

# Lidar-Based Exploration using Sampling-Based Planning

Hands-on Planning Project 2025

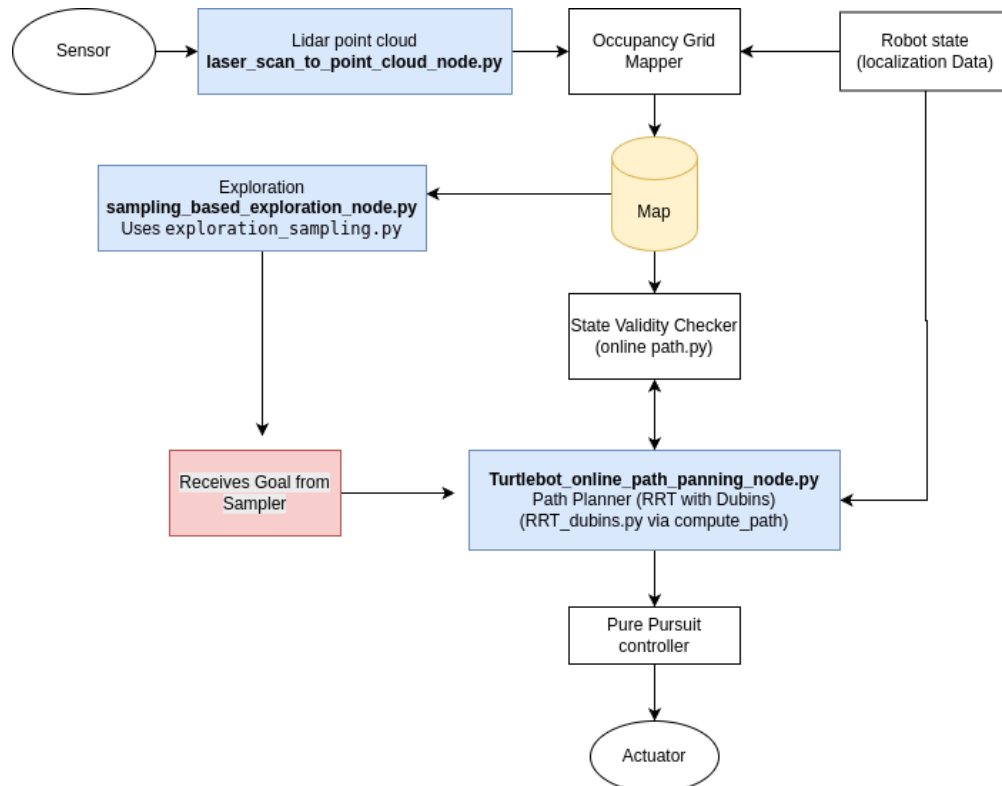
## 1. Environment Setup

- **Operating System:** Ubuntu 20.04
- **ROS Version:** ROS Noetic
- **Simulator:** Stonefish Turtlebot Simulator

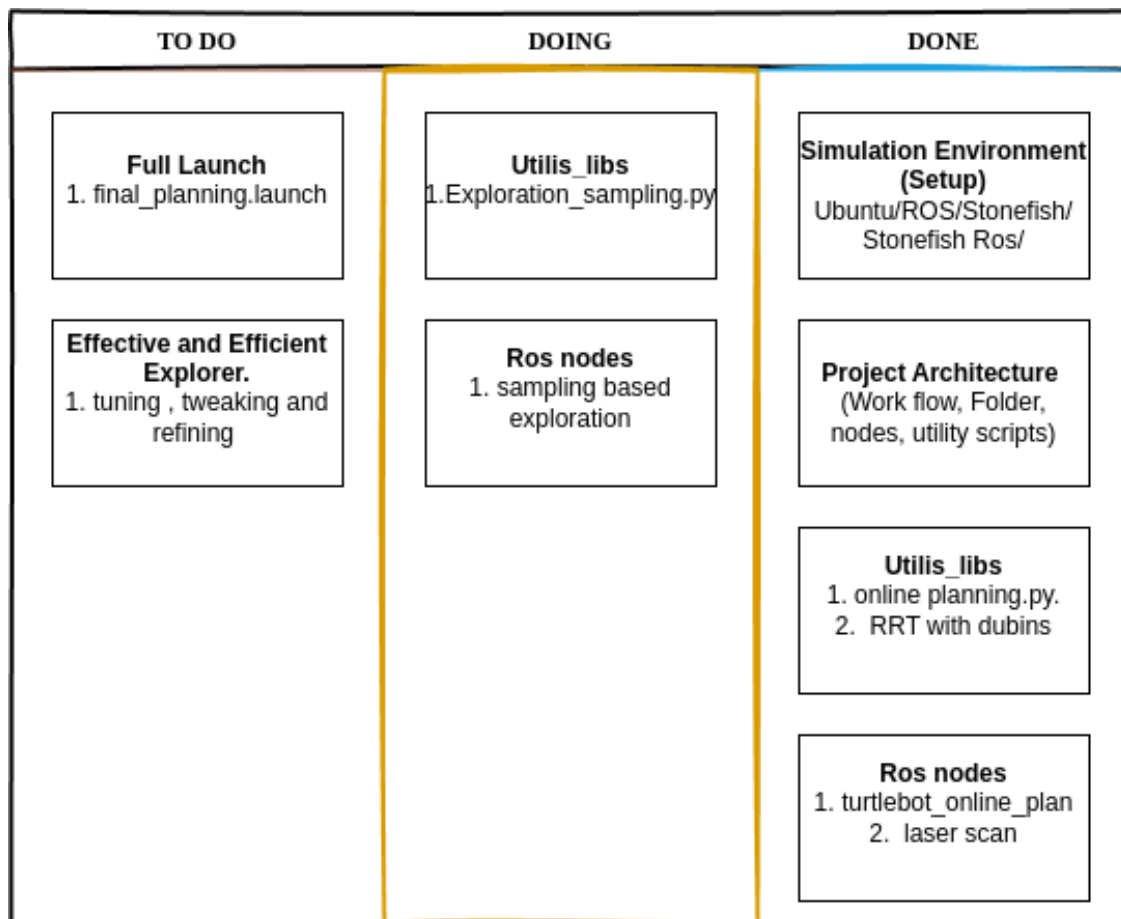
## 2. Project Architecture

- Real-time 2D mapping with OctoMap.
- Sampling-based goal generation.
- Path planning using an RRT\* planner with Dubins curves.
- Motion execution using a Pure pursuit controller.

### Workflow Diagram



## Kanban Board



## 3. Completed Modules

### 3.1 laser\_scan\_to\_point\_cloud\_node.py

Converts Lidar data from the RPLidar sensor to 3D PointCloud2 messages

### 3.2 turtlebot\_online\_path\_planning\_node.py

This is the main online planning and control node. It:

- Listens to occupancy grid, odometry, and goal topics.
- Uses a State Validity Checker to assess obstacles.
- Plans paths using RRT\* with Dubins motion primitives.
- Executes the path with pure pursuit controller.
- Publishes visualizations to RViz for path, and trajectory

### 3.3 online\_planning.py

This module defines:

- `StateValidityChecker`: Converts map coordinates, validates points/paths.
- `compute_path()`: Wraps RRT planner to produce smooth paths.
- `pure_p_control()`: A simple velocity controller based on pursuit logic.

### 3.4 RRT\_dubins.py

Implements an RRT\* planner using Dubins paths:

- Node and tree representation.
- Path smoothing and rewiring (RRT\*).
- Validity checking with Dubins path segments

## 4. Work in Progress

### 4.1 exploration\_sampling.py

Responsible for generating sampling-based exploration goals. using information gain heuristics.

### 4.2 sampling\_based\_exploration\_node.py

This ROS node will:

- Periodically call `exploration_sampling` to select new goals.
- Send those goals to the planner for execution.

## 5. Directory Structure

```
catkin_ws/src/lidar_based_exploration/  
Blank diagram.png  
CMakeLists.txt  
config  
Hands-on Planning Project 2025.pdf  
launch/  
    sampling_exploration.launch  
    stonefish.launch  
package.xml  
README.md  
rosgraph.png  
src/  
    laser_scan_to_point_cloud_node.py  
    sampling_based_exploration_node.py  
    turtlebot_online_path_planning_node.py  
    utils_lib/
```

```
exploration_sampling.py  
online_planning.py  
__pycache__/  
RRT_dubins.py
```