

Autonomous Road Dust Collector

ME UGRP PRESENTATION

YASH PURSWANI
ME22B214

Project Overview

- Road dust is a major source of urban PM_{10} to $PM_{2.5}$ pollution.
- Traffic continuously resuspends fine, harmful particles.
- Fine PM causes serious respiratory and cardiovascular diseases.
- Manual road cleaning exposes workers to high dust levels.
- Mechanical cleaning often leads to secondary dust re-suspension.
- Urban growth demands safer, scalable road-cleaning solutions.



We propose an **Autonomous Road Dust Collection Robot for safe, efficient, and fine-particle-controlled cleaning.**

Objectives

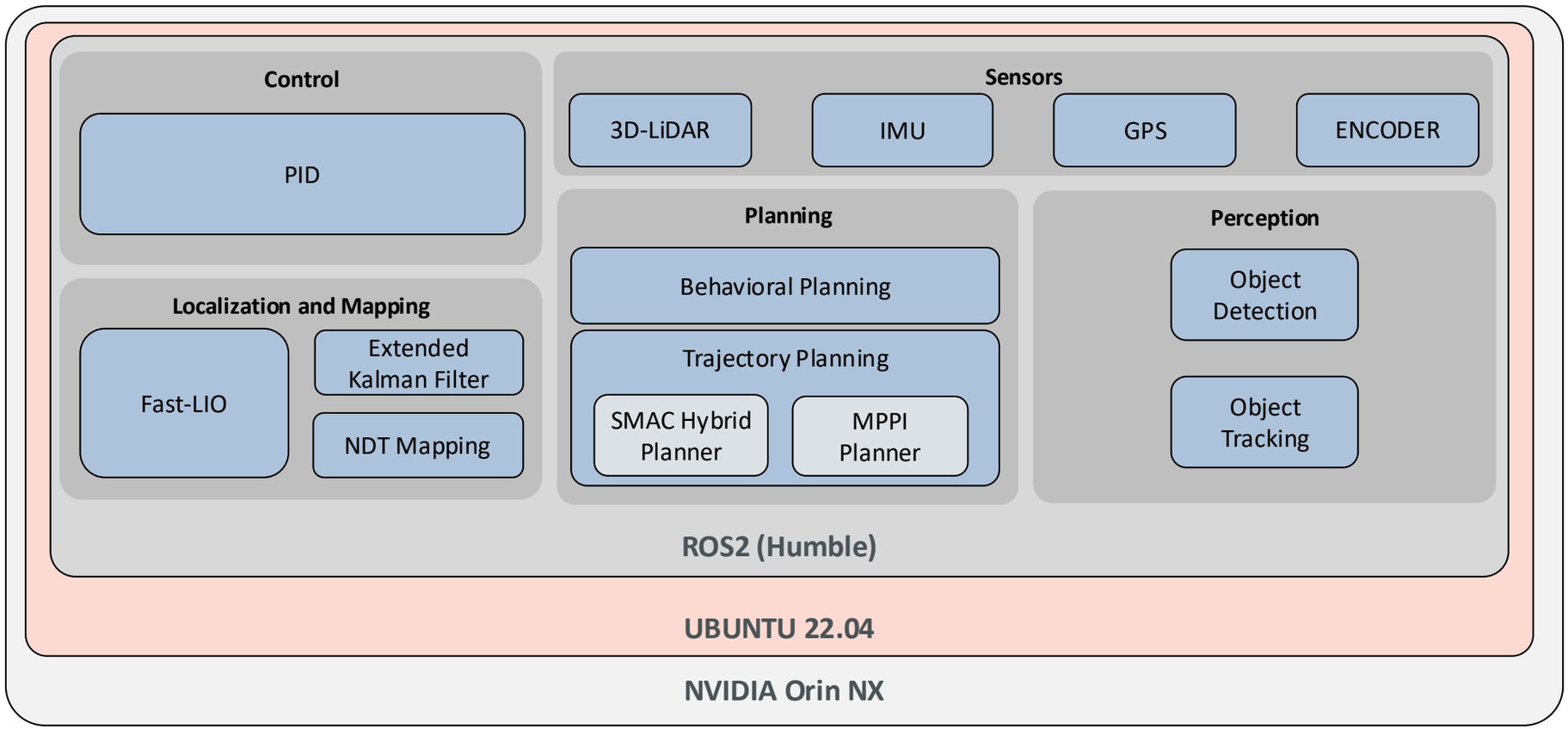
PRIMARY OBJECTIVES

- Develop a robust system framework for efficient road dust collection and particle containment.
- Enable autonomous navigation and mobility for unmanned operation in structured and semi-structured environments.

TECHNICAL SUB-TASKS

- **Multiple Sensor-Fusion** — Integration of GNSS, LiDAR, and IMU with optimized sensor placement to ensure robust perception and minimize electromagnetic interference (EMI).
- **Pose Estimation** — Get accurate robot state and odometry estimation using an Extended Kalman Filter (EKF).
- **Path Planning** — Integrate global and local path-planning algorithms that optimize coverage area while minimizing energy consumption.
- **Obstacle Avoidance** — Implement a real-time reactive safety system capable of identifying and navigating around static obstacles.
- **Testing and Tunning** — To optimize control parameters and path-planning algorithms through iterative performance testing

System Architecture



System Architecture

HARDWARE

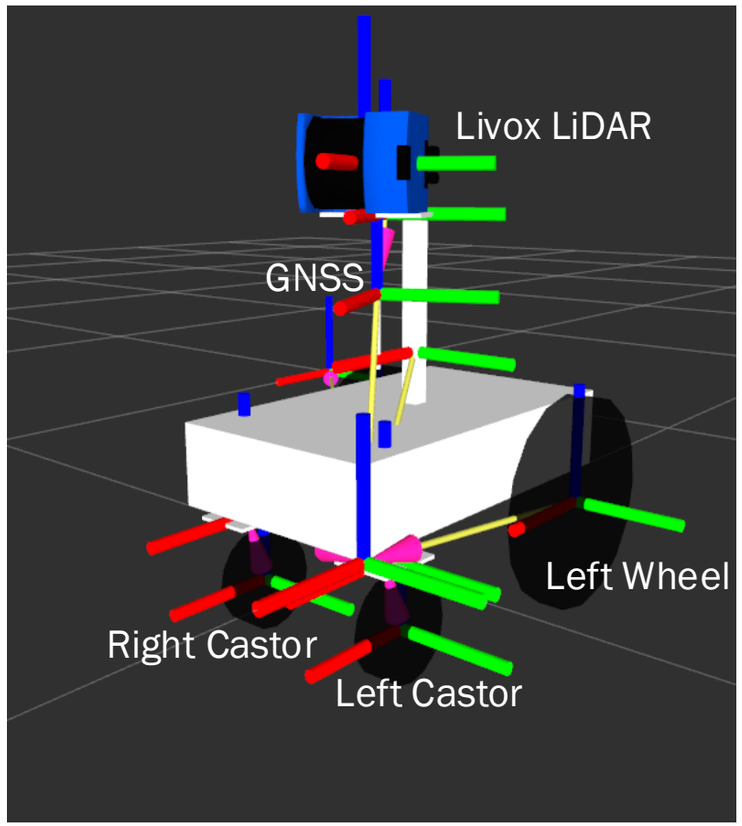
- **Chassis:** Lightweight modular aluminium extrusion frame
- **Drive System:** Hub motors with independent control (ODrive S1)
- **Motion Control:** Encoder-based closed-loop PID velocity control
- **Onboard Computer:** NVIDIA Jetson NX for mapping and navigation
- **Battery:** AshvaVolt 24V with 10000mAh capacity for extended range

SOFTWARE

- **Framework:** ROS 2 Humble for modular robotic software integration
- **Perception Input:** Livox Mid360 LiDAR data for environment sensing
- **Odometry & Mapping:** FAST-LIO2 for real-time LiDAR–inertial SLAM
- **Localization Support:** Simplertk3b GNSS-compass for global position & heading
- **Navigation Stack:** Navigation2 (Nav2) for path planning and motion control

Sensor Placement

- **Avoid electromagnetic interference (EMI):** Keep GPS, IMU, and communication modules away from motors, drivers, and high-current wiring to prevent signal degradation and noise.
- **Clear 360° field of view:** Position LiDARs and cameras to avoid occlusions from the robot body, ensuring maximum environmental coverage.
- **Accurate TF setup in URDF:** Correct sensor transforms enable proper frame alignment and reliable sensor fusion.
- **Consistent environment perception:** Good placement improves obstacle detection, localization accuracy, and overall navigation robustness.



Robot State Estimation

LiDAR-Inertial Odometry

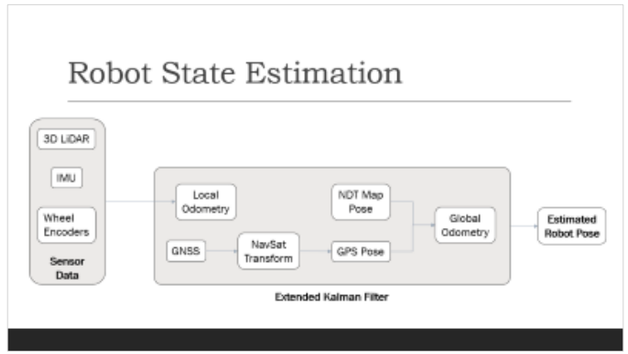
- Utilized **FAST-LIO2** to tightly couple 3D LiDAR scans with IMU data.
- Provides **high-frequency, low-drift** motion estimation resilient to robot vibrations and motion blur.

Map-Based Localization

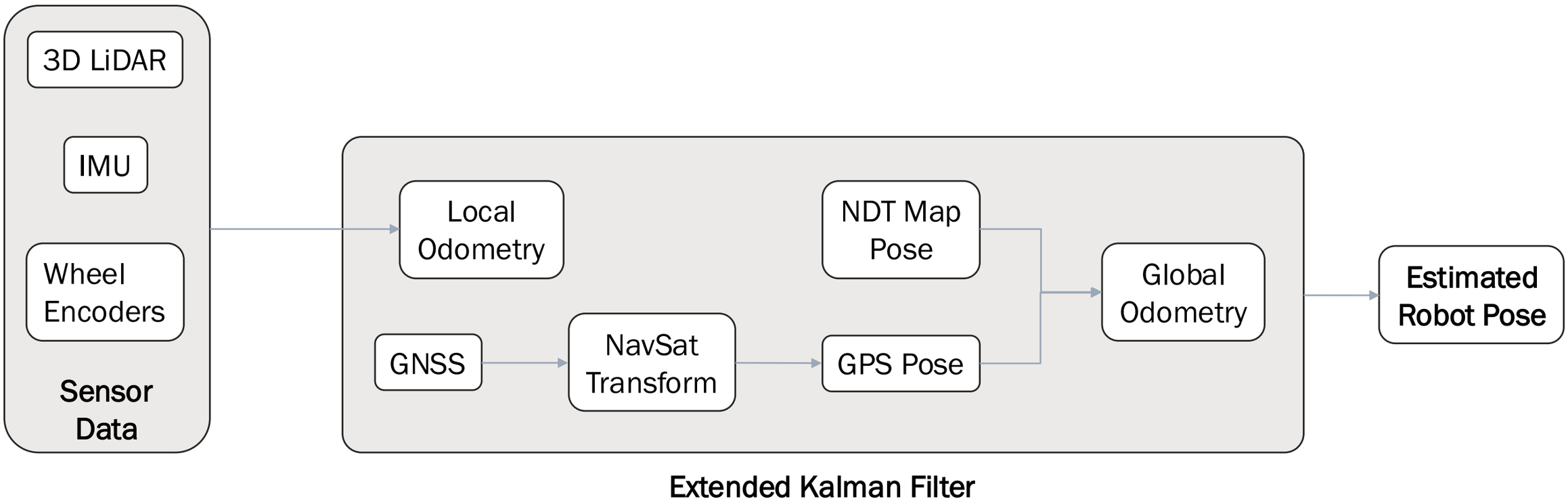
- Employed **Normal Distributions Transform (NDT)** to match real-time scans against the pre-built global map.
- Corrects accumulated odometry errors to maintain sub-decimeter positioning accuracy over extended runs

Dual-EKF Pose Estimation

- **Local EKF:** Fuses Wheel Encoders, IMU and LiDAR odometry to ensure smooth, continuous velocity for the controller.
- **Global EKF:** Integrates GNSS/Compass data and NDT map pose to correct long-term drift and align the robot with the global map frame.



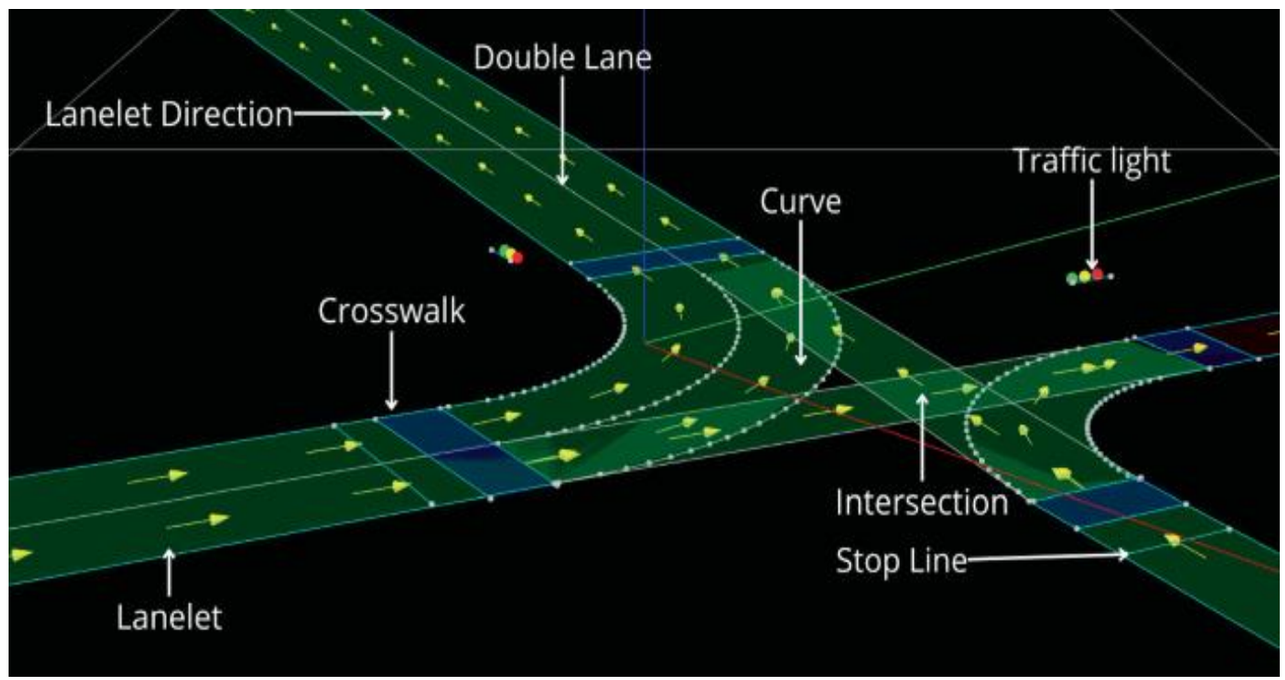
Robot State Estimation



Lanelets

Lanelets provide a **structured vector-based representation of road segments**, embedding semantic and geometric information about the environment.

- **Encodes road structure:** Represents lanes, boundaries, and drivable regions as connected segments.
- **Generated from mapping data:** Can be created through post-processing LiDAR point cloud maps to extract road geometry.
- **Improves navigation efficiency:** Enables structured path generation instead of unstructured free-space exploration.
- **Supports coverage planning:** Used as input for coverage planner to generate efficient and systematic road-cleaning paths.

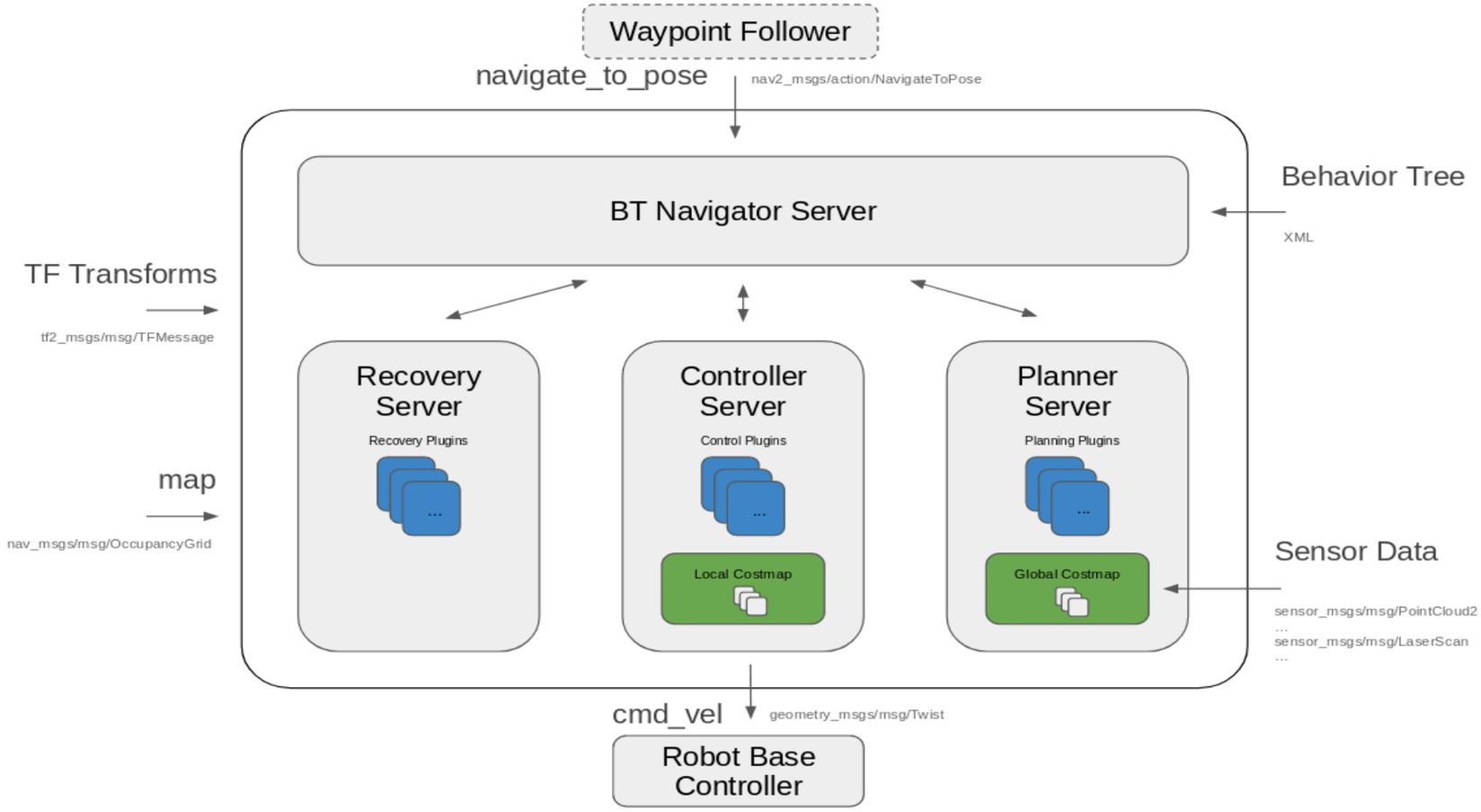


Navigation2 Architecture

Nav2 is structured as a set of **lifecycle-managed ROS 2 nodes** coordinated through **action servers** and **behavior trees**. This architecture enables flexible sequencing of navigation tasks, modular plugin-based components, and robust handling of failures.

The navigation pipeline used in this project consists of the following core components:

- **Costmaps** — Provide 2D environmental representations that model obstacles and free space for safe path planning and control.
- **Planner Server** — Generates a collision-free **global path** from the robot's current pose to the goal using the global costmap.
- **Controller Server** — Computes real-time **velocity commands** to follow the global path while avoiding dynamic obstacles using the local costmap.
- **Recovery Behaviors** — Execute corrective actions (e.g., spin, backup, clear costmap) when planning or control fails, improving system robustness.
- **Velocity Smoother** — Filters and limits velocity commands to ensure smooth, stable, and safe robot motion.



Navigation2 Architecture

Credits: [Nav2 — Nav2 1.0.0 documentation]

Costmaps

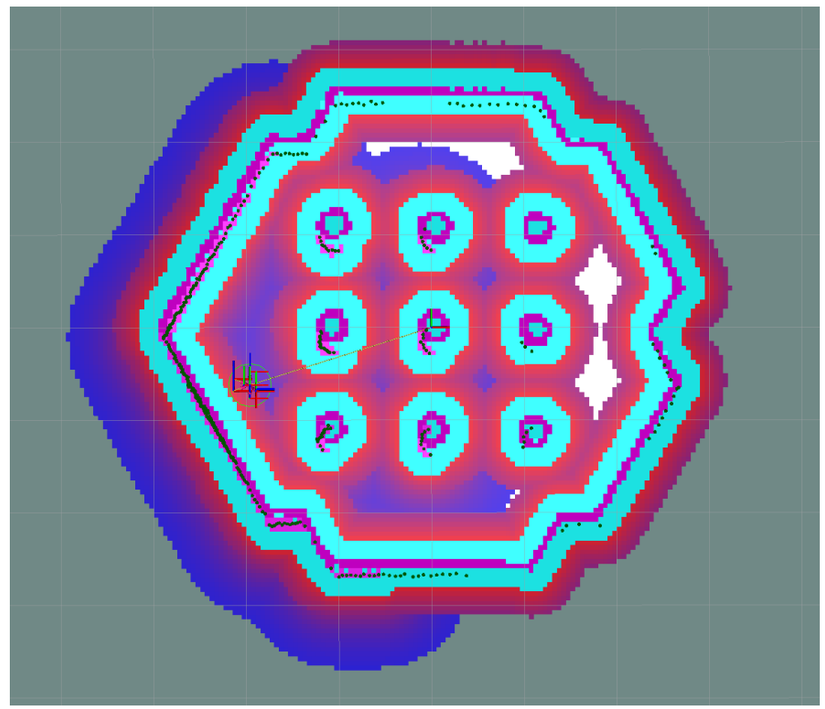
Costmaps provide a **2D grid representation** of the environment, assigning costs to areas based on obstacle proximity and traversability.

Global Costmap

- Covers the entire known environment
- Used by the global planner. Built from:
  - Static map (SLAM / map server)
  - Sensor-detected obstacles
  - Inflation layer (safety buffer)

Local Costmap

- Rolling window centered on robot
- Used by the controller
- Updates at high frequency
- Focuses on dynamic obstacles



Common Layers

- **Obstacle Layer** – LiDAR data
- **Inflation Layer** – Safety margin around obstacles
- **Static Layer** – map from SLAM

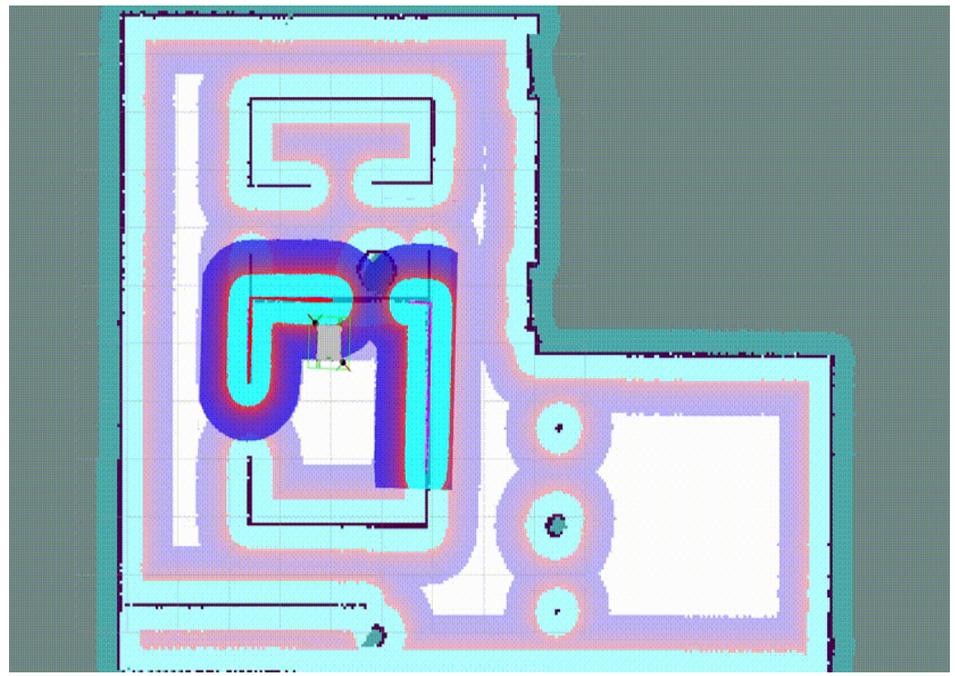
Planner and Controller Server

Planner Server – SMAC Hybrid A*

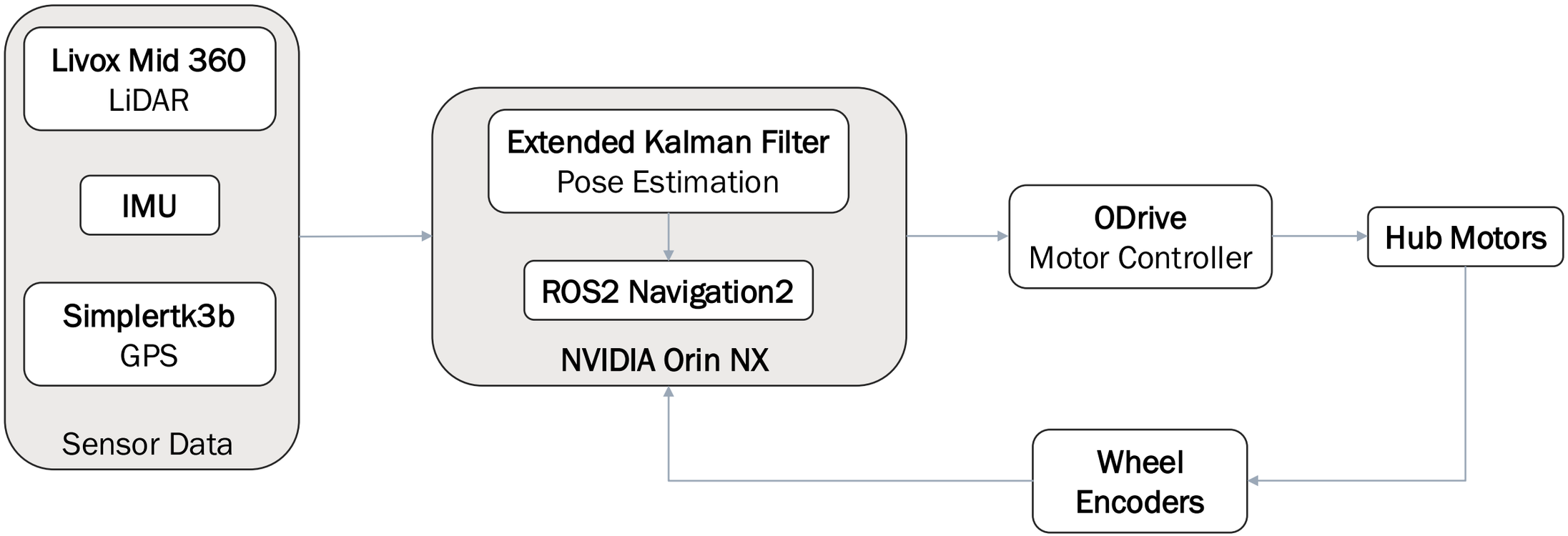
- Generates a collision-free global path to the goal
- Uses Hybrid A* (SE2 planning with orientation)
- Accounts for robot kinematics (non-holonomic motion)
- Optimized for large environments and complex turns

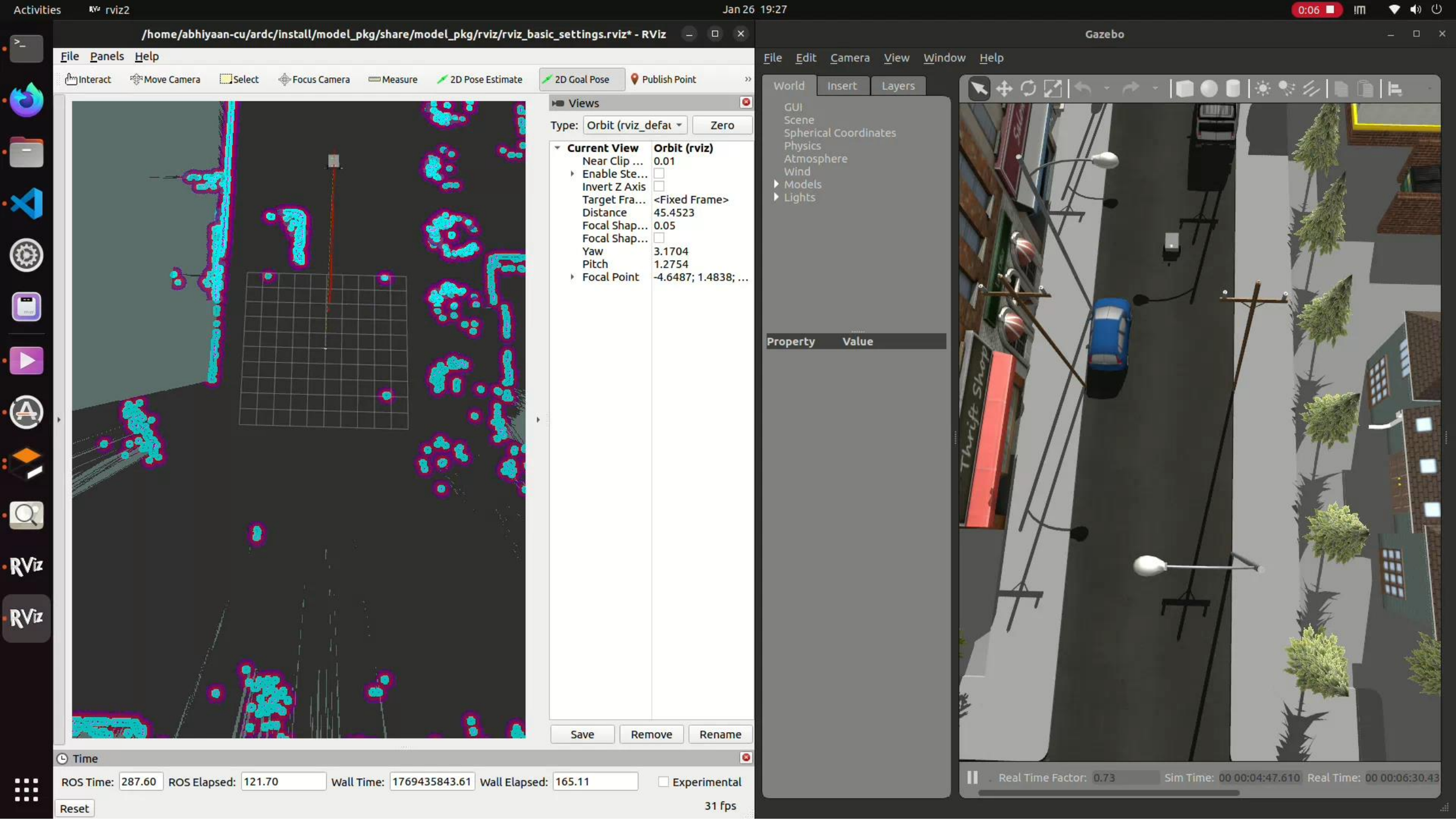
Controller Server – MPPI Controller

- Computes real-time velocity commands (`/cmd_vel`)
- Uses Model Predictive Path Integral (MPPI) control
- Samples motion trajectories and selects the lowest-cost option
- Provides smooth, stable, and optimal motion tracking

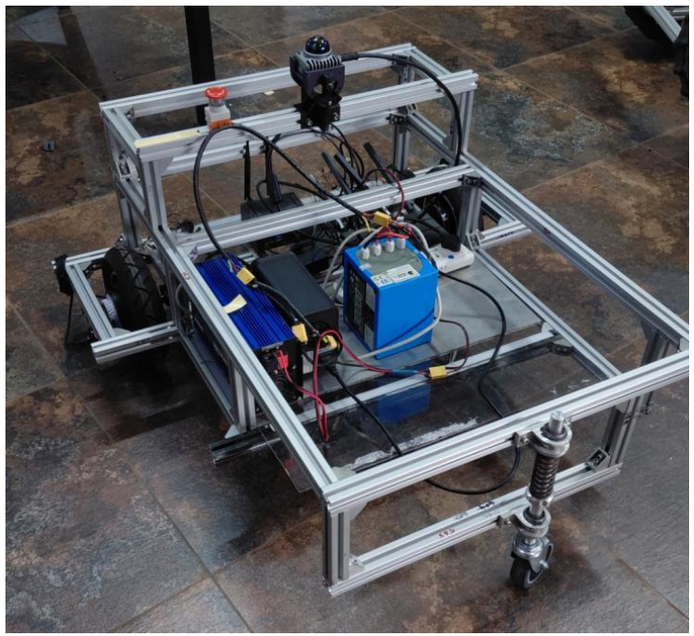


Navigation Control Loop

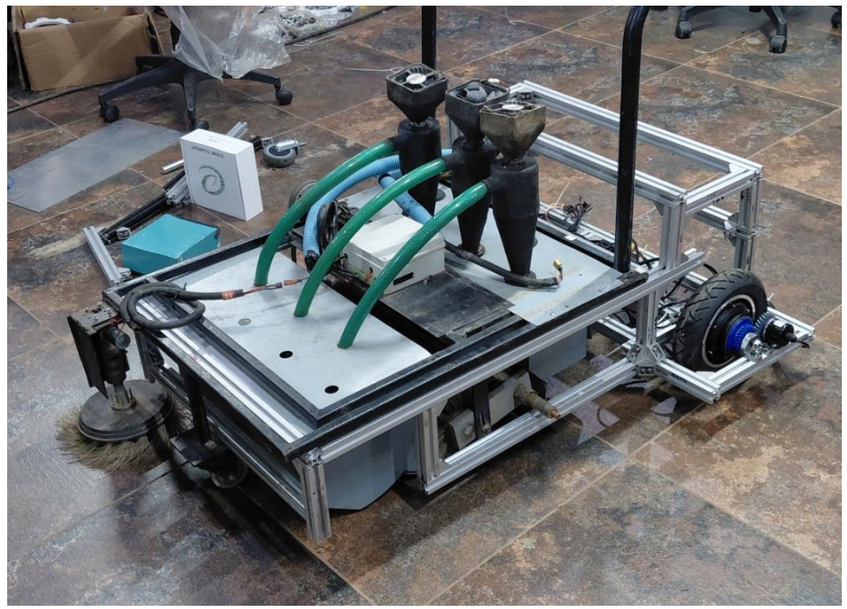



Activities
rviz2
Jan 26 19:27
0:06
/home/abhiyaan-cu/ardc/install/model_pkg/share/model_pkg/rviz/rviz_basic_settings.rviz* - RViz
File Panels Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Goal Pose
Publish Point
Views
Type: Orbit (rviz_defa
Zero
Current View Orbit (rviz)
Near Clip ... 0.01
Enable Ste...
Invert Z Axis
Target Fra... <Fixed Frame>
Distance 45.4523
Focal Shap... 0.05
Focal Shap...
Yaw 3.1704
Pitch 1.2754
Focal Point -4.6487; 1.4838; ...
Save
Remove
Rename
Time
ROS Time: 287.60
ROS Elapsed: 121.70
Wall Time: 1769435843.61
Wall Elapsed: 165.11
Experimental
Reset
31 fps
Gazebo
File Edit Camera View Window Help
World
Insert
Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property
Value
Thrift Shop
Real Time Factor: 0.73
Sim Time: 00 00:04:47.610
Real Time: 00 00:06:30.43

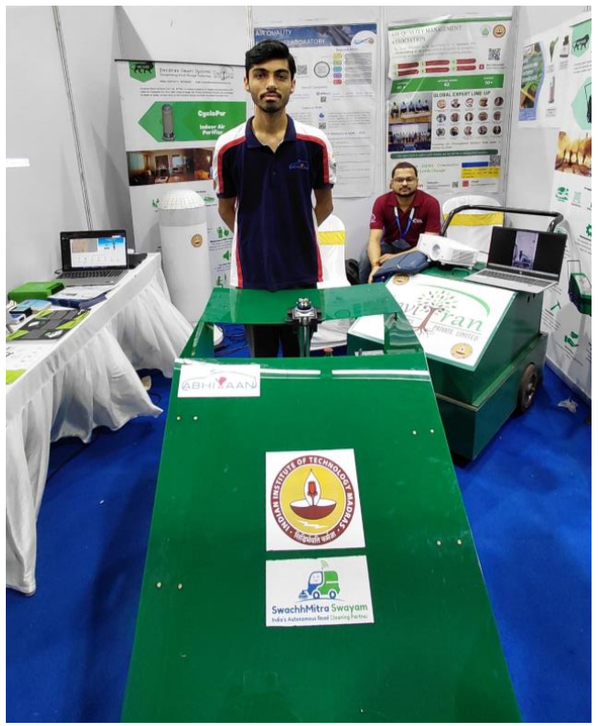
Images



Autonomous Chassis



Cleaning Module Installation



IOH Presentation

Thank You!
QnA

TEAM ABHIYAAN

CENTRE FOR INNOVATION