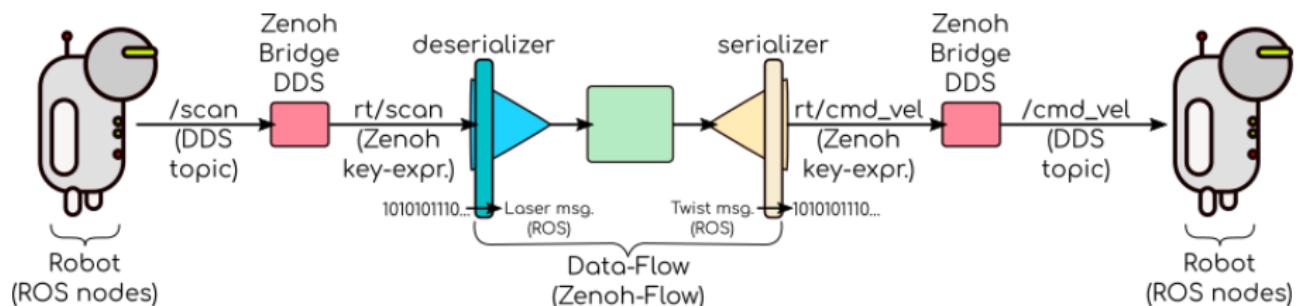
1. En cuanto al software, como se puede observar en la imagen, este trabajo se compone en su base de nodos de ROS2 (DDS), que se pueden comunicar con los nodos de Zenoh-Flow (Zenoh) mediante el Zenoh-bridge-DDS, encargado de traducir los mensajes.
2. Así mismo, se necesitan routers de zenoh corriendo en cada máquina para que estas puedan recibir los mensajes de Zenoh enviados por Zenoh-Flow desde la máquina principal y traducirlos mediante el uso del bridge a DDS, y poder así comunicarse con los nodos de ROS2.

Desarrollo

Programación de flujo de datos

Componentes software

- Zenoh (protocolo de comunicaciones).
- Zenoh-Flow (framework para programación de flujos de datos).
- Zenoh-bridge-DDS (traductor Zenoh-DDS).



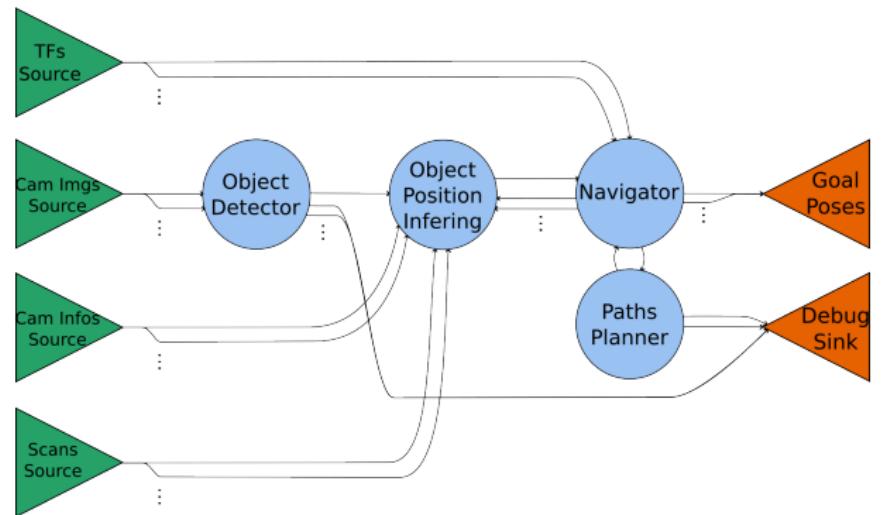
1. La siguiente imagen representa la forma de aplicar este paradigma de programación junto con ROS2, ejecutando el flujo de datos con Zenoh-Flow, y utilizando los mismos serializadores y deserializadores de ROS2 para codificar la información, para posibilitar las comunicaciones entre los robots mediante el Zenoh-bridge-DDS.

Pruebas y experimentos

Bases del proyecto

Trabajo previo

- Aplicaciones previas
- Actualizaciones.
- Mejoras y modificaciones.

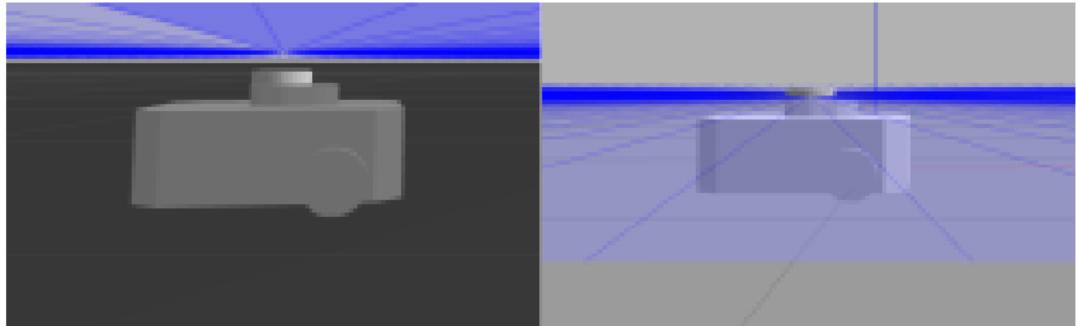


1. Con este contexto podemos visualizar un flujo de datos como una red de nodos que forman un grafo dirigido, en el que los datos fluyen entre dichos nodos.
2. En la imagen podemos ver un ejemplo de uno de ellos, concretamente de la aplicación desarrollada para este proyecto, basada en la búsqueda de un objeto en una sala por múltiples robots.
3. En concreto, el desarrollo de este proyecto comenzó durante mis prácticas de empresa en París, en la empresa ZettaScale, desarrolladora de todos los softwares relativos a Zenoh que he mencionado, incluído este.

Pruebas

Pruebas en simulación

- **Correcciones** de nodos de Zenoh-Flow.
 - nodo Navigator.
 - nodo ObjDetector.
- **Modificaciones** modelo 3D del Turtlebot 3.



1. Desde entonces y durante el desarrollo de este trabajo de fin de grado se han ido aplicando arreglos de fallos, actualizaciones, correcciones y mejoras, hasta hacer funcionar correctamente las comunicaciones de red con Zenoh como protocolo, así como la propia aplicación robótica desarrollada, tanto en simulación, como en un entorno de laboratorio con robots reales.

Pruebas

Pruebas de laboratorio

- **Telecomunicaciones**, (ROS2, CycloneDDS, Zenoh API, [Zenoh](#), [Zenoh-Flow](#), [Zenoh-bridge-DDS](#)...)
- **Modificaciones** para laboratorio
 - Mapas
 - Cambios de nombres de frames
 - Solución de bugs (nodo PathsPlanner)
- **Escalabilidad** a 3 robots.
- Pruebas con **detector de objetos** (nodo [ObjDetector](#)).

• <https://youtu.be/JNb0i57Nfjg>

1. Con la resolución de distintos fallos se consiguió también llevar la aplicación al entorno real de laboratorio, con el único problema de que no fue posible trasladar el nodo de detección debido fallos externos en el software, relativos a los nodos de ROS2 de las cámaras y a la cola de mensajes del sensor láser generada por Zenoh-Flow a la hora de comenzar el flujo de datos.

Conclusiones

1. Para acabar esta presentación, vamos a repasar lo hecho, unas breves conclusiones y las líneas futuras.

Objetivos cumplidos

- Desarrollo de forma de programación de flujos de datos junto a ROS2.
- Desarrollo de una aplicación robótica que lo demuestre.
- Sencillez de código y hardware económico.
- Solución a la brecha de aprendizaje.

Líneas futuras

- Actualización a versiones más estables.
- Paquete de instalación.
- Aplicación con interfaz sencilla e intuitiva.

Flujo de datos en multirobótica

Unai Sanz Conejo

u.sanz.2019@alumnos.urjc.es



Trabajo fin de grado

26 de junio de 2024

1. Y hasta aquí mi exposición.
2. Quedo a disposición del tribunal...