



Tecnológico de Monterrey

Actividad R_2. Cálculo del error de un robot móvil diferencial

Mariam Landa Bautista A01736672

Emiliano Olguin Ortega A01737561

Yestli Darinka Santos Sánchez A01736992

Elias Guerra Pensado A01737354

22 de Abril del 2025

Implementación de Robótica Inteligente

Resumen

En el presente reto se desarrolló una solución basada en ROS2 para el cálculo del error en la navegación de un robot móvil diferencial (Puzzlebot). La implementación contempló la creación de nodos que permiten obtener la posición en tiempo real mediante odometría, así como calcular y publicar los errores de distancia y orientación con respecto a un punto deseado.

Objetivo general

Desarrollar un sistema en ROS2 capaz de calcular en tiempo real los errores de distancia y orientación (ed y $e\theta$) de un robot móvil diferencial con respecto a un punto objetivo.

Objetivos específicos

1. Implementar un nodo de odometría que publique la posición actual del robot móvil utilizando las velocidades de sus ruedas.
2. Desarrollar un nodo que calcule y publique los errores de distancia y orientación entre la posición actual del robot y un punto deseado.
3. Validar el correcto funcionamiento del sistema mediante pruebas prácticas con visualización del error en tiempo real.
4. Sentar las bases para la futura implementación de controladores de lazo cerrado utilizando la información del error calculado.

Introducción

Los robots móviles diferenciales son plataformas ampliamente utilizadas en entornos educativos e industriales debido a su simplicidad y capacidad de maniobra, este tipo de robots cuenta con dos ruedas motrices dispuestas en un mismo eje, lo cual permite controlar su desplazamiento mediante la diferencia de velocidades entre ambas, para lograr una navegación autónoma eficiente, es esencial conocer con precisión la posición y orientación del robot dentro del entorno, tarea que se realiza a través de la odometría.

La odometría estima la posición del robot a partir del desplazamiento angular de sus ruedas, utilizando un modelo cinemático diferencial, sin embargo, este método puede presentar errores acumulativos causados por deslizamientos, irregularidades del terreno o imprecisiones en los sensores.

Por ello, calcular el error entre la posición actual y un punto objetivo se vuelve crucial para el diseño de sistemas de navegación y control, este reporte aborda la implementación de un sistema en ROS2 que permite calcular dicho error en tiempo real, sentando las bases para

futuras aplicaciones en controladores de lazo cerrado y trayectorias inteligentes, además, se presentan evidencias de funcionamiento que respaldan la viabilidad del sistema propuesto.

Solución del problema

Para resolver el reto, se desarrollaron dos nodos en ROS2 utilizando Python. El primer nodo implementado fue el de odometría, cuya función es estimar la posición del robot en el plano a partir de la lectura de las velocidades angulares de las ruedas (encoders). Para ello, se aplicó un modelo cinemático diferencial que actualiza la posición (x , y) y orientación (θ) del robot en tiempo real, publicando estos datos mediante el mensaje "Pose2D".

El segundo nodo desarrollado fue el de cálculo de errores, el cual suscribe la posición actual del robot y un punto objetivo para calcular dos métricas fundamentales: el error de distancia (ed) y el error de orientación ($e\theta$). Se asegura además que todos los ángulos sean normalizados en el rango $[-\pi, \pi]$ para evitar errores de interpretación.

Durante las pruebas, se utilizó "teleop_twist_keyboard" para controlar manualmente al robot y observar si el sistema actualizaba correctamente su posición y los errores en tiempo real. Esta prueba permitió verificar visualmente el funcionamiento de ambos nodos y su comunicación efectiva.

Resultados

Se realizaron pruebas prácticas en las cuales el robot fue conducido manualmente utilizando comandos de teclado mientras se visualizaba la posición estimada y el cálculo del error hacia un punto objetivo. A pesar de algunas dificultades iniciales, como errores en la calibración de los encoders y la necesidad de normalizar los ángulos, el sistema logró funcionar de forma adecuada.

Las evidencias mostraron que el cálculo de los errores se actualizaba en tiempo real y respondía de forma coherente ante los desplazamientos del robot. En situaciones en las que el robot se alejaba o giraba, los errores ed y $e\theta$ respondían correctamente, demostrando que el sistema cumple con los requisitos planteados en el reto.

Conclusiones

Se logró implementar exitosamente un sistema en ROS2 capaz de estimar la posición de un robot móvil diferencial mediante odometría y calcular en tiempo real el error de distancia y orientación hacia un objetivo. Esto permitió cumplir con los objetivos propuestos de la actividad, validando la interacción entre múltiples nodos y el uso de "teleop_twist_keyboard" como herramienta de prueba.

Sin embargo, algunos objetivos no se cumplieron completamente durante las primeras pruebas debido a errores menores como la falta de normalización de ángulos o desfases en la lectura de encoders. Estas dificultades se resolvieron ajustando el código y realizando pruebas iterativas.

Una posible mejora a la metodología sería incorporar un sistema de corrección basado en sensores externos, como cámaras, para reducir el error acumulativo que genera la odometría. Además, la integración de un controlador de lazo cerrado sería el siguiente paso lógico para cerrar el ciclo de navegación autónoma.

Bibliografía

- ROS2 Documentation. (<https://docs.ros.org/en/foxy/index.html>)
- The Construct. "How to Create ROS2 Packages and Nodes." (<https://www.theconstructsim.com/>)
- Robotics Stack Exchange. "How to normalize angles in Python?" (<https://robotics.stackexchange.com/questions/17389/how-to-normalize-angles-in-python>)