Advanced Localization Technologies for Autonomous Robotic Apple harvesting

Gabriel Basus, Advisor: Siavash Farzan
Cal Poly College of Engineering
Department of Electrical Engineering

Figure 1: Picture of the robot with the integrated sensor mounted on top.

Figure 2: A picture of an apple orchard, exemplifying a location in which the robot developed in this research could be used.

LEARN BY DOING
CAL POLY
Noyce School *of* Applied Computing
COLLEGE OF ENGINEERING

Abstract

- **Goal:** Build a base robotic platform capable of autonomous localization, mapping, and autonomous navigation for autonomous robotic apple harvesting.
- **Methods:** Setting up an Unmanned Ground Vehicle, integrating a 2D LiDAR and Running SLAM, localization, and navigation codes.
- **Results:** The robot can use the integrated LiDAR to create a 2D map of its location, localize itself within the map, and navigate through the map to a given goal.
- **Conclusions:** This study contributes immensely to the field of agriculture by providing a robotic platform that can be used to automate agricultural tasks.

Materials and Methods

Simulation
- The robot can be simulated using Gazebo and RVIZ
- Gazebo simulates the robot by generating data based on a simulated environment

→

Robot Setup
- Save disk image
- Upgrade to ROS 2 Humble

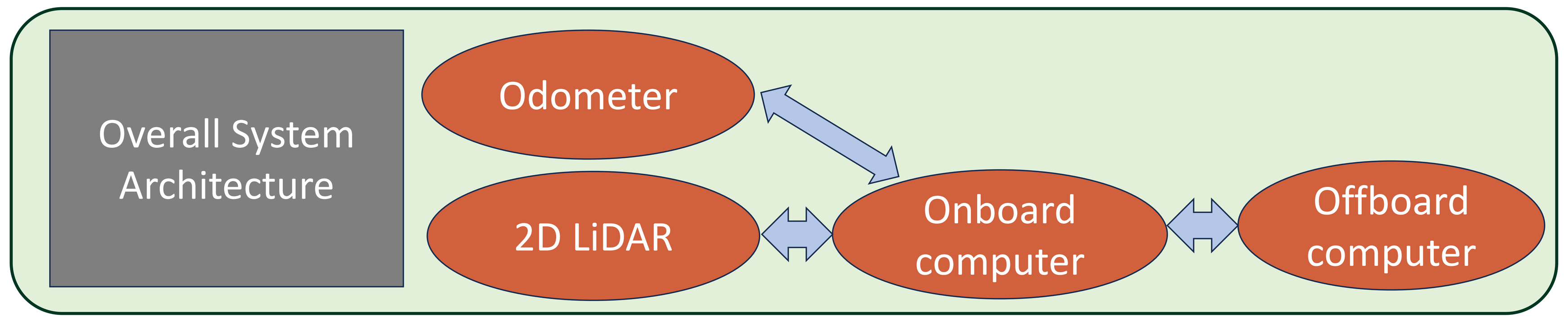
→

Sensor Integration
- Re-pin power wire to make it compatible with the robot
- Mount using 3D printed component

Methodologies

Simultaneous Localization and Mapping (SLAM)

- **Algorithm Used**: slam_toolbox with advanced scan matching and pose graph optimization.
- **LiDAR Scan Matching**:
  - Employs **Karto SLAM** algorithms for aligning sequential LiDAR scans.
  - Utilizes **Iterative Closest Point (ICP)** and **Correlative Scan Matching** for precise map building.
- **Pose Graph Optimization**:
  - Constructs a pose graph where each node represents a robot pose.
  - Implements **Loop Closure Detection** to correct drift by identifying previously visited locations.
  - Applies **Graph-Based Optimization** techniques (e.g., SPA, GTSAM) for global map consistency.
- **Map Generation**:
  - Produces a 2D occupancy grid map representing obstacles and free space.
  - Allows adjustable **map resolution** and **update rates** to suit different environments.



Methodologies

Localization

- **Algorithm Used**: Adaptive Monte Carlo Localization (AMCL) utilizing particle filters.
- **Particle Filter Mechanism**:
  - Maintains a set of particles representing possible robot poses.
  - **Prediction Step**: Updates particles based on motion models using odometry data.
  - **Update Step**: Adjusts particle weights by comparing LiDAR data to the existing map.
- **Sensor Integration**:
  - **LiDAR Data**: Compares observed scans with predicted scans to redefine particle weights.
  - **Odometry Data**: Provides motion estimates for particle prediction and movement modeling.
- **Adaptive Resampling**:
  - Dynamically adjusts the number of particles using **KLD-sampling** for computational efficiency.
  - Concentrates computational resources on the most probable robot locations.
- The robot can localize itself in a room with less than 21% error.

Navigation

- **Navigation Stack**: Nav2 for comprehensive path planning and control.
- **Global Path Planning**:
  - Utilizes **A*** or **Dijkstra's Algorithm** to compute optimal paths on the global costmap.
  - Considers obstacle costs and robot kinematics in path formulation.
- **Local Path Planning**:
  - Implements the **Dynamic Window Approach (DWA)** for real time obstacle avoidance and path following.
  - Calculates feasible velocity commands within the robot's dynamic constraints and environment.
- **Costmap Management**:
  - **Global Costmap**: Represents the static environment derived from the SLAM-generated map.
  - **Local Costmap**: Continuously updated with live sensor data to account for dynamic obstacles.
  - **Obstacle Inflation**: Adds safety buffers around obstacles based on the robot's footprint and safety requirements.
- **Recovery Behaviors**:
  - **Clearing Rotations**: Performs in-place rotations to reassess surroundings when navigation is impeded.
  - **Re-planning**: Automatically generates new paths if obstacles are detected along the current trajectory.

Results

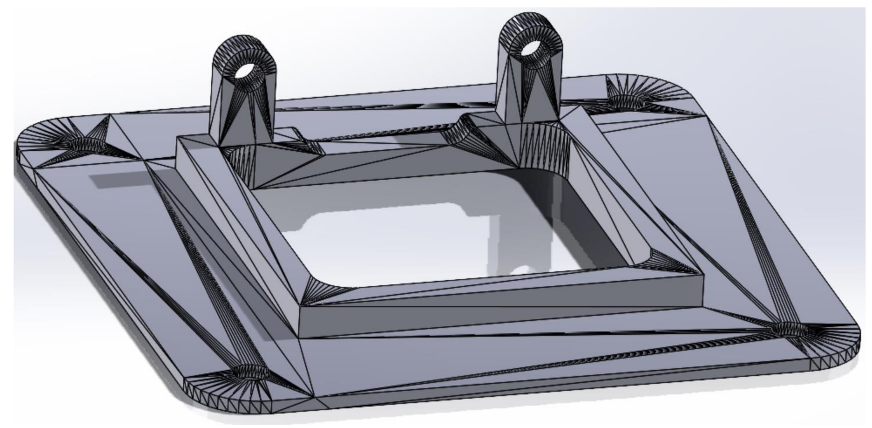


Figure 3: Design of the 3D printed Mount used to mount the LiDAR.

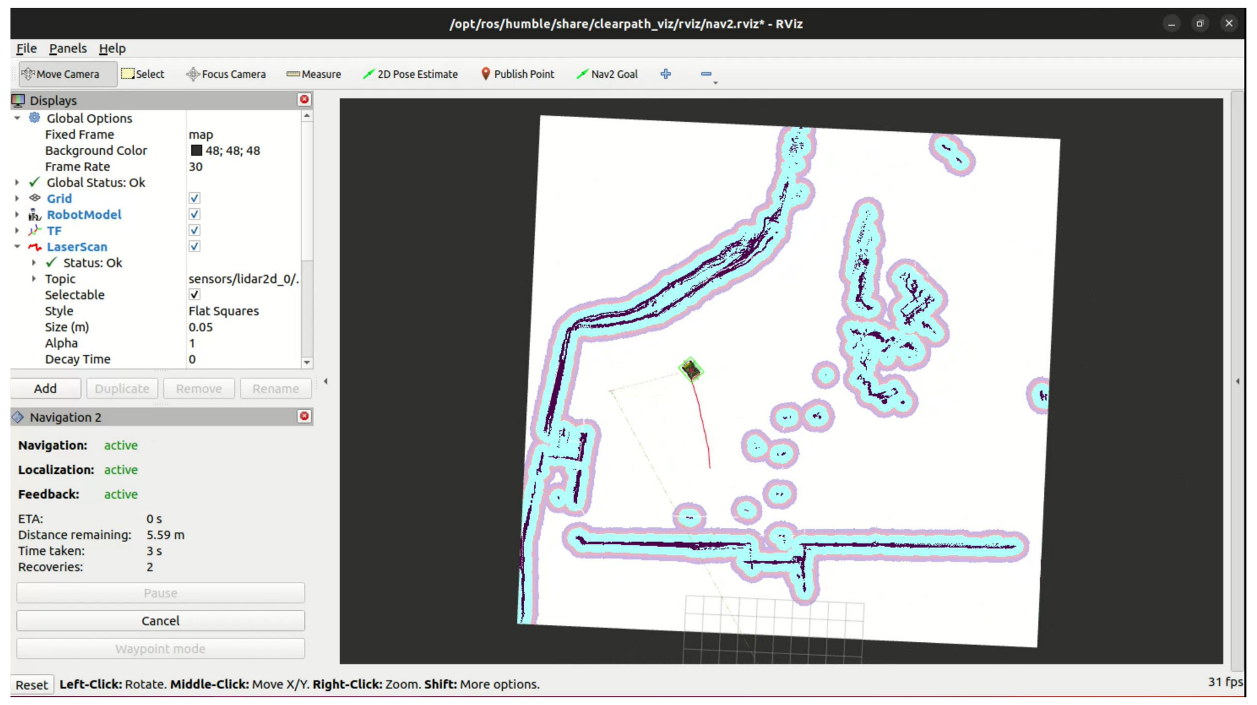


Figure 4: RVIZ environment where the map is displayed. The map being displayed includes the costmap zones in blue and purple.

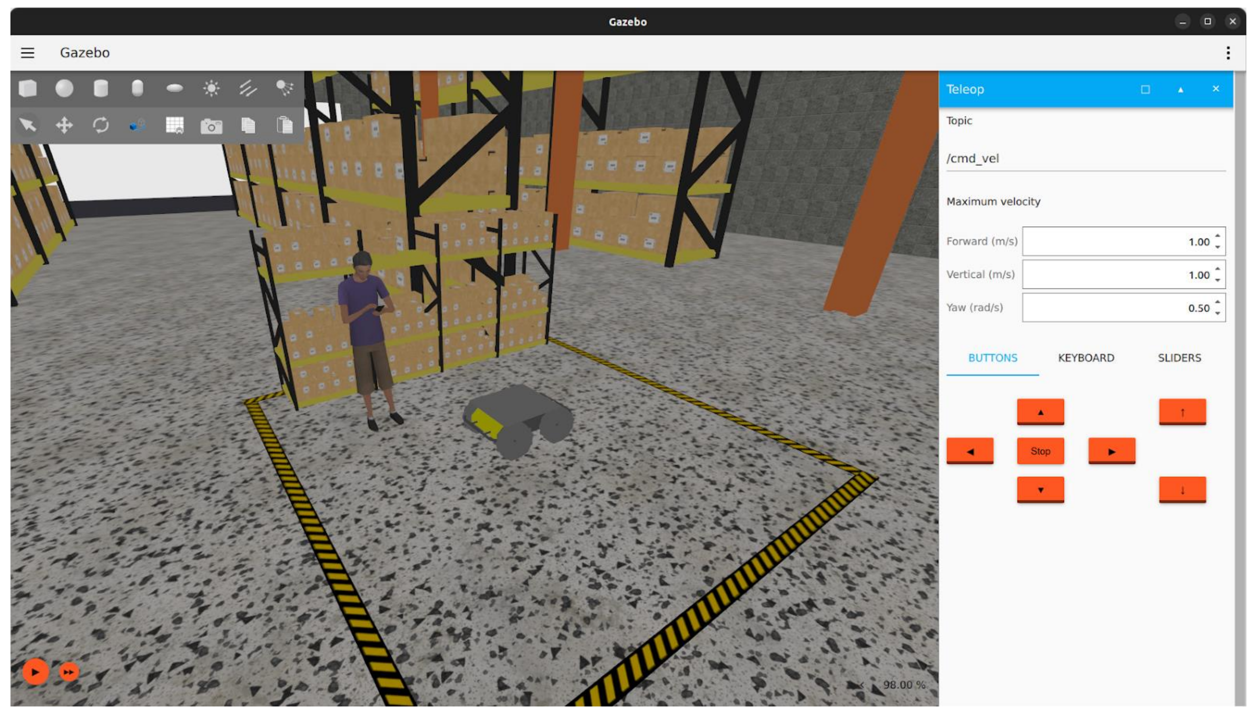


Figure 5: Screenshot of a Gazebo Simulation of the robot in a simulated environment

REFERENCES

1 *Husky user manual*. Clearpath Robotics Documentation. (2024, March 12). https://docs.clearpathrobotics.com/docs/robots/outdoor_robots/husky/user_manual_husky/
2 *Hokuyo UST10-LX User Manual*.

Acknowledgements

This research was funded by the Noyce School of Applied Computing and the Electrical Engineering Department at Cal Poly.