Instituto Tecnológico y de Estudios Superiores de Monterrey

TE3001B Fundamentos de Robótica(Gpo 101)

Reto semanal 1. Manchester Robotics

Autores:

Yestli Darinka Santos Sánchez // A01736992

Emiliano Olguín Ortega // A01737561

Erik García Cruz // A01732440

Profesores:

Juan Manuel Ahuactzin Larios

Rigoberto Cerino Jiménez

Alfredo García Suárez

Jueves 20 de Febrero de 2025 Semestre (6) Feb-Jun 2025

Campus Puebla

Presentación

Este documento presenta el desarrollo del Mini Challenge 1 en el contexto de ROS2. Se detalla el proceso de implementación de dos nodos: un generador de señales senoidales y un procesador de señales, junto con la visualización de los resultados en rqt_plot.

Resumen

Este reporte describe el desarrollo del Mini Challenge 1 en ROS2, donde se implementaron dos nodos: un generador de señal senoidal y un procesador de la señal. Los resultados fueron visualizados en rqt_plot y ejecutados mediante un archivo de lanzamiento.

Objetivos

General:

Implementar un sistema en ROS2 que genere y procese una señal senoidal, permitiendo su visualización en rqt_plot.

Particulares:

Desarrollar un nodo generador de señales senoidales.

Implementar un nodo que procese la señal generada.

Configurar un archivo de lanzamiento para ejecutar ambos nodos y rqt_plot simultáneamente.

Introducción

ROS2 (Robot Operating System) es un conjunto de herramientas y bibliotecas diseñadas para el desarrollo de sistemas robóticos, su arquitectura modular permite la comunicación entre diferentes componentes a través de nodos y tópicos, facilitando la integración y escalabilidad de sistemas robóticos complejos.

ROS2 se basa en un modelo de comunicación distribuida, donde los componentes principales son los nodos, que representan procesos individuales dentro del sistema, y los tópicos, que permiten el intercambio de mensajes entre nodos de manera asíncrona. Además, ROS2 utiliza otros elementos clave como servicios, que establecen comunicación sincrónica entre nodos, y acciones, que permiten la ejecución de tareas más complejas que requieren retroalimentación continua.

En este mini reto, se trabajará con dos nodos interconectados a través de tópicos. El primer nodo generará una señal senoidal, mientras que el segundo nodo la procesará y modificará

según un criterio predefinido. Ambos nodos enviarán sus resultados a rqt_plot para su análisis visual, permitiendo observar el comportamiento de la señal antes y después del procesamiento.

Solución del problema

1. Generación de Señal (Nodo generator_robotronicos)

El primer nodo se encarga de generar una señal senoidal y publicar dos valores:

- El tiempo transcurrido (time_robotronicos).
- La señal senoidal (signal_robotronicos)

Flujo del Nodo:

Se inicializa un publicador para el tiempo (publisher_t) y otro para la señal (publisher_s).

Se define un timer que se ejecuta cada 0.1 segundos para:

- Publicar el tiempo transcurrido.
- Calcular y publicar la señal senoidal (sin(t)).
- Mostrar logs con los valores generados.

Se incrementa el tiempo en 0.1s en cada iteración.

La señal generada es una función seno dependiente del tiempo.

2. Procesamiento de la Señal (Nodo process_robotronicos)

El segundo nodo recibe los valores publicados por el generador y aplica un procesamiento a la señal.

Flujo del Nodo:

- Se crean suscriptores para leer el tiempo y la señal original.
- Se define un publicador para enviar la señal procesada.
- Cuando recibe la señal:
    - Modifica la señal original con la fórmula:

self.processed_signal = np.sin((self.timer+1.0)*2.0) * 2.0

Publica la señal procesada en /proc_signal_robotronicos.

3. Lanzamiento de los Nodos

El archivo de lanzamiento en ROS 2 es una herramienta poderosa que permite iniciar múltiples nodos y configurarlos simultáneamente. En este proyecto, el archivo launch se encarga de ejecutar:

1. El nodo generador de señal (generator_robotronicos).

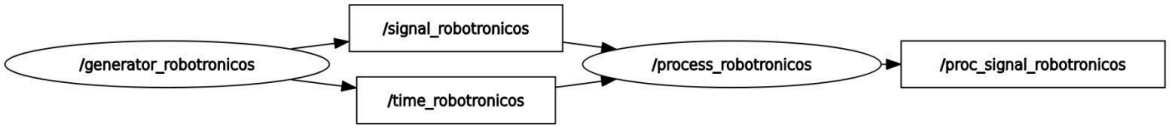
2. El nodo procesador de señal (process_robotronicos).
3. Las herramientas de visualización:
    - rqt_plot (para graficar las señales).
    - rqt_graph (para mostrar la red de nodos y tópicos).

Resultados

Los resultados obtenidos muestran la correcta generación y procesamiento de la señal. Se verificó que la modificación realizada por el nodo procesador afectaba correctamente la amplitud y/o fase de la señal original.

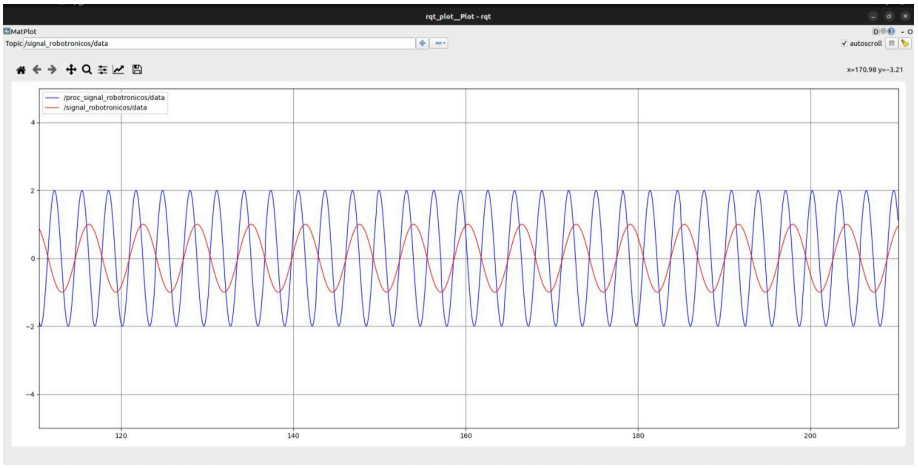
-Rqt graph:

El gráfico generado por `rqt_graph` muestra la estructura de comunicación entre los nodos y los tópicos del sistema ROS 2. En este caso, el nodo `/generator_robotronicos` es el encargado de generar tanto la señal sinusoidal como el tiempo, publicando en los tópicos `/signal_robotronicos` y `/time_robotronicos`. El nodo `/process_robotronicos` recibe estos dos tópicos, los procesa y publica la señal modificada en el tópico `/proc_signal_robotronicos`. El análisis muestra que el sistema funciona correctamente, ya que no hay conexiones rotas ni nodos desconectados, las flechas indican un flujo de datos adecuado, y todos los tópicos tienen tanto productores como consumidores definidos, lo que asegura que la comunicación entre los nodos es eficiente y estable.



-Rqt plot:

El gráfico generado por rqt_plot muestra la evolución en el tiempo de las señales publicadas en los tópicos /signal_robotronicos/data y /proc_signal_robotronicos/data. La señal original, representada en rojo, exhibe una frecuencia menor y una amplitud más controlada en comparación con la señal procesada en azul, que muestra una mayor frecuencia y variaciones más rápidas. Esto sugiere que el procesamiento aplicado ha modificado la señal, posiblemente amplificando ciertas frecuencias o filtrando componentes de baja frecuencia. El análisis indica que el resultado es completo, ya que no hay interrupciones en los datos y ambas señales mantienen una evolución coherente, lo que confirma que el sistema de adquisición y procesamiento de datos está funcionando correctamente.



La actividad se considera satisfactoria ya que cumple con los objetivos planteados.

Conclusión

En este proyecto, se lograron los objetivos planteados, ya que se implementó con éxito un sistema en ROS2 que genera y procesa una señal senoidal, permitiendo su visualización en rqt_plot. Los objetivos particulares también se cumplieron: se desarrolló un nodo generador de señales senoidales, se implementó un nodo procesador de la señal, y se configuró un archivo de lanzamiento para ejecutar ambos nodos junto con las herramientas de visualización de manera simultánea. La correcta generación y procesamiento de la señal se verificó mediante la visualización en rqt_plot y el análisis de la estructura de comunicación en rqt_graph, lo que demostró que los nodos estaban correctamente interconectados y que la comunicación entre ellos era eficiente y estable.

Los objetivos se lograron debido a una planificación adecuada y a la implementación correcta de los nodos en ROS2. El uso de herramientas como rqt_plot y rqt_graph permitió validar que la señal generada y procesada cumplía con los requisitos establecidos. Además, la documentación oficial de ROS2 y los tutoriales proporcionaron una base sólida para el desarrollo del proyecto, lo que facilitó la creación y configuración de los nodos y el archivo de lanzamiento.

Aunque los objetivos se cumplieron, una posible mejora a la metodología implementada sería incluir pruebas automatizadas para validar el funcionamiento de los nodos en diferentes escenarios, lo que permitiría detectar errores de manera más eficiente. Además, se podría

explorar la implementación de un sistema de logging más robusto para facilitar la depuración en caso de fallos. Otra mejora podría ser la integración de más nodos o tópicos para extender la funcionalidad del sistema, como la incorporación de filtros adicionales o la generación de señales más complejas. Finalmente, se podría considerar la documentación detallada de cada paso del proceso para facilitar la replicación del proyecto en el futuro.

Bibliografía

Fmrico. (s. f.). *GitHub - fmrico/book_ros2*. GitHub. https://github.com/fmrico/book_ros2

Understanding nodes — ROS 2 Documentation: Humble documentation. (s. f.). https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Nodes/Understanding-ROS2-Nodes.html

Creating a package — ROS 2 Documentation: Humble documentation. (s. f.). https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Creating-Your-First-ROS2-Package.html

Creating a workspace — ROS 2 Documentation: Humble documentation. (s. f.). https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Creating-A-Workspace/Creating-A-Workspace.html