



Tecnológico de Monterrey

**Instituto Tecnológico de Estudios Superiores de
Monterrey**

TE3002B.502

Implementación de Robótica Inteligente

Challenge 1.

Gpo 502

Profesor:

Rigoberto Cerino Jiménez

Alumnos:

Daniela Berenice Hernández de Vicente	A01735346
Alejandro Armenta Arellano	A01734879
Dana Marian Rivera Oropeza	A00830027

Fecha: 14 de Abril del 2023

Resumen

En este reporte se podrá observar una descripción de lo que son los custom messages, parameters y launch file, cómo funcionan, todo esto con tal de comprender más allá de simplemente resolver el challenge semanal impuesto por Manchester Robotics.

Aunado a estas descripciones y definiciones, se puede encontrar la solución paso por paso al challenge de esta segunda semana.

Objetivos

En este primer challenge se espera que los estudiantes repasen los conceptos vistos tanto en las sesiones con Manchester Robotics como las primeras dos sesiones donde se explica el uso de gazebo.

De igual manera es importante destacar que esta actividad se divide en dos partes.

Por lo cual la primera parte consiste en lo siguiente:

- Crear un nodo el cual sea capaz de manejar el robot simulado en gazebo el cual tendrá de trayectoria un cuadrado con una longitud de 2 m por lado.
- Usar el mismo nodo mencionado anteriormente para dirigir el robot físicamente con la misma trayectoria.
- Emplear un control de lazo abierto, el cual deberá ser diseñado por el alumno.
- El usuario deberá seleccionar la velocidad o el tiempo en el cual finalizará la trayectoria, ahora bien, el controlador deberá estimar las velocidades, aceleraciones o el tiempo requerido, esto depende del valor ingresado inicialmente por el usuario.
- El controlador debe tener en consideración las perturbaciones, no linealidades y el ruido que se pueda encontrar.

Por otro lado la segunda parte consiste en lo siguiente:

- Crear un nodo el cual genera trayectorias de acuerdo al usuario.
- La trayectoria debe ser definida por el usuario mediante el archivo de parámetros.
- La trayectoria debe ser definida por diferentes puntos, velocidades del robot o tiempo, esto dependerá del usuario.
- Ahora bien, para cada punto el nodo deberá estimar la velocidad lineal y rotacional cuando el tiempo sea dado por el usuario, o estimar el tiempo en caso de que la velocidad sea dada.
- El nodo debe dar a conocer al usuario si el punto es alcanzable de acuerdo con el comportamiento dinámico del robot móvil y los parámetros que el usuario ha proporcionado.
- El controlador debe tomar en consideración lo que el alumno haya definido como robusto para este caso.

Introducción

Semanalmente se presenta un reto por Manchester Robotics, los cuales serán resueltos con la ayuda de programas especializados como ROS y sus complementos como Gazebo, además de ciertos lenguajes de programación como Python o C++.

Aunado a esto cada semana es proporcionada una clase base con respecto a lo que se manejara durante el challenge semanal y ciertos conceptos fundamentales para la resolución del mismo, así como una sesión de preguntas y respuestas donde el objetivo es resolver dudas con respecto al mismo challenge semanal.

Como complemento se proporcionan algunas actividades las cuales de alguna manera llevan de la mano al alumno con el fin de lograr que este pueda desarrollar el challenge semanal con éxito.

En esta primera semana al ya contar con una noción fuerte de la arquitectura de trabajo de ROS, se nos introdujo a los siguientes temas, los cuales van más relacionados al manejo de un robot móvil (Puzzlebot):

- Gazebo:
 - Este es un software el cual se emplea para la realización de simulaciones realistas tanto de la física e interacciones de los objetos, así como de la dinámica y sensores de diferentes robots. [1]
 - Donde para ejecutar el simulador es necesario descargar los archivos que simulan el mundo deseado, y ejecutar las siguientes líneas de código.

catkin_make

source devel/setup.bash

roslaunch puzzlebot_world puzzlebot_simple_world.launch

- Puzzlebot (Robot Móvil):
 - El Puzzlebot es un robot móvil diseñado por Manchester Robotics para la educación, y se visualiza en la imagen 1.
 - El cerebro del Puzzlebot es una Hacker Board. Esta actúa como un nodo computacional robusto para cada uno de los componentes del kit de inicio y para su interfaz de comunicación para computadoras externas (NVIDIA Jetson Nano).
 - Cuenta con sensores listos para usarse, en nuestro caso cuenta con una cámara, la cual en el futuro nos ayudará a poder procesar imágenes por computadora. [2]

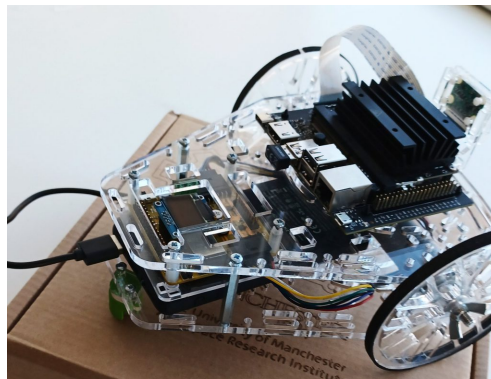


Imagen 1.

- Comunicación SSH:
 - Esta funciona mediante el uso de un modelo cliente-servidor con el fin de permitir la autenticación de dos sistemas remotos y el cifrado de los datos que pasan entre ellos. [3]
 - Por otro lado, esta permite acceder a otro ordenador a través de la red, con el fin de ejecutar comandos en la máquina remota y poder mover ficheros entre dos máquinas. [4]
- ROS Master URI:

- Es la variable que le dice a un nodo dónde puede localizarse el maestro. [5]

Solución del problema

Para la solución de la primera parte del problema se creó un solo nodo, el cual es el encargado del controlador, aunado a esto se creó un solo tópico denominado *cmd_vel*.

Mientras que para la segunda parte del problema fue necesario crear dos nodos, uno para la generación de la trayectoria y otro para el controlador, aunado a esto se crearon dos tópicos denominados *pose* y *cmd_vel*.

Cabe mencionar que por temas de practicidad los códigos de cada uno de los nodos se encontrarán específicamente en la sección de anexos donde se incluirá una liga al github donde se encuentran cada uno de ellos.

Primera parte:

Primero el nodo *Path_Generator* extrae la cadena de texto que se encuentra en el archivo de parámetros, para posteriormente enviarla al nodo controller vuelve esa cadena de texto en flotante que posteriormente se vuelve lista, separando con comas en cada espacio.

Se agrega un contador el cual cada vez que se ejecute una orden, aumenta, lo que genera que la lista de instrucciones aumente.

Para este caso en particular se cuenta con 8 instrucciones, 4 recorridos de 2 metros y 4 giros.

Segunda parte:

Primero el nodo *Path_Generator* extrae la cadena de texto que se encuentra en el archivo de parámetros, para posteriormente enviarla al nodo controller vuelve esa cadena de texto en flotante que posteriormente se vuelve lista, separando con comas en cada espacio.

Se agrega un contador el cual cada vez que se ejecute una orden, aumenta, lo que genera que la lista de instrucciones aumente.

Para esto se cuenta con un *if*, el cual indica cuando se realiza un movimiento seguido de un giro y así va aumentando.

Para este caso en particular se cuenta con 9 instrucciones, 4 recorridos y 5 giros, ya que se cuenta como uno el giro inicial de la trayectoria.

De igual manera, en el nodo controller se lee el encoder, lo que hace que conforme el robot móvil avance se genere una ecuación que a su vez actualiza la información recibida, restando de la original con el fin de conocer cuánto le falta por recorrer.

Una vez que llega al dato deseado, se ocupa otra variable para definir la rotación de las ruedas con respecto al ángulo que el usuario implementa, funcionando de la misma manera si el encoder determina que ya se ha llegado al ángulo deseado, el programa avanza a la siguiente instrucción.

Resultados

Es importante destacar que cada uno de los resultados obtenidos se grabaron y se encuentran en la sección de anexos dentro del link de github que es proporcionado en dicha sección.

Primera parte:

Como resultado de la primera parte del challenge 1, obtenemos la realización de manera correcta de la trayectoria cuadrada, donde cada lado de la misma mide 2 metros, donde dependiendo del dato ingresado es si se estima la velocidad o el tiempo en el que tarda en realizarse la trayectoria completa.

Para la realización de esta parte del challenge se incluye tanto un video donde se observa al robot realizando la trayectoria cuadrada, como un video donde se observa la misma simulación realizada en gazebo.

Segunda parte:

Como resultado de la segunda parte del challenge 1, obtenemos la realización de manera correcta de la trayectoria proporcionada por el usuario, donde se ingresan unos puntos, tiempo o velocidad para generarse, y dependiendo de lo ingresado es si se calcula el tiempo o las velocidades lineales y rotacionales.

Para la realización de esta parte del challenge se incluye tanto un video donde se observa al robot realizando la trayectoria dada por el usuario, como un video donde se observa la misma simulación realizada en gazebo.

Conclusiones

Aún cuando en algunos momentos parecía inalcanzable, se consiguió realizar con éxito cada uno de los objetivos antes mencionados.

Se logra realizar de manera correcta el recorrido de ambas trayectorias, aunque se puede observar que en ocasiones no las trayectorias no se realizan de manera uniforme, es decir que se va de lado el robot móvil, esto puede ser por dos factores, tanto por el controlador empleado, como por el tiempo de muestreo.

De igual manera se destaca que al ser el robot móvil un “ambiente” relativamente nuevo para nosotros, evidentemente nos topamos con ciertas dificultades para manejarlo, pero que con ayuda del profesor encargado del mismo fue más accesible poder buscar la solución correcta al challenge de esta semana.

Bibliografía

1. *documentacion:gazebo [RyCh]*. (2015, 10 agosto). RyCh. Recuperado 18 de abril de 2023, de <https://rych.dcc.uchile.cl/doku.php?id=documentacion:gazebo#:~:text=Gazebo%20es%20un%20software%20que,los%20paquetes%20desarrollados%20para%20ROS.>
2. *Puzzlebot – Manchester Robotics*. (s. f.). Manchester Robotics. Recuperado 18 de abril de 2023, de <https://manchester-robotics.com/puzzlebot/>
3. A, D., & A, D. (2023). ¿Cómo funciona el SSH? *Tutoriales Hostinger*. [https://www.hostinger.mx/tutoriales/que-es-ssh#:~:text=La%20forma%20en%20que%20funciona, puede%20cambiar%20si%20es%20necesario\).](https://www.hostinger.mx/tutoriales/que-es-ssh#:~:text=La%20forma%20en%20que%20funciona, puede%20cambiar%20si%20es%20necesario).)
4. *Comandos básicos de ssh*. (s. f.). Recuperado 18 de abril de 2023, de https://www3.uji.es/~galdu/ssh_vs_rsh/x165.html#:~:text=SSH%20es%20un%20programa%20que,de%20rlogin%2C%20rsh%20y%20rcp.
5. *ROS/EnvironmentVariables - ROS Wiki*. (2019, 19 septiembre). ROS. Recuperado 18 de abril de 2023, de http://wiki.ros.org/ROS/EnvironmentVariables#ROS_MASTER_URI

Anexos

Link al Github:

https://github.com/Bere901/-Retos_Manchester_Robotics_IRI/tree/main/Week%201