

Práctica 3 Robótica

1. Objetivos de la práctica

Desarrollar un código en ROS (C++ o Python) que permita a el robot TurtleBot3 (waffle o Burger) moverse a través de 10 puntos de referencia (waypoints). Para esto, el robot debe navegar por un entorno utilizando el principio de “Navegación Autónoma”, esto quiere decir, utilizando módulos de: Mapeo(*mapping*), Localización(*localization*), Planificación(*planning*) y seguimiento(*tracking*).

2. Descripción

En esta actividad, los estudiantes deberán utilizar ROS (Robot Operating System) para programar un robot TurtleBot3(waffle o Burger) en el entorno de simulación “house” provisto por el mismo paquete (*package*). El robot debe navegar autónomamente por el entorno sin colisionar.

Para esto, se debe usar los recursos y herramientas de SLAM (Simultaneous Localization and Mapping) y la simulación de navegación proporcionados por la documentación de TurtleBot3: <https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/#gazebo-simulation>.

Los estudiantes configurarán el robot para que navegue a través de 10 puntos de referencia predefinidos. Para lograr esto debe utilizar SLAM para crear el mapa del entorno, tal como lo muestra el enlace siguiente:

https://emanual.robotis.com/docs/en/platform/turtlebot3/slam_simulation/

Para el correcto funcionamiento de SLAM debe instalar paquete gmapping con el siguiente comando:

- `sudo apt-get install ros-noetic-slam-gmapping`

Posteriormente utilice las capacidades que proveen los módulos de planificación y seguimiento de caminos (node *move_base*) que se utiliza en: https://emanual.robotis.com/docs/en/platform/turtlebot3/nav_simulation/. De esta manera es posible navegar de un punto inicial a un punto final de manera autónoma. NOTA: Sintonicice los parámetros de planificación tal como aparece en: <https://emanual.robotis.com/docs/en/platform/turtlebot3/navigation/#tuning-guide> . Para el correcto funcionamiento de paquete *move_base*, debe instalar paquete gmapping con el siguiente comando:

- `sudo apt-get install ros-noetic-dwa-local-planner`

Con todo lo anterior en funcionamiento se debe desarrollar un código (nodo) el cual gestione navegar sin colisionar por el entorno “house” a través de 10 *waypoints*, tal como muestra la Fig.1. El código puede ser escrito en Python o en C++. De esta manera el “nodo” encargado de la gestión de navegación, se basa: **gestionar el avance a través de los distintos waypoints considerando un control en lazo cerrado de la posición del robot con respecto al punto donde debe llegar.**

En resumen, tenga en consideración lo siguiente:

- Cree un script en Python o C++ que implemente un gestor de waypoints.
- Defina una lista de 10 waypoints en el formato `(x, y, θ)` que el robot debe visitar.
- Use los mensajes de `geometry_msgs/PoseStamped` para enviar los objetivos de navegación al TurtleBot3.

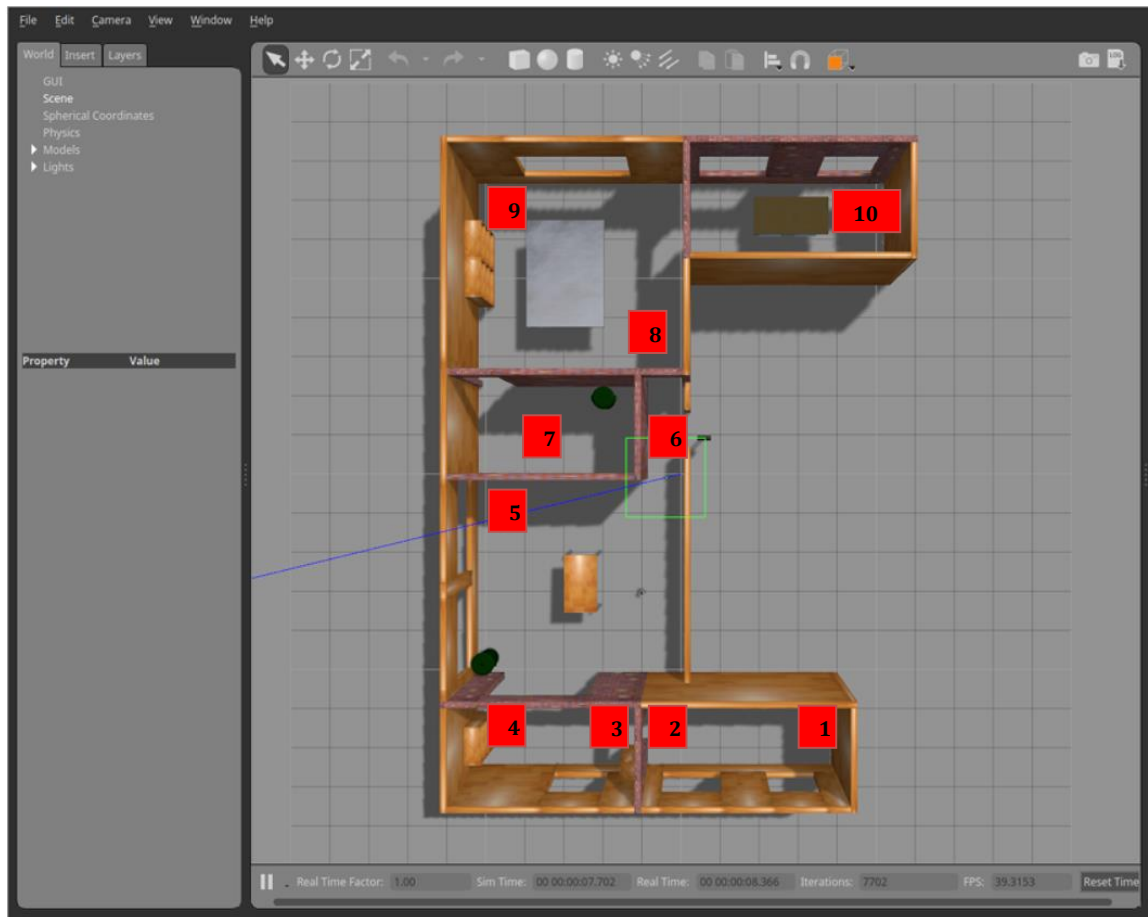


Fig. 1. Entorno de simulación con los 10 waypoint definidos

3. Herramientas a utilizar

- ROS: Sistema operativo robótico utilizado para programar el TurtleBot3.
- TurtleBot3: Robot simulado en el entorno de ROS.
- Gazebo: Simulador de robótica utilizado para probar y visualizar la navegación del TurtleBot3.
- RViz: Herramienta de visualización en ROS para monitorear la navegación y SLAM.
- SLAM Toolbox: Paquete de ROS para la localización y mapeo simultáneos.
- Navigation Stack: Paquete de ROS para la navegación autónoma (move_base).

4. Modalidad de evaluación

- **Plazo:** La actividad debe ser completada y entregada en un plazo de 2 semanas, es decir, fecha de entrega 10/07/2024, hasta las 23.59 hrs.
- **Modalidad de Entrega:** Los estudiantes deben subir su código a un repositorio en GitHub al enlace provisto en UCAMPUS. Además, deben incluir un video (screen record de toda la pantalla del computador, NO SE ACEPTA GRABACIONES CON EL TELEFONO) de la simulación mostrando al TurtleBot3 completando la ruta de los 10 waypoints. Para esto todo alumno debe realizarse una cuenta en GitHub con tal de poder subir el código.
- **IMPORTANTE:** La cuenta GitHub debe ser creada con anticipación, con tal de poder incorporarlos como usuarios en el GitHub del Grupo de Robótica (GR) y así puedan subir el código. Una vez creada la cuenta deben enviarme por UCAMPUS vuestro usuario para agregarlos al GitHub del GR. Si crean la cuenta el día de la entrega no podre agregarlos. Por último, el video debe ser subido a vuestro propio SharePoint (de la cuenta Microsoft institucional). El enlace del video debe ser guardado en un archivo .txt y subirlo también a GitHub.

- El código debe funcionar correctamente, estar claramente estructurado y con comentarios dentro de este. El código representa el 50 % de la nota.
- El video debe tener un buen enfoque. Debe mostrar a la vez Gazebo + RVIZ. En este último debe visualizarse las flechas de la localización, el camino calculado para moverse es un waypoint a otro. También en el video se debe dejar ver al menos en un momento el nombre del usuario del portátil como muestra la Fig. 2. Se recomienda utilizar el programa "SimpleScreenRecorder".

```

Oct 10 16:47
h2s@forum: ~
h2s@forum: ~ 102x14
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [327 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [612 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [427 kB]
Hit:5 https://packagecloud.io/eugeniy/tabby/ubuntu jammy InRelease
Fetched 1,691 kB in 22s (75.6 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
163 packages can be upgraded. Run 'apt list --upgradable' to see them.
h2s@forum:~$
h2s@forum: ~ 52x11
64 bytes from del11s10 (142.250.182.174): icmp_seq=2
4366 ttl=128 time=5.86 ms
64 bytes from del11s10 (142.250.182.174): icmp_seq=2
4367 ttl=128 time=12.2 ms
64 bytes from del11s10 (142.250.182.174): icmp_seq=2
4368 ttl=128 time=17.6 ms
64 bytes from del11s10 (142.250.182.174): icmp_seq=2
4369 ttl=128 time=14.4 ms
64 bytes from del11s10 (142.250.182.174): icmp_seq=2
4370 ttl=128 time=17.1 ms
h2s@forum: ~ 47x11
h2s@forum:~$ cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04 (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu"

```

Fig 2. Imagen de terminal con nombre del usuario del computador.