Information Technology Course
Module Autonomous Intelligent Systems
by Prof. Dr. Peter Nauth

Winter Semester 24/25

Obstacle Avoidance for TurtleBot3 Using ROS2 and Docker-Compose

Team: OBS-07

Hasibuzzaman
hasibuzzaman@stud.fra-uas.de
Matriculation No. 1448140

Md Sohel Rana
sohel.rana@stud.fra-uas.de
Matriculation No. 1428128

Objectives

- Implement an obstacle avoidance algorithm using ROS2.
- Utilize LIDAR for real-time obstacle detection.
- Simulate the navigation in Gazebo.
- Enable teleoperation functionality.
- Deploy using Docker for portability.

Introduction

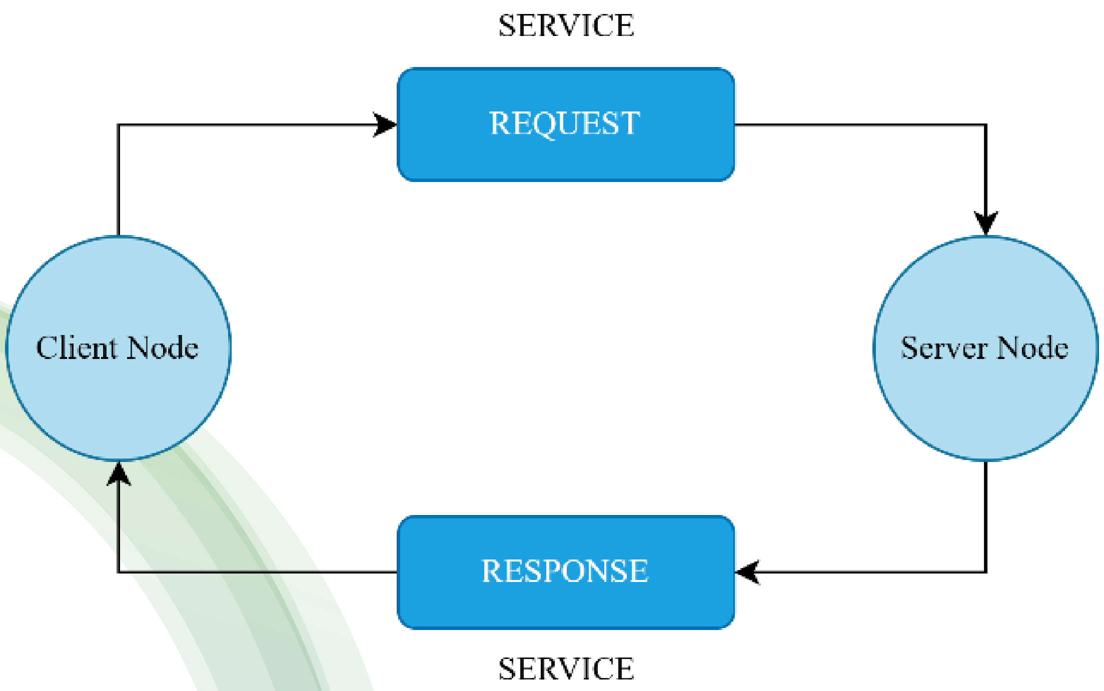


Fig. 1: ROS2 Services

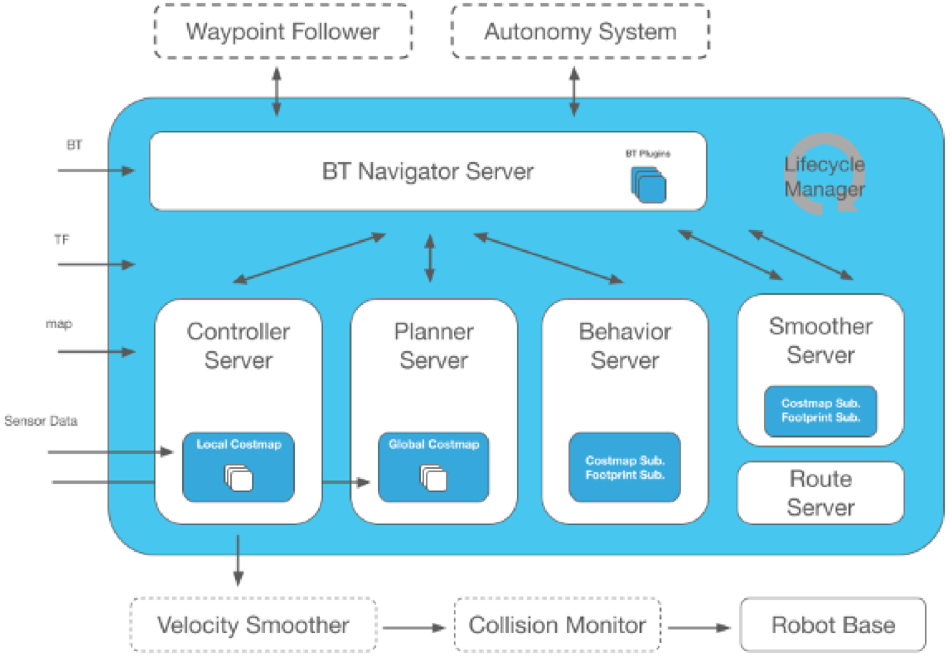


Fig. 2: Navigation2 stack.

Introduction

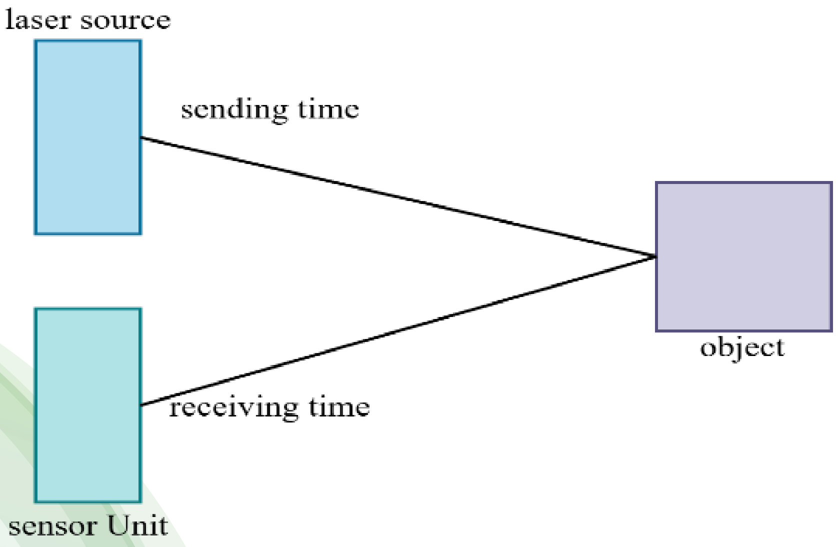


Fig. 3: LiDAR

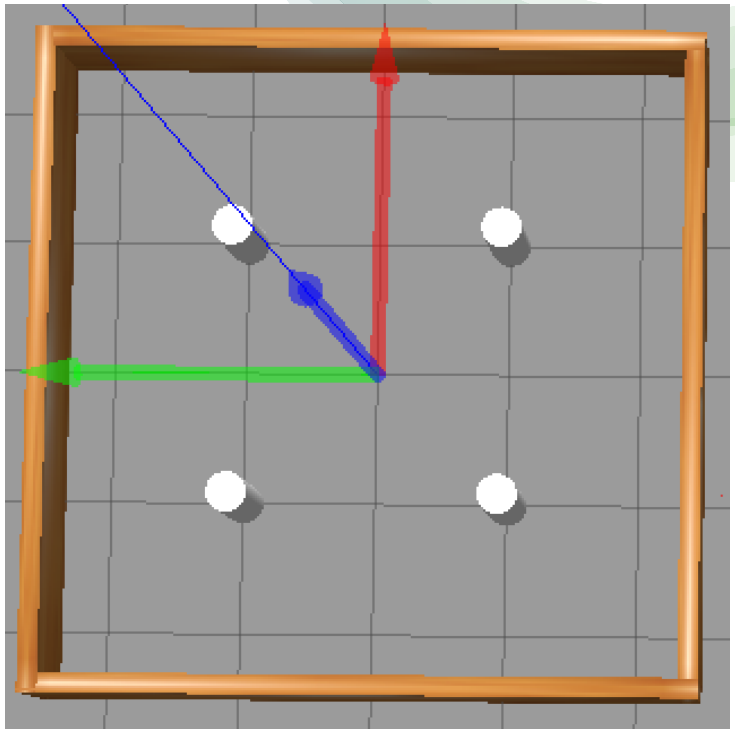


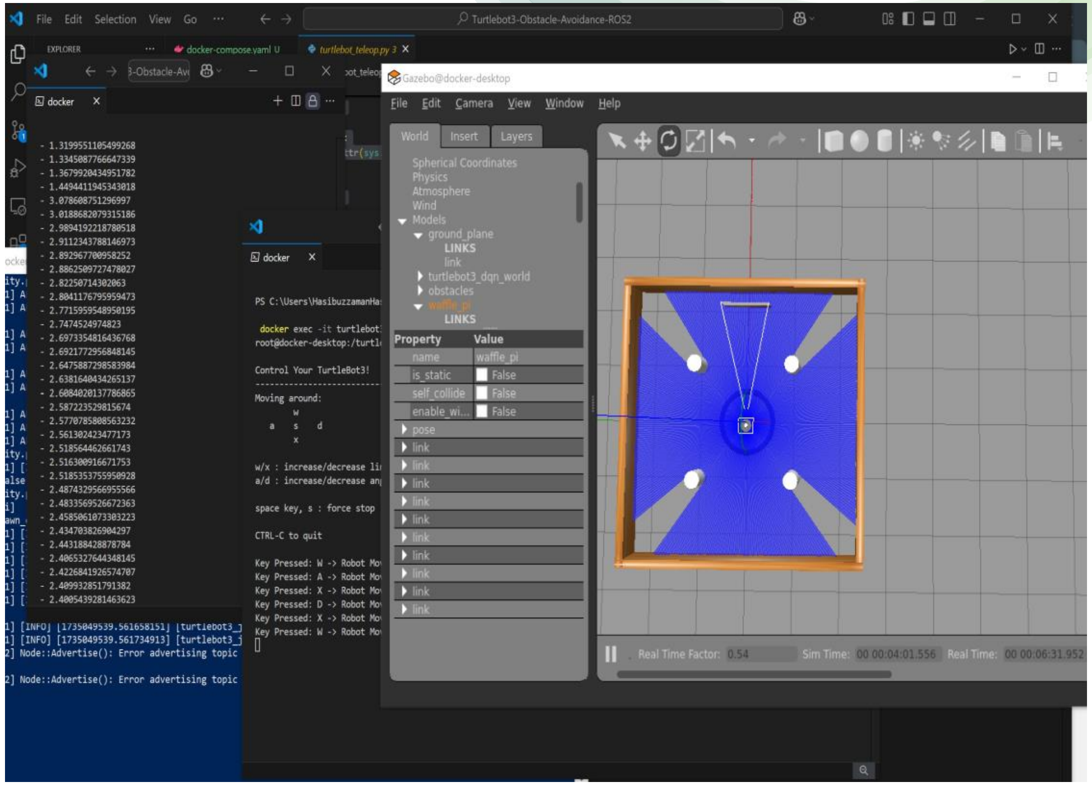
Fig. 4: Gazebo.

Methodology

System Architecture:

The system's components are as follows:

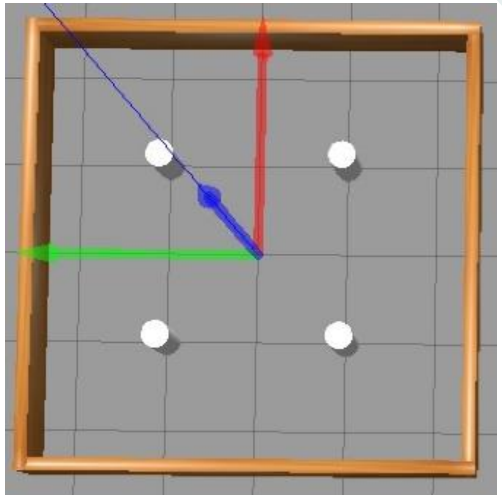
- ROS2 Humble allows real-time processing for robot control and communication.
- The Gazebo simulator offers a virtual testbed for obstacle avoidance strategies.
- Docker-Compose arranges and deploys ROS2 nodes for modular development.
- A LIDAR sensor collects environmental data to detect and avoid obstacles.

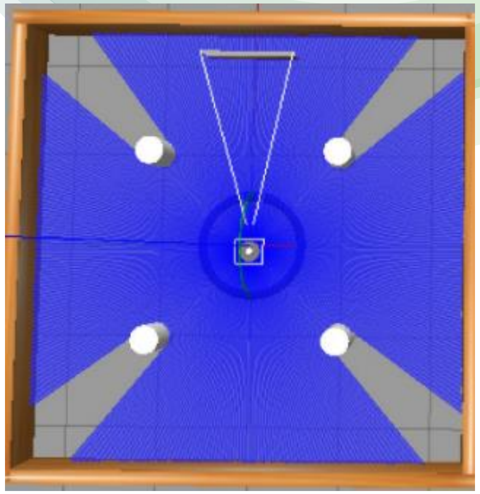


Methodology

Hardware and Software Setup:

- Hardware: TurtleBot3, LIDAR sensor.
- Software: ROS2 Humble, Gazebo, Docker, Nav2 Stack.
- Development Environment: Docker Desktop, Python scripts.





Result and Discussion

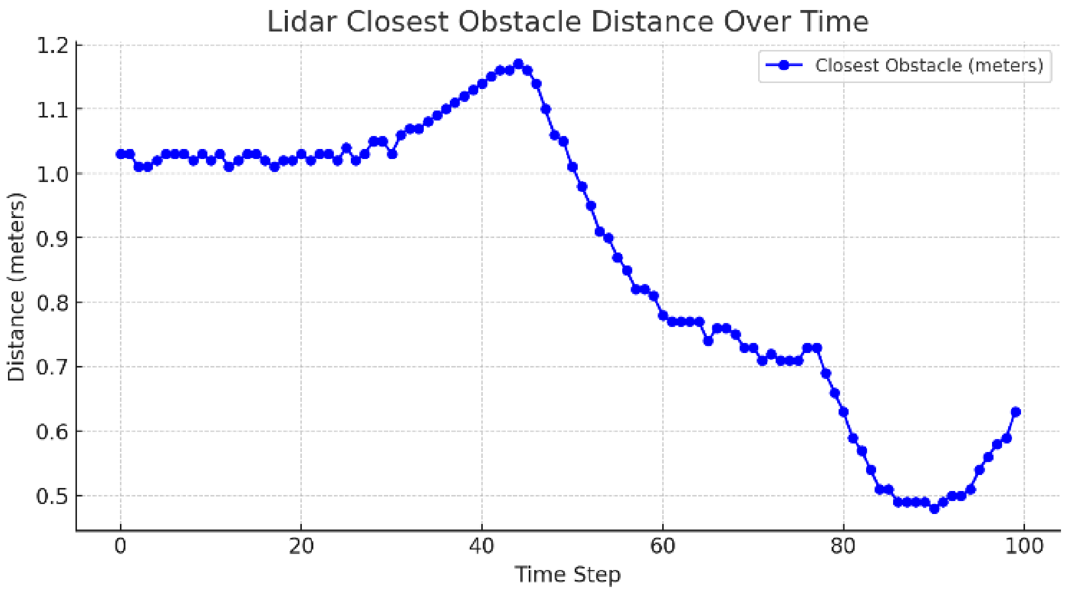


Fig. 5: LiDAR closest obstacle observation

Initially, the closest obstruction was located approximately 1.03 meters away. Gradually, the distance decreased as the robot moved forward. The robot identified an approaching collision at a distance of around 0.28 meters. When the system replied by altering its velocity, it was verified that the real-time obstacle avoidance was successful.

Result and Discussion

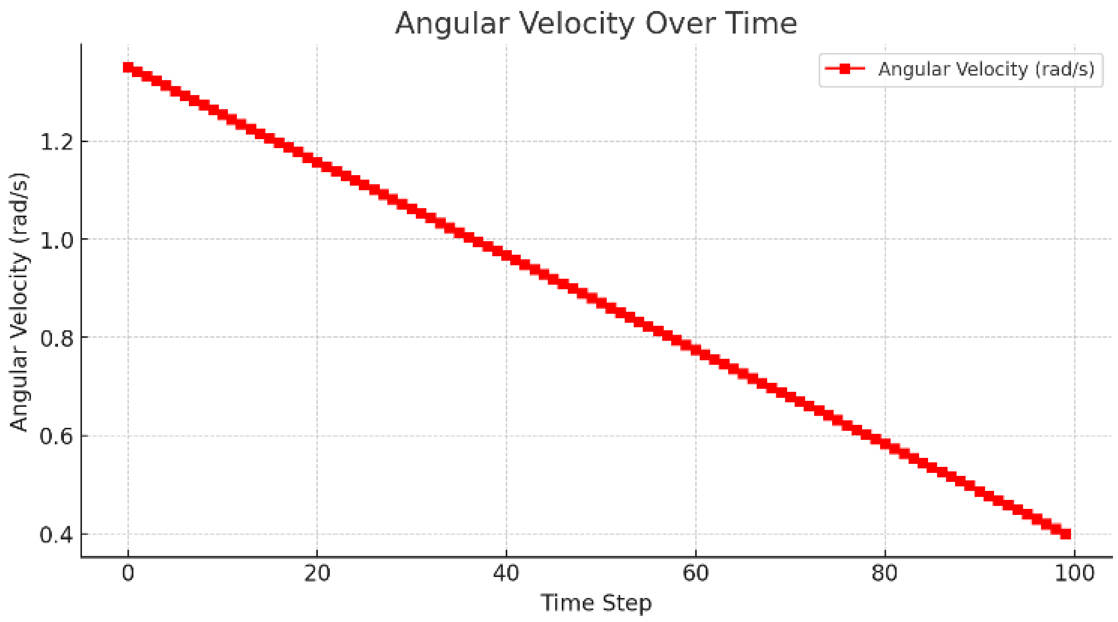


Fig. 6: Angular velociy over time.

The angular velocity was adjusted while the linear velocity remained constant at 0.26 m/s, according to the cmd_vel data. This implies that the machine was proactively turning to stay clear of obstructions.

A strong avoidance maneuver was indicated by the angular velocity increasing from 1.35 rad/s to 1.5 rad/s as the object approached.

In order to steady the trajectory, the robot's angular velocity steadily dropped as it rotated.

Once the obstruction was avoided, the velocity stabilized and onward motion was possible.

Conclusion and Future Work

- Based on the information collected, the robot showed successful object avoidance behavior by continuously analyzing its environment and changing course when an obstruction was detected within a critical threshold.
- The smooth velocity changes showed that the program dynamically changed the movement instead of halting suddenly.
- The avoidance strategy worked properly because there were zero collisions.

❑ **In the future, multi-sensor fusion and adaptive path-planning approaches will be used to improve robustness in complex scenarios.**

Thank You