



Tecnológico de Monterrey

Campus Puebla

Materia

Fundamentación de Robótica TE3001B

Tema

Challenge 1

Integrantes

José Jezarel Sánchez Mijares A01735226

Frida Lizett Zavala Pérez A01275226

Antonio Silva Martínez A01173663

Fecha

Abril 17 2023

Contenidos

| | |
|------------------------------|----------|
| Challenge 1 | 0 |
| Resumen | 2 |
| Objetivos | 2 |
| Introducción | 2 |
| Solución del problema | 3 |
| Resultados | 4 |
| Conclusiones | 4 |
| Referencias | 4 |

Resumen

Para este reto se aplicaron conocimientos previos al bloque, como lo son el uso de ROS y sus elementos de comunicación, así como nuevas herramientas como la simulación en Gazebo y el uso del Puzzle-bot de forma física.

Objetivos

El objetivo del reto consiste en generar trayectorias con el robot haciendo uso de un control de lazo abierto, que permita controlar el movimiento del robot para llegar al punto deseado.

Introducción

Este reto funcionó como un primer acercamiento a los sistemas dinámicos de los robots móviles, así como para aprender a generar trayectorias específicas para un robot diferencial a partir de la manipulación de las velocidades lineales y angulares del mismo. Este reto es muy significativo ya que el trazado de trayectorias es una técnica importante para permitir que los robots se muevan de manera autónoma y precisa, y permite al robot navegar por su entorno de manera eficiente y segura.

Para el desarrollo de este reto se requirió como paso previo conocer sobre las

herramientas que se necesitarán para comenzar a generar la solución, como primer paso conocimos Gazebo el cual es un simulador de robots de código abierto que se utiliza para modelar y simular robots en entornos virtuales. Es una herramienta muy útil para diseñar y probar algoritmos de control de robots antes de implementarlos en el mundo real. Se pueden crear modelos de robots y entornos virtuales utilizando una variedad de herramientas de modelado 3D.

Además de la interfaz gráfica, Gazebo también tiene una interfaz de línea de comandos que permite crear scripts de control y automatización para las simulaciones. Lo cual puede ser útil si se busca realizar pruebas automatizadas o realizar simulaciones a gran escala. Gazebo es una herramienta muy útil para modelar y simular robots en entornos virtuales, lo que ayuda a los desarrolladores de robótica a probar y mejorar sus algoritmos de control antes de implementarlos en el mundo real.

El robot con el que se estará trabajando es el Puzzle-bot, el cual es un robot educativo diseñado para enseñar sobre programación y robótica. El robot tiene la capacidad de moverse y girar en diferentes direcciones. Para este reto se requirió crear la conexión

entre este robot y ROS, para ello una vez instalado Ros en la computadora se requiere instalar los controladores específicos del Puzzle-bot y hacer la configuración correspondiente, para después ejecutar Ros y los nodos del Puzzle-bot y finalmente poder controlar el comportamiento y movimiento del robot.

En cuanto a la comunicación con el robot se usó el protocolo SSH (Secure Shell) el cual es una forma segura de conectarse y controlar de forma remota un dispositivo, como una computadora o un servidor, a través de una conexión encriptada. Esto significa que la información que se transmite entre los dispositivos está protegida y es difícil de interceptar y descifrar. Es una forma segura de conectarse y controlar dispositivos de forma remota mediante una conexión encriptada que protege la información que se transmite entre los dispositivos, es importante saber que para establecer una conexión SSH, se necesita un cliente SSH instalado en la computadora y un servidor SSH instalado en el dispositivo remoto.

ROS Master URI es una dirección que se utiliza para que los diferentes nodos de ROS se comuniquen entre sí. Es como una especie de "punto central" que permite que

los diferentes nodos de ROS se conecten y compartan información.

Cuando inicias una aplicación de ROS, debes establecer una dirección para el ROS Master URI, que es la dirección IP y el número de puerto del dispositivo en el que se ejecuta el nodo maestro de ROS. Los diferentes nodos pueden conectarse a este nodo maestro utilizando la dirección ROS Master URI para compartir información, como datos de sensores o comandos de control del robot.

Solución del problema

Para lograr la solución del reto, se realizaron diversas pruebas en diferentes entornos como lo son matlab, gazebo, python y ROS.

Como paso previo se hizo una primera prueba de conexión con el robot, para familiarizarnos con la comunicación y el proceso de comunicación con Ros, así como la ejecución del entorno en combinación con los nodos que generan el movimiento.

El primer elemento utilizado fue Matlab en donde nosotros realizamos las primeras pruebas para poder realizar las trayectorias que utilizaríamos para el movimiento de nuestro Puzzle-bot, en este entorno de

programación realizamos una serie de simulaciones en donde nosotros determinamos una serie de puntos por los cuales nuestro robot tendría que pasar, para esto lo que se hizo fue el utilizar el número de muestras y el tiempo total de muestreo para determinar el recorrido, en donde lo que se hizo fue dividir en diferentes períodos nuestro número de muestras y mediante la manipulación de las velocidades lineales y angulares se generó el movimiento deseado y de esta manera obtener una simulación precisa de la trayectoria que se deseaba seguir. Posteriormente, se procedió a desarrollar un código en Python para el cual usamos de base las simulaciones de Matlab desarrolladas anteriormente para generar la simulación en Gazebo y así poder modificar las simulaciones acorde a las especificaciones requeridas.

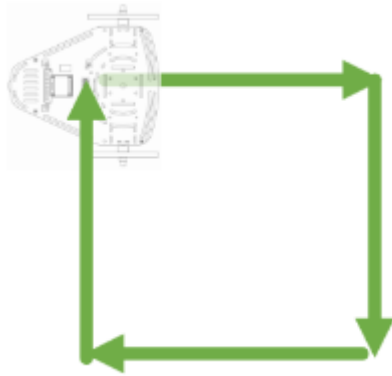
Una vez que se obtuvo la simulación en Gazebo, se transmitieron los códigos desarrollados mediante ROS a un Puzzlebot, el robot encargado de realizar las trayectorias requeridas. De esta forma, se logró que el robot siguiera las trayectorias de forma precisa y acorde a lo que se esperaba.

Resultados

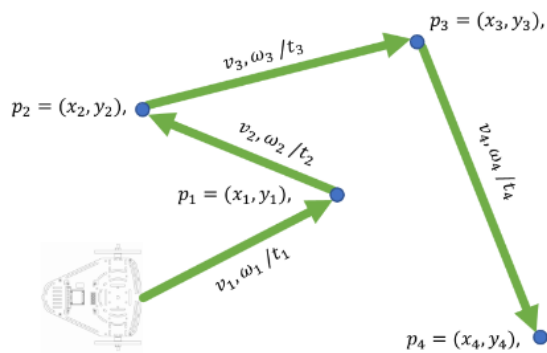
Los resultados que se obtuvieron fueron las trayectorias que se habían planteado al inicio, en donde utilizamos tanto el tiempo como las distancias y las velocidades de nuestro Puzzle-bot para completar el recorrido. Lo que se hizo fue plantearlo de dos formas diferentes, cuando se utilizó el tiempo para las trayectorias se determinó el tiempo de ejecución de una tarea y posteriormente al finalizar continuaba con el siguiente paso los cuales casi siempre el patrón consistía en avanzar y girar. Por otra parte al utilizar distancias y velocidades el comportamiento era similar pero al contrario de usar el tiempo, los ángulos a los que giraba el Puzzle-bot eran más precisos y la distancia recorrida por el robot era más exacta cuando nosotros la declaramos.

A continuación se adjuntan evidencias sobre el funcionamiento del robot, en donde la trayectoria cuadrada se maneja con tiempo y la trayectoria en zig zag se maneja con las distancias:

Cuadrado: bit.ly/40yT1R6



Zig zag: bit.ly/3Ahemni



Conclusiones

Al trabajar con diferentes aspectos del robot nos pudimos dar cuenta de cómo influye en el comportamiento de las trayectorias, siendo que al utilizar el tiempo para realizar las trayectorias nos pudimos dar cuenta que existe cierto nivel de inexactitud al realizar la acción de girar, ya que al únicamente contar con la velocidad y el tiempo tenemos que declarar cierto tiempo para que se realice la acción de girar y puede que no sea tan exacta como se desea, mientras que al utilizar el otro método nos dimos cuenta que las acciones digitado se podía hacer más precisa y eficiente ya que si

conocíamos el ángulo que necesitábamos para realizar la acción nosotros lo podríamos declarar y realizar la acción de manera más precisa.

En general, el uso de ROS para trazar trayectorias con robots ofrece muchas ventajas y permite a los desarrolladores crear robots autónomos y precisos que pueden ser utilizados en una amplia variedad de aplicaciones, este reto funciona como la base de un sinfín de aplicaciones, ya que nos ayudó a comprender la forma y el comportamiento del robot ya que al implementar un control de lazo abierto se puede observar mayormente el comportamiento natural del movimiento del robot, para posteriormente comenzar a implementar otros tipos de control.

Referencias

Willow Garage. (2013). ROS wiki: Master. Recuperado de <http://wiki.ros.org/master>

De Zúñiga, F. G. (2022). SSH: qué es y cómo funciona este protocolo. *Blog De arsys.es*.

<https://www.arsys.es/blog/ssh>

Pérez, Y. A., & Castañeda, M. A. (2017).

La educación y la robótica

educativa: Una revisión

bibliográfica. Publicaciones e

Investigación, 10(2), 21-31.

Recuperado de

[https://hemeroteca.unad.edu.co/ind](https://hemeroteca.unad.edu.co/index.php/publicaciones-e-investigacion/article/view/1593/1940)

[ex.php/publicaciones-e-investigaci](https://hemeroteca.unad.edu.co/index.php/publicaciones-e-investigacion/article/view/1593/1940)

[on/article/view/1593/1940](https://hemeroteca.unad.edu.co/index.php/publicaciones-e-investigacion/article/view/1593/1940)