CEN449 Project I: Autonomous Navigation Mobile Robot Report

Student Name: İbrahim Emre YILDIZ

Student ID: 2020555069

Date: December 21, 2025

Video Link: https://youtu.be/aqQFxjvh7aA

---

1. Introduction and Objectives

The primary objective of this project is to develop and simulate an autonomous mobile robot capable of navigating and exploring an unknown environment using ROS and Gazebo. Unlike traditional navigation tasks, this project specifically requires the robot to operate without a pre-saved map, relying instead on real-time SLAM (Simultaneous Localization and Mapping).

Key Objectives:

- Implementing core ROS principles including nodes, topics, and launch files.
- Integrating a mobile robot model with a navigation stack in a Gazebo simulation.
- Achieving autonomous obstacle avoidance and path planning in a dynamic environment.

---

2. System Architecture and Design

The system follows a modular ROS architecture where multiple nodes communicate to achieve autonomous behavior.

- **Gazebo Simulator:** Provides the physics engine and the 3D world with 7-8 obstacles.
- **Gmapping Node:** Performs SLAM by processing LaserScan data and Odometry to build a 2D occupancy grid map in real-time.
- **Move Base Node:** The core navigation unit containing the Global Planner (for long-distance pathing) and the Local Planner (for immediate obstacle avoidance).
- **RViz:** Used for visualizing the robot's sensor data, the generated map, and the planned global path.

---

3. Implementation Details and Key Algorithms

The implementation involved configuring a custom mobile robot and tuning the ROS Navigation Stack.

A. SLAM and Mapping

I utilized the **Gmapping** algorithm to satisfy the "no pre-saved map" requirement. The robot starts in a completely unknown (gray) area in RViz and gradually discovers the environment through its Lidar sensor.

B. Navigation and Path Planning

The navigation is handled by the **move_base** package:

- **Global Planner:** Uses the A* or Dijkstra algorithm to find the shortest path to the goal on the currently known map.
- **Local Planner (DWA):** Uses the Dynamic Window Approach to calculate safe velocity commands (v, ω) that avoid obstacles while following the global path.

C. Technical Optimization

To ensure robustness, the inflation_radius was set to **0.20 meters**, providing a safety buffer around obstacles to prevent collisions during tight turns.

---

4. Challenges Encountered and Solutions

During the development process, several technical challenges were addressed:

1. **The Self-Collision Issue:** Initially, the robot was unable to plan a path because it perceived itself as an obstacle.
    - **Solution:** I modified the URDF/Xacro file's Lidar plugin, increasing the <min> range from **0.10m** to **0.30m**. This ensured the laser ignored the robot's own chassis.
2. **Global Planner Failures:** The robot often failed to find paths to "Unknown Space."
    - **Solution:** I implemented a manual exploration phase using teleop_twist_keyboard to clear immediate surroundings before initiating autonomous goals.

---

5. Discussion of Results and Limitations

The simulation successfully demonstrated autonomous navigation:

- **Functionality:** The robot successfully navigated between 7-8 obstacles and reached designated goals without a prior map.
- **Robustness:** The integration of real-time SLAM allowed the robot to adjust its path as new obstacles were discovered.
- **Limitations:** The CPU usage in the virtual machine occasionally caused lag in laser data processing, which was mitigated by increasing the VM's resource allocation.

---

6. Conclusion and Future Improvements

This project successfully bridged the gap between ROS theory and hands-on simulation. The robot demonstrated reliable autonomous exploration and navigation.

Future Improvements:

- Implementing **Autonomous Exploration** nodes (e.g., Frontier Exploration) to remove the need for manual goal setting.
- Integrating a **Camera sensor** for visual SLAM to improve localization accuracy in feature-less environments.