# Simulation of a Robot and a Maze in Gazebo Using A* Path Planning and RealSense Pointclouds

Vedh & Ashna

## 1 Introduction

### 1.1 Problem Statement

We were given the task of simulating a maze in Gazebo and simulating a robot in the maze which contains obstacles and everyday objects like chair, table, etc. These everyday objects were pointclouds. The robot had to avoid these obstacles and find the shortest optimal path to the goal.

### 1.2 Tools Used

- ROS2 Humble
- Gazebo
- Rviz
- Rviz Navigator2
- Turtlebot3 simulations
- Meshlab
- Blender
- Cloud Compare
- Jupyter Notebook
- Intel Realsense D400 camera
- Realsense SDK Viewer

# 2 System Architecture

## 2.1 System Overview

- Ubuntu version 22.04
- ROS2 Humble
- Gazebo simulator version 11.10.2
- Rviz2
- Python 3.10

## 2.2 Topics

```
/amcl/transition_event
/bt_navigator/transition_event
/clicked_point
/clock
/cmd_vel
/controller_server/transition_event
/diagnostics
/global_costmap/global_costmap/transition_event
/goal_pose
/imu
/initialpose
/joint_states
/local_costmap/local_costmap/transition_event
/map_server/transition_event
/odom
/parameter_events
/performance_metrics
/planner_server/transition_event
/robot_description
/rosout
/scan
/tf
/tf_static
/waypoint_follower/transition_event
```

## 2.3 Nodes

- Gazebo with the TurtleBot3 simulator
- Map server

- Rviz
- A* planner

# 3 Implementation

## 3.1 Workspace and Package Structure

```
vedh_ros/
|-- src/
    |-- astar_planner/
        |-- src/
        |   |-- astar_planner/
        |       |-- __init__.py
        |       |-- planner_node.py
        |       |-- map_loader.py
        |-- launch/
        |   |-- turtlebot3_sim_launch.py
        |-- maps/
        |   |-- my_map.yaml
        |-- package.xml
        |-- setup.py
```

## 3.2 Launch File Used

- `turtlebot3_sim_launch.py` (combined launch file)

## 3.3 Important Configuration Files

- `my_map.yaml`
- `my_map.pgm`

### 3.4 How the Components Interact

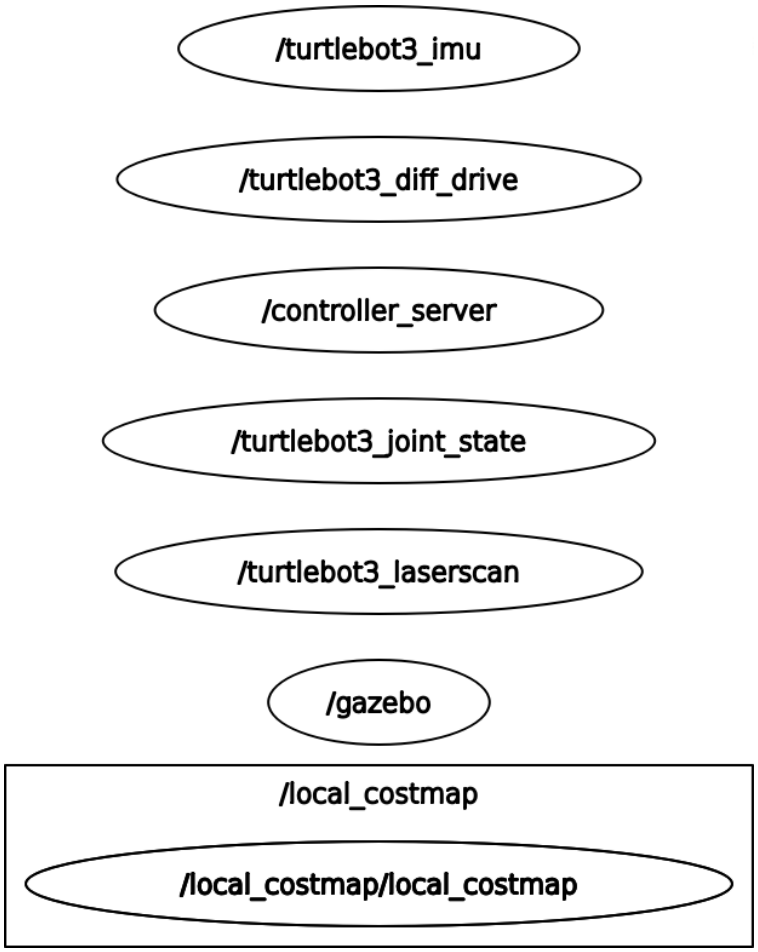


Figure 1: ROS 2 Node Graph for TurtleBot3 Navigation

## 4 Simulation

### 4.1 How We Launched and Ran the Simulation

```
ros2 launch astar_planner turtlebot3_sim_launch.py
ros2 launch turtlebot3_navigation2 navigation2.launch.py
```

### 4.2 Steps to Reproduce the Result

```
cd ~/vedh_ros
rm -rf build install log
rm -rf ~/.ros/log
ros2 daemon stop
ros2 daemon start
```

```
colcon build --symlink-install
source install/setup.bash
```

### 4.3 Gazebo and RViz Setup

- World file: `turtlebot3_world`
- Robot spawned: TurtleBot3 Burger
- Plugins:
  - Differential drive plugin
  - IMU plugin
  - LiDAR plugin

## 5 Results

### 5.1 What Worked

1. TurtleBot3 spawned in Gazebo with maze
2. RViz and Nav2 opened
3. Goal set and path formed
4. Path followed by robot (overriding our code)
5. Pointcloud of a chair generated using RealSense
6. Converted pointcloud to .dae format
7. Custom chair pointcloud added to Gazebo

### 5.2 What Did Not Work

- Could not make own robot with custom plugin
- Could not implement own A* code
- Could not get RealSense-generated pointcloud into Gazebo

# 6 Challenges Faced

## 6.1 Problems

1. Lack of familiarity with Ubuntu
2. Complexity of ROS2 parameters
3. Confusion in URDF/plugin integration
4. Lack of resources and time
5. Difficulties with ROS2 packages and robot selection
6. Integration and visualization issues

## 6.2 How We Solved Them

- Used TurtleBot3 instead of Iris
- Found compatible Gazebo version
- Used Blender and YouTube tutorials for integration
- Converted mesh to .dae via Meshlab and Blender
- Used RViz Nav2 to visualize robot path

## 6.3 What We Learnt

- Combining depth images into 3D pointcloud
- A* algorithm
- ROS2 package structure
- Mesh conversion and usage of RViz/Nav2

## 6.4 Features We Would Like to Add

- Custom A* node for path planning
- Use different robots
- Generate better pointcloud using RealSense

# 7 References


1. https://github.com/ROBOTIS-GIT/turtlebot3
2. https://github.com/ROBOTIS-GIT/turtlebot3_simulations
3. https://github.com/IntelRealSense/realsense-ros
4. https://gazebosim.org/home
5. https://www.geeksforgeeks.org/a-search-algorithm/
6. https://robotics.stackexchange.com/
7. https://www.ros.org/
8. https://youtu.be/ySN5Wnu88nE?si=8R-NHAd9O-_Qgq_X
9. https://youtu.be/-L-WgKMFuhE?si=haQBEODBOgMYEdsb
10. https://rs-india.org/