ROBOTIC OPERATING SYSTEMS
BATCH-A GROUP-18
FOLLOW BOT

SATYAK GB – CB.EN.U4AIE21060

HARSHANTH – CB.EN.U4AIE21016

SRIRAM SHIVKUMAR – CB.EN.U4AIE21072

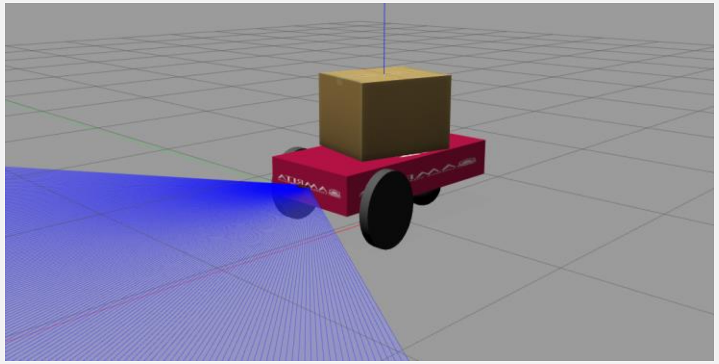
INTRODUCTION

Follow-Bot uses a laser sensor to detect and track the closest object and follow it.

Assuming the closest object is a moving human, our Follow-Bot(harshy) tracks and follows it.

The body of harshy-bot allows it to carry an object that is the size of a box.

Implementing harshy in real life allows us to walk freely without carrying anything that is smaller than a box.



USING ROS2 AND GAZEBO TO CREATE HARSHY

We have created three packages for the creation and simulation of harshy in gazebo.

The three packages are:

- **harshy** – provides all other packages
- **harshy_follow** – provides node with follow logic
- **harshy_gazebo** – robot model, simulation world and launch file

We have used pre-built gazebo world which consists of a city in which we spwan harshy and another downloaded model which is a human.

HARSHY_FOLLOW

- This package consists of harshy_follow.py, CMakeLists.txt and package.xml
- harshy_follow.py creates a ros2 node that subscribes to laser scan data and controls harshy's movement based on the closest detected object
- It also calculates linear and angular velocities using proportional control and publishes them on the 'cmd_vel' topic

HARSHY_GAZEBO

- It consists of launch, model, world and CMakeLists.txt and package.xml.
- Launch file is used for launching gazebo with the world that contains harshy and also launches the harshy_follow alongside it.
- Model file contains the models used in creating the gazebo world and also the model for harshy which is written in sdf format.

HARSHY-MODEL

The "harshy" robot model includes a chassis, caster, left and right wheels. Each link is defined with its visual and collision properties, such as geometry, material, and inertia. The visual elements determine the appearance of the robot in the simulation, while the collision elements are used for collision detection and physics calculations.

CONTD.

The code also includes a plugin named "diff_drive" that implements a Gazebo-ROS diff-drive controller. This controller provides a ROS interface for controlling the robot's movement. It specifies the joints used for the wheels, their kinematics (wheel separation and diameter), and limits for torque and acceleration. It also enables publishing odometry information and transforms for localization.

CONCLUSION

In conclusion, the developed robot "harshy" utilizing ROS2 and Gazebo simulation effectively demonstrates the ability to track and detect the closest object, assumed to be a human, using a laser sensor. It autonomously follows the human as they move, while also possessing the capability to carry objects smaller than an average box. This project exemplifies the successful integration of robotics, artificial intelligence, and sensor technologies, showcasing the potential for practical assistance in daily activities. Harshy's functionality and performance highlight the progress made in robotic applications, paving the way for further advancements in object tracking, human following, and object transportation tasks.