
ROBOTICS LAB SIMULATION PROJECTS

Concept used:

1. SLAM

SLAM (simultaneous localization and mapping) is a method used for autonomous vehicles that lets you build a map and localize your vehicle in that map at the same time. SLAM algorithms allow the vehicle to map out unknown environments.

Consider a home robot vacuum. Without SLAM, it will just move randomly within a room and may not be able to clean the entire floor surface. In addition, this approach uses excessive power, so the battery will run out more quickly. On the other hand, robots with SLAM can use information such as the number of wheel revolutions and data from cameras and other imaging sensors to determine the amount of movement needed. This is called localization. The robot can also simultaneously use the camera and other sensors to create a map of the obstacles in its surroundings and avoid cleaning the same area twice. This is called mapping.

How Slam Works

there are two types of technology components used to achieve SLAM. The first type is sensor signal processing, including the front-end processing, which is largely dependent on the sensors used. The second type is pose-graph optimization, including the back-end processing, which is sensor-agnostic.

1. Visual Slam
2. LiDAR Slam

Command to install slam in ros2 :→ `sudo apt-get install ros-foxy-slam-toolbox`

Below I have shown Some images of my robot working using those I will explain its working and use of slam here.

1. In the first image we can see the Gazebo robot (bottom to right side of image).
2. In the second image we can see the rviz view of the robot . Here the robots sensors working can be noticed . the blue region in the object detected by the sensors using slam and in the purple region it is accordingly mapping the region where it can go.
3. In the third image I have directed the robot to navigate to another location . hence robot is moving.

4. Here while moving robot in simultaneously locating obstacle and mapping the region .
this is how it moves smoothly without any struggle.



