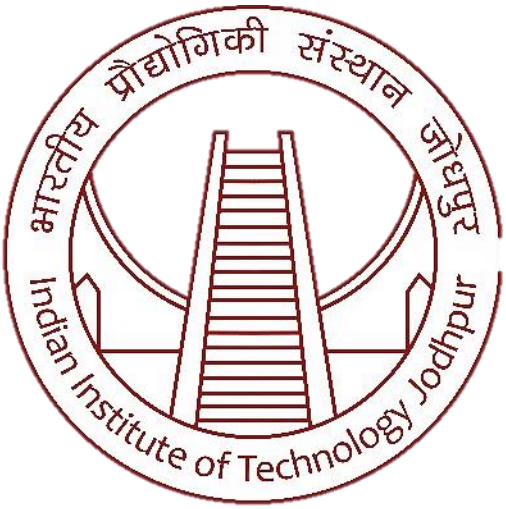
CSL 7650

AUTONOMOUS SYSTEMS



ASSIGNMENT-3

PHASE -1

Submitted by: -

ARASHDEEP SINGH

M23IRM003

Project : Virtual Campus Guide Robot

Team : I am doing this project alone.

Tools Used:-

Robotics Operating System(ROS1)

Gazebo Simulator

Objectives:

- To design and implement an autonomous guide robot that can navigate a campus environment.
- To enable the robot to learn and map the environment autonomously without prior knowledge.
- To utilize advanced pathfinding algorithms to ensure efficient navigation.

Methodologies:

- **Environment Learning**: Implement learning algorithms that allow the robot to explore and learn the layout of the campus environment dynamically.
- **Pathfinding**: Use shortest path algorithms, backward-forward search, and value iteration to find the most efficient routes.
- **Simulation and Testing**: Employ ROS (Robot Operating System) and Gazebo for simulation purposes to test and refine the robot's capabilities in a controlled virtual environment.

Expected Outcomes(Use Cases):

- A fully autonomous guide robot capable of navigating the campus with minimal human intervention.
- A robust system that can adapt to changes in the environment and still find the optimal path.
- Robot exhibiting reactive and active behaviors.

Courses Followed:

Course | Hello (Real) World with ROS – Robot Operating System | edX

Course | Autonomous Mobile Robots | edX

Environment Built:-

The xml file for environment is shared as separate file.

Gazebo Simulator is integrated with ROS1 to develop environment. Here is the example of launch file used for launching the world .

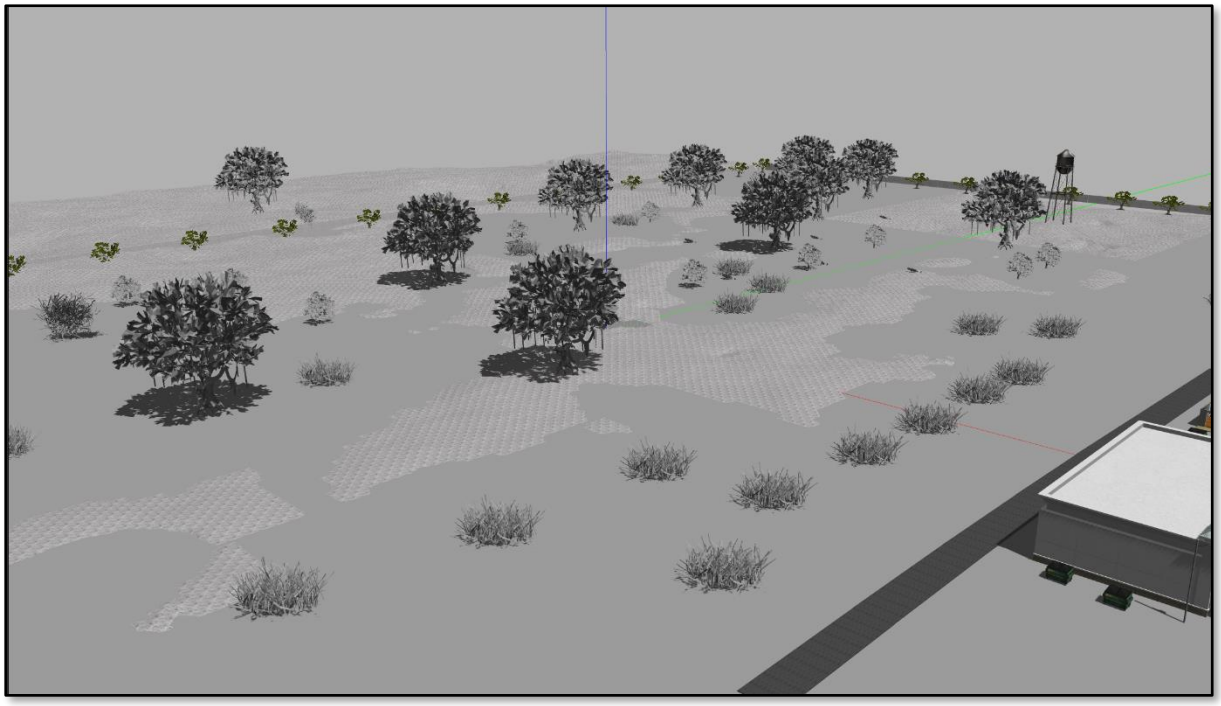
```
<launch>
<!-- these are the arguments you can pass this launch file, for
example paused:=true -->
        <arg name="paused" default="false"/>
        <arg name="use_sim_time" default="true"/>
        <arg name="gui" default="true"/>
        <arg name="headless" default="false"/>
        <arg name="debug" default="false"/>
<include file="$(find gazebo_ros)/launch/empty_world.launch"> <!-- CHANGE HERE  -->
        <arg name="debug" value="$(arg debug)" />
        <arg name="gui" value="$(arg gui)" />
        <arg name="paused" value="$(arg paused)"/>
        <arg name="use_sim_time" value="$(arg use_sim_time)"/>
        <arg name="headless" value="$(arg headless)"/>
        <arg name="world_name" value="$(find task_1)/world/sahiwala.world"/> <!-- CHANGE HERE  -->
</include>
</launch>
```

Details about Environment:-

I have built a campus with three main sections. One is academic section. Second is shopping and residential area. The third is the entry point of the campus. I will make this environment interactive using URDF(Unified Robot Description Format). The agent itself also will take actions using the urdf file. Also an inaccessible region is created in environment.

** Less colours are used so that simulation remains lite as my main focus is on agent ‘s behaviour not on how environment looks.

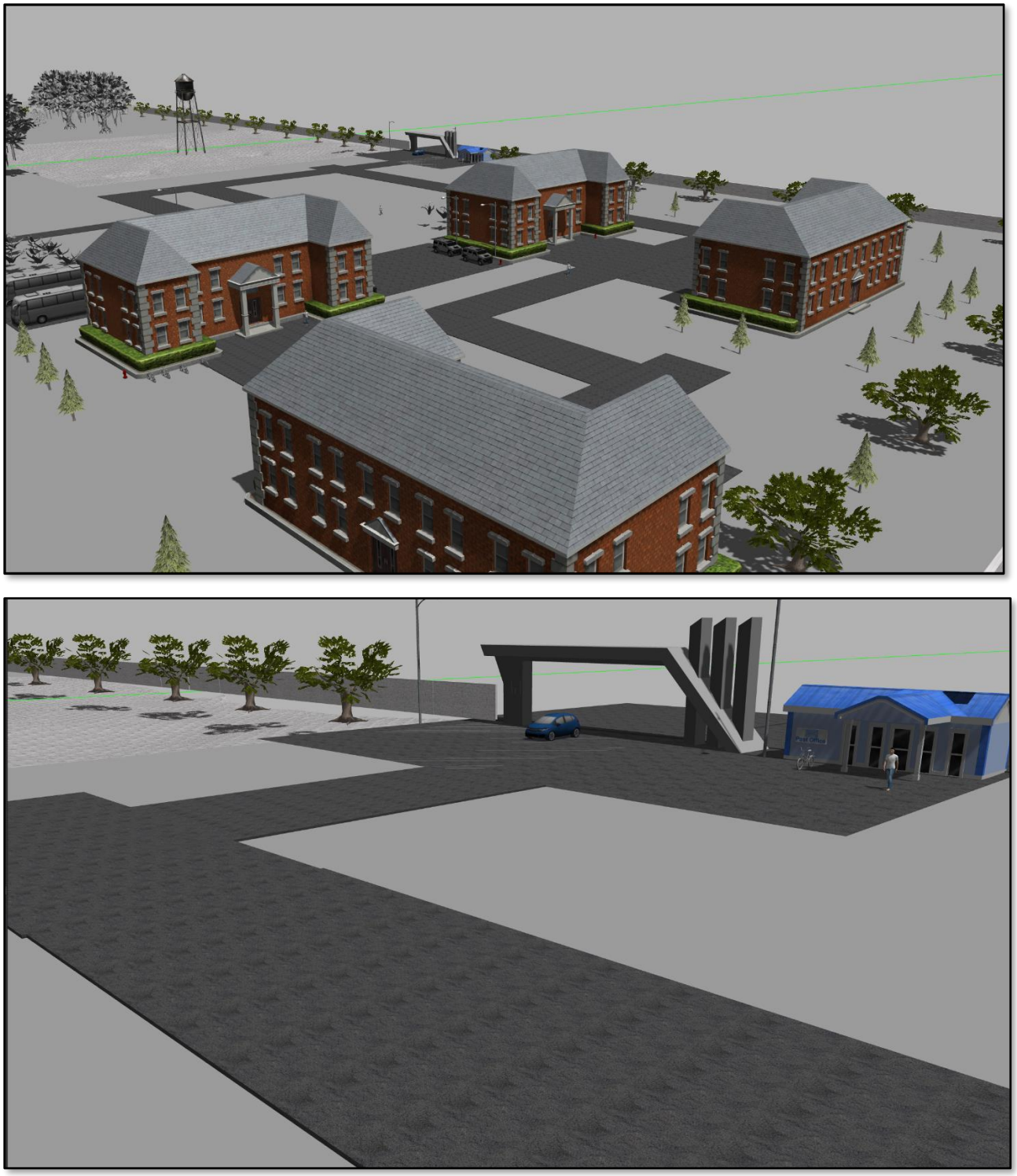
** Remarks :- As environment is built using the availble models on gazebo repository, there is chance that other students may have some similar models.



In accessbile region of Environment

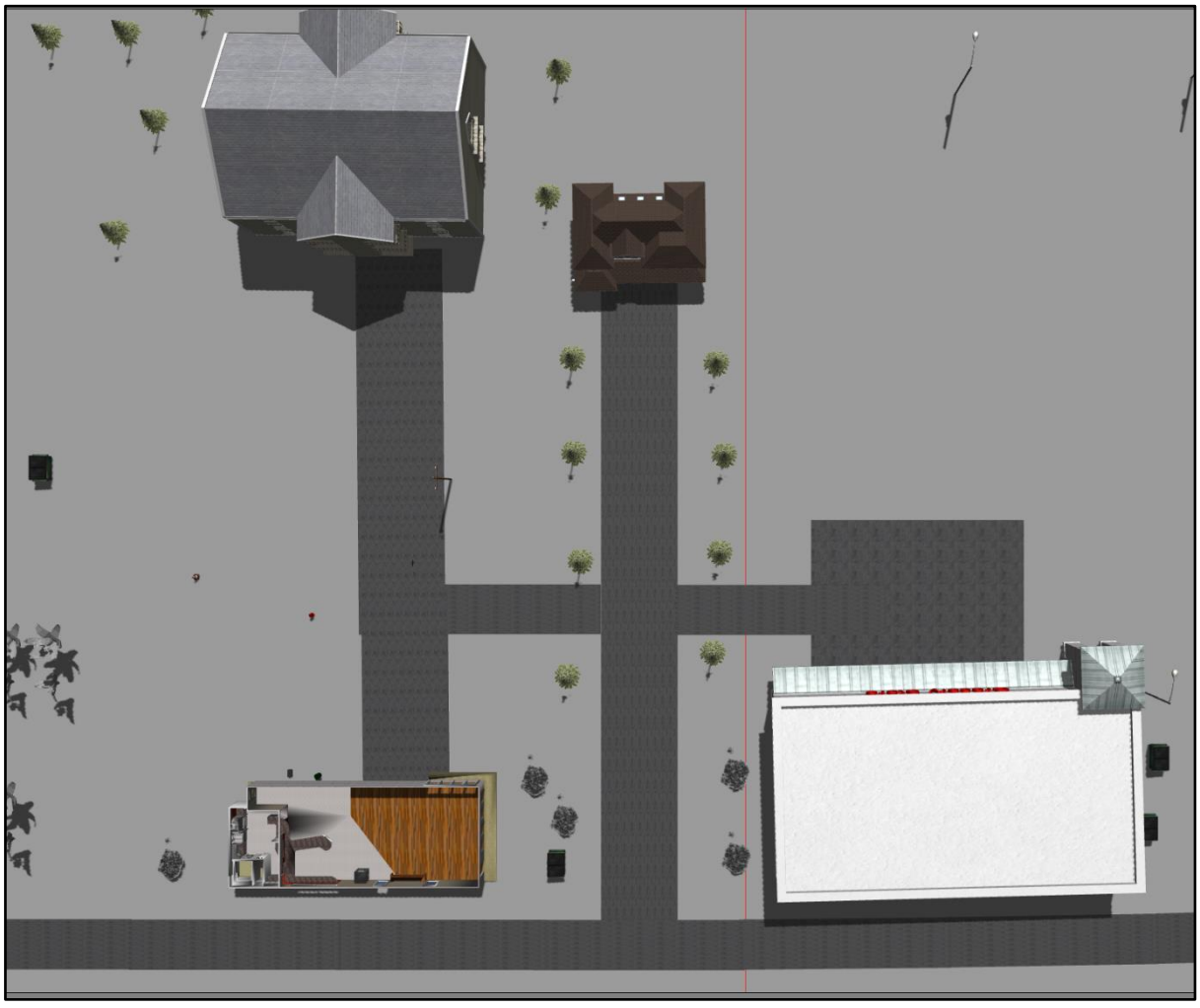
Top View of the Environment

Academic building and Library with Parking



Main Gate with Post office

Academic Buildings



Shopping and Residential Area

Future Developments:-

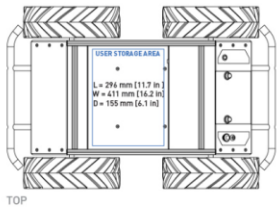
1. **Mapping and Localization**:
    - Implement SLAM (Simultaneous Localization and Mapping) techniques to create a map of the environment.
    - Use sensors (e.g., LIDAR, cameras) to build the map while the robot moves around.
2. **Path Planning and Navigation**:
    - Define a global path (waypoints) for the robot to follow.
    - Use A (A-star)*, or RRT (Rapidly Exploring Random Trees) algorithms to plan the path.
    - Implement local obstacle avoidance using sensor data (e.g., LIDAR) to adjust the robot's trajectory in real-time.
3. **Control and Actuation**:
    - Develop control algorithms to move the robot based on the planned path.
    - Use ROS controllers (e.g., PID controllers) to actuate the robot's motors and achieve desired velocities.
4. **Integration**:
    - Combine mapping, localization, path planning, and control into an integrated system.
    - Test the robot's navigation in the simulated environment.

**A suitable learning approach will be implemented as needed.

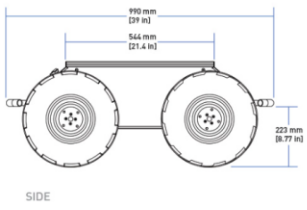
Agent Chosen:-

I have chosen a real-life robot as my agent. The robot is Husky by Clearpath Robotics. I will implement this as my virtual agent so that it becomes easy when there is on-field implementation.

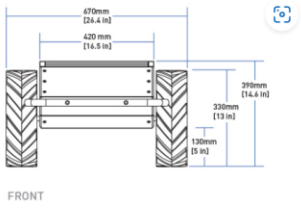
TECH SPECS



TOP



SIDE



FRONT

SIZE AND WEIGHT

EXTERNAL DIMENSIONS	990 x 670 x 390 mm (39 x 26.4 x 14.6 in)
INTERNAL DIMENSIONS	296 x 411 x 155 mm (11.7 x 16.2 x 6.1 in)
WEIGHT	50 kg (110 lbs)
MAX PAYLOAD	75 kg (165 lbs)

SPEED AND PERFORMANCE

MAX SPEED	1.0 m/s (2.2 mph)
RUN TIME (TYPICAL USE)	3 hours
USER POWER	5V, 12V and 24V fused at 5A each
DRIVERS AND APIS	ROS Melodic, ROS Kinetic, C++ Library, Mathworks

I will use this [husky/husky_gazebo/package.xml at noetic-devel · husky/husky · GitHub] repository for my reference.