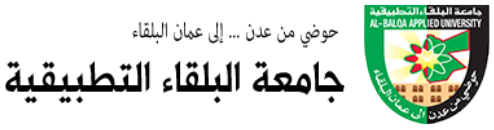


ROS Workshop

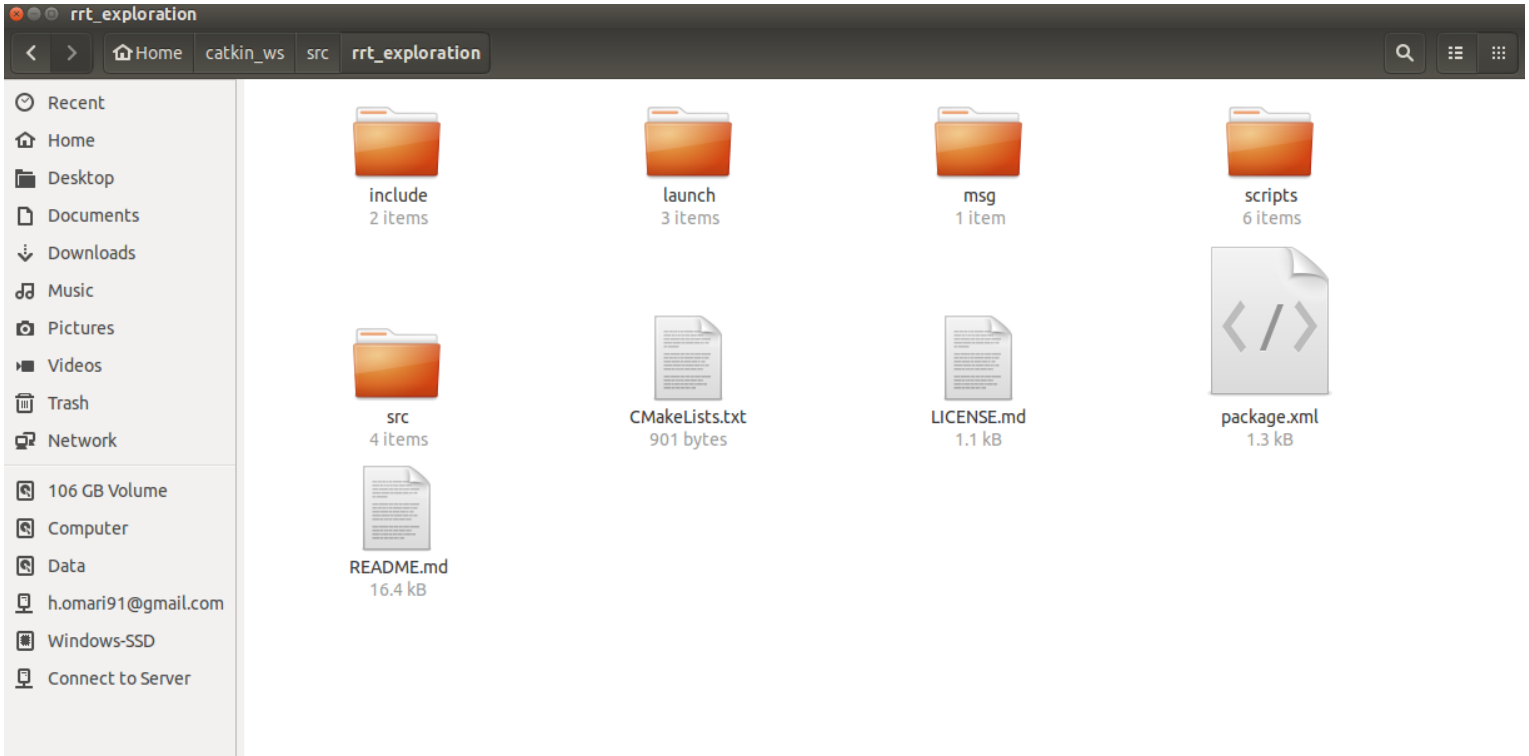
Packages, Tools, and Libraries

January 12, 2023

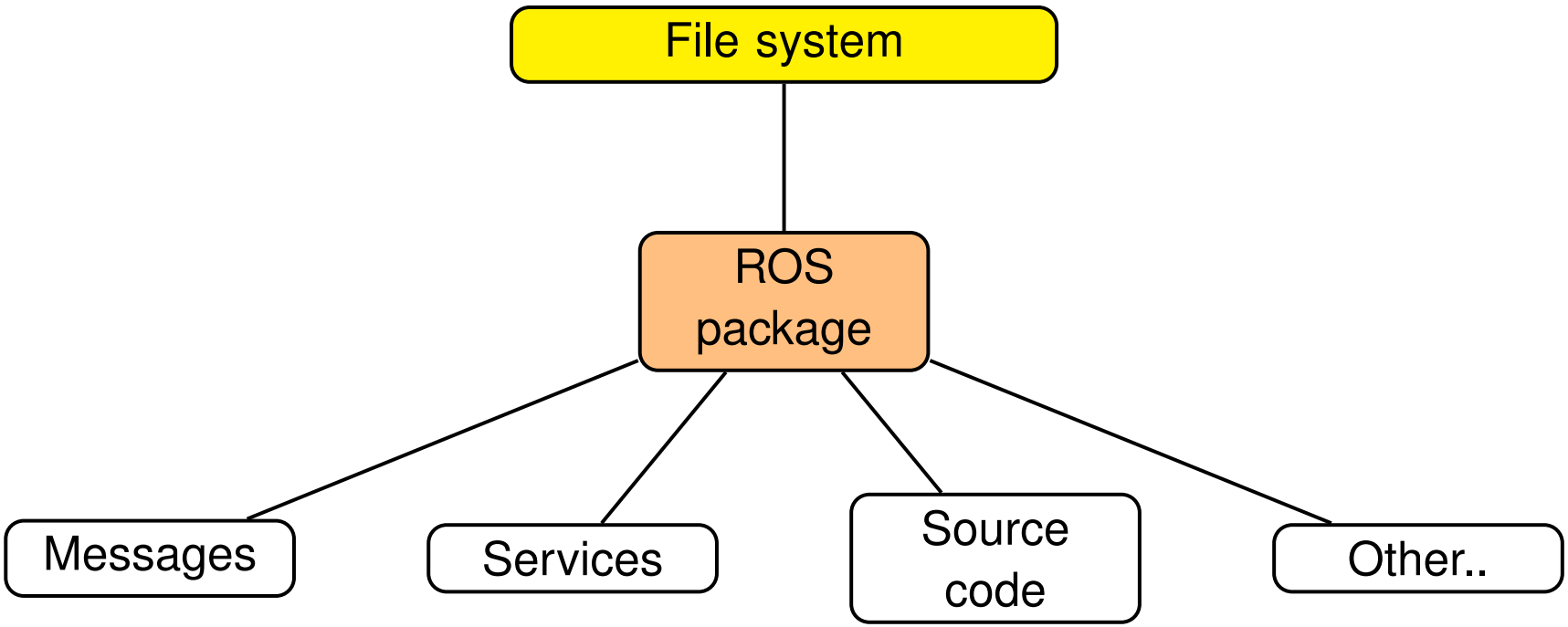
Hassan Umari

ROS Packages

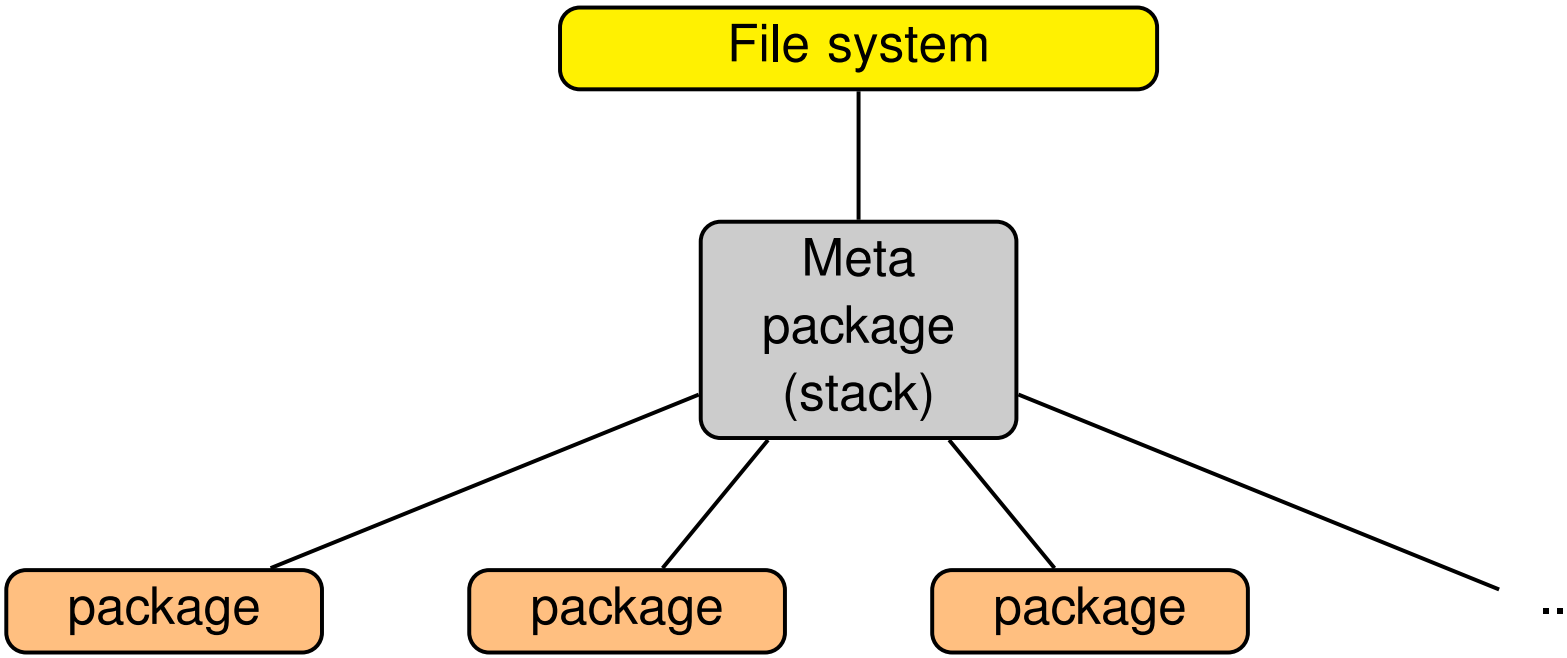
Inside a ROS package:



ROS Packages



ROS Packages



ROS Packages

How to install packages:

- From source (many packages available on GitHub).
- Using **apt**. Example:

```
sudo apt install ros-noetic-navigation
```

Demo

package, rosrun, launch files

Gazebo

- Gazebo is a simulator that is bundled with ROS distributions.
- Gazebo includes a physics engine, 3D rendering (OpenGL), and support for simulating sensors and actuators.
- Simulation environment can be defined in a `.world file`.

Example

RViz

- RViz can be used to **visualize** several commonly used ROS messages
- Examples: laser scans, occupancy gird maps, point clouds, images, robot frames, etc..

RViz

Coordinate Frames

- A robotic system involves several coordinate frames that change over time.
- A common task in robotics, is to find the transformation between frames.
- Visualization of these frames is also very important.

Coordinate Frames

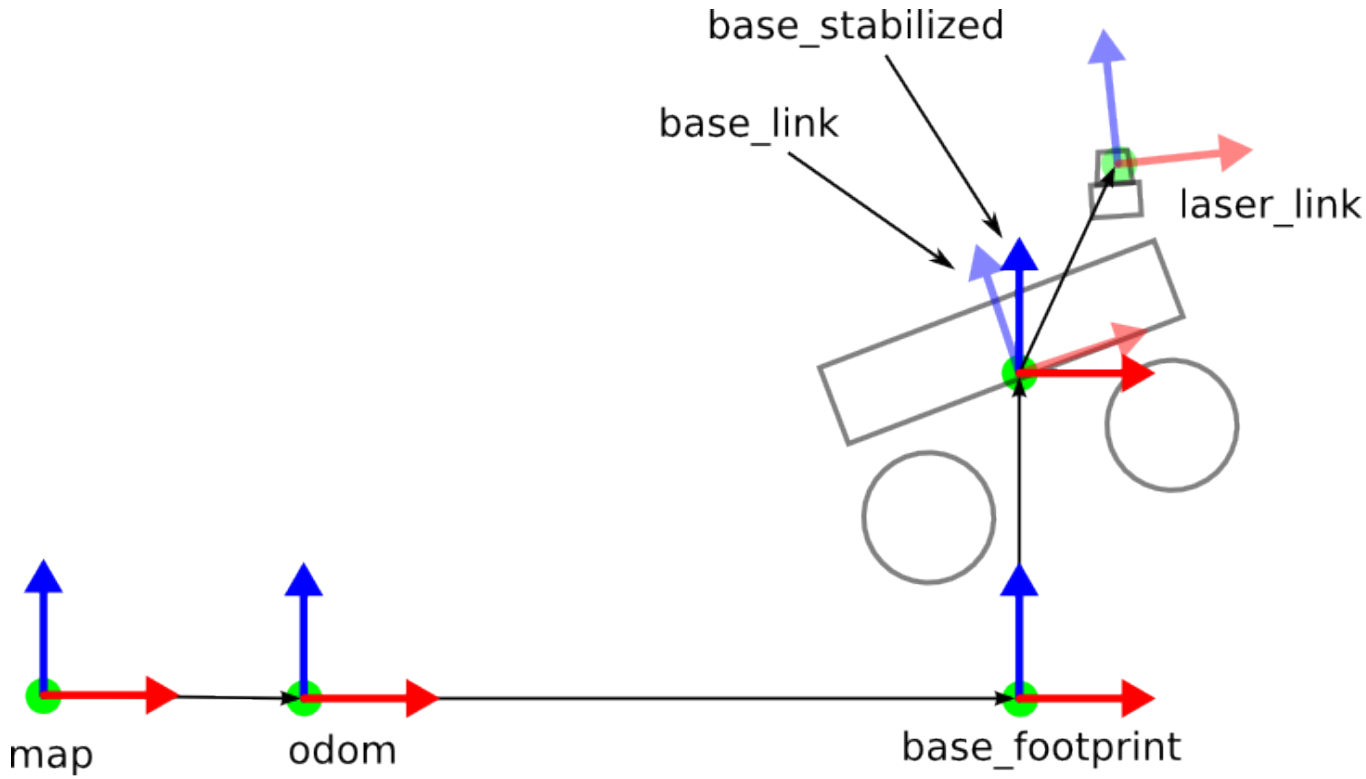
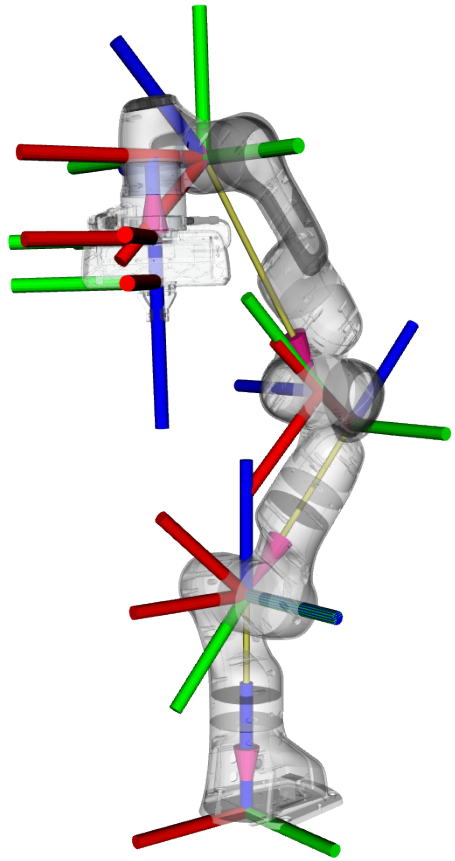


image source: http://wiki.ros.org/hector_slam/Tutorials/SettingUpForYourRobot

Coordinate Frames

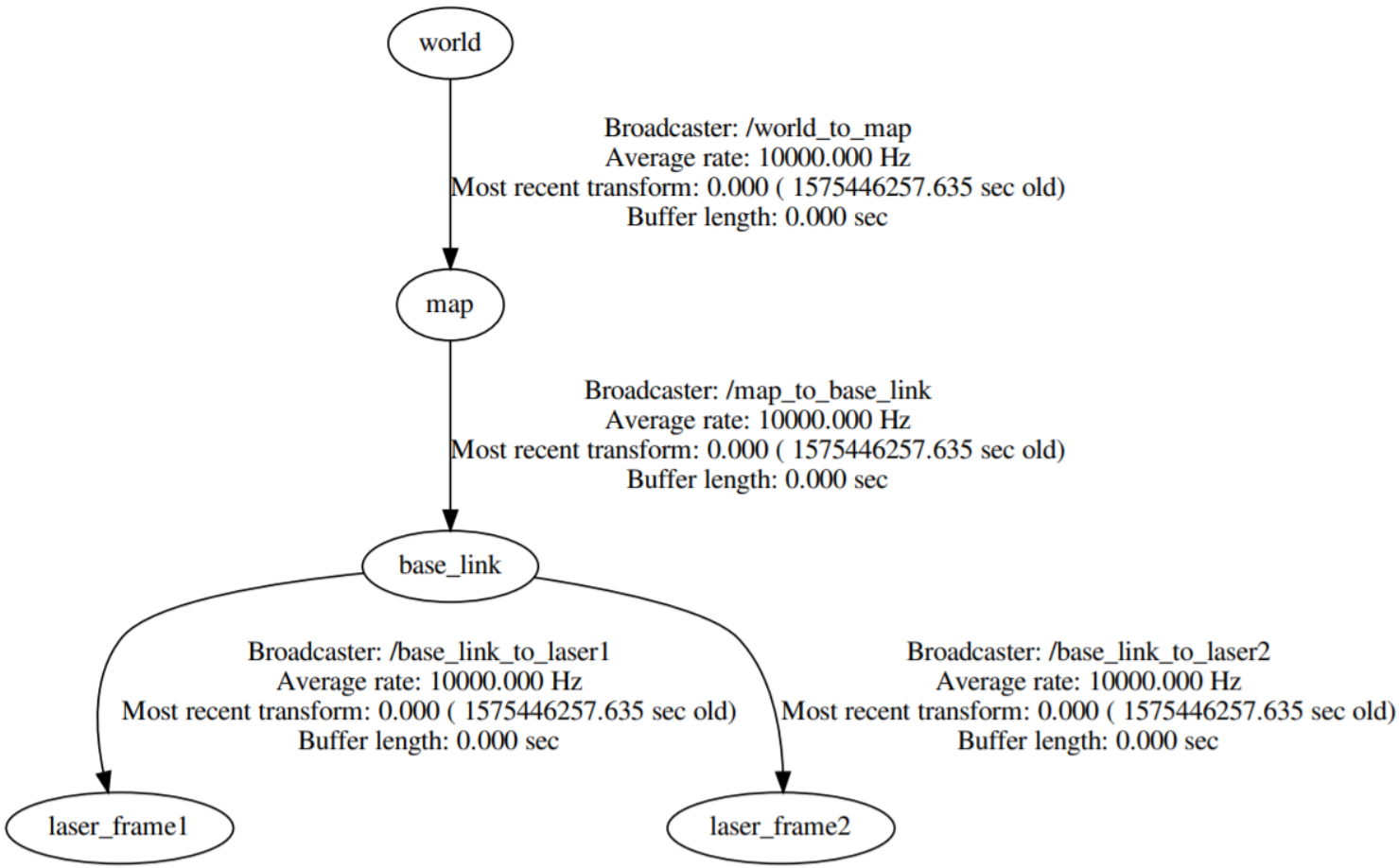


Source: https://ros-planning.github.io/moveit_tutorials/doc/robot_model_and_robot_state/robot_model_and_robot_state_tutorial.html

tf - The Transform Library

- The tf is a library for handling **transformation** between frames.
- The library stores the relationship between frames in a **tree** structure buffered in **time**.
- In ROS, it comes as a package called **tf2**.

tf - The Transform Library



Source: https://spl.hevs.io/spl-docs/tools/ros/tf2.html

tf - The Transform Library

- There is no central server. Each client listens to published transforms and maintains a copy of the tree.
- What a node can do:
  - Broadcast a transformation
  - Listens for all transformations
  - query for a transform at a chosen time instance
- Comes with a tool to quickly publish static transformations.

Demo

URDF

- URDF = Unified Robotics Description Format
- XML format for representing a robot model

URDF

```
<robot>
        <link>
        ...
        </link>
        <link>
        ...
        </link>

        <joint>
        ...
        </joint>

</robot>
```

URDF

- URDF file can be loaded by **robot_state_publisher**
- It will publish link frames to tf

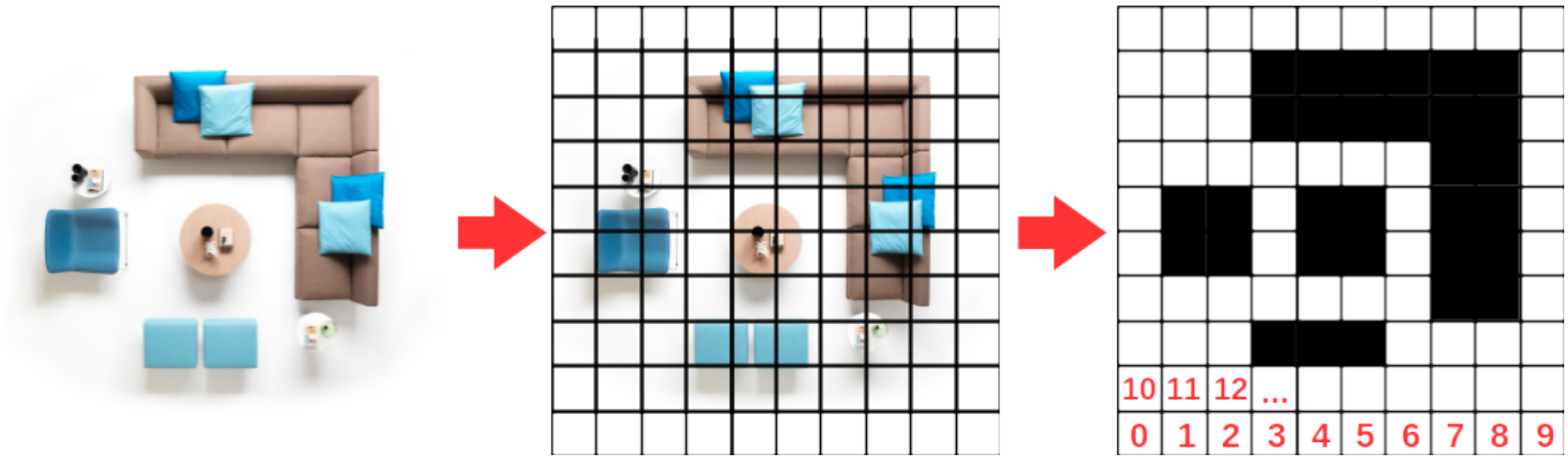
Example

URDF + Robot state publisher

Gmapping

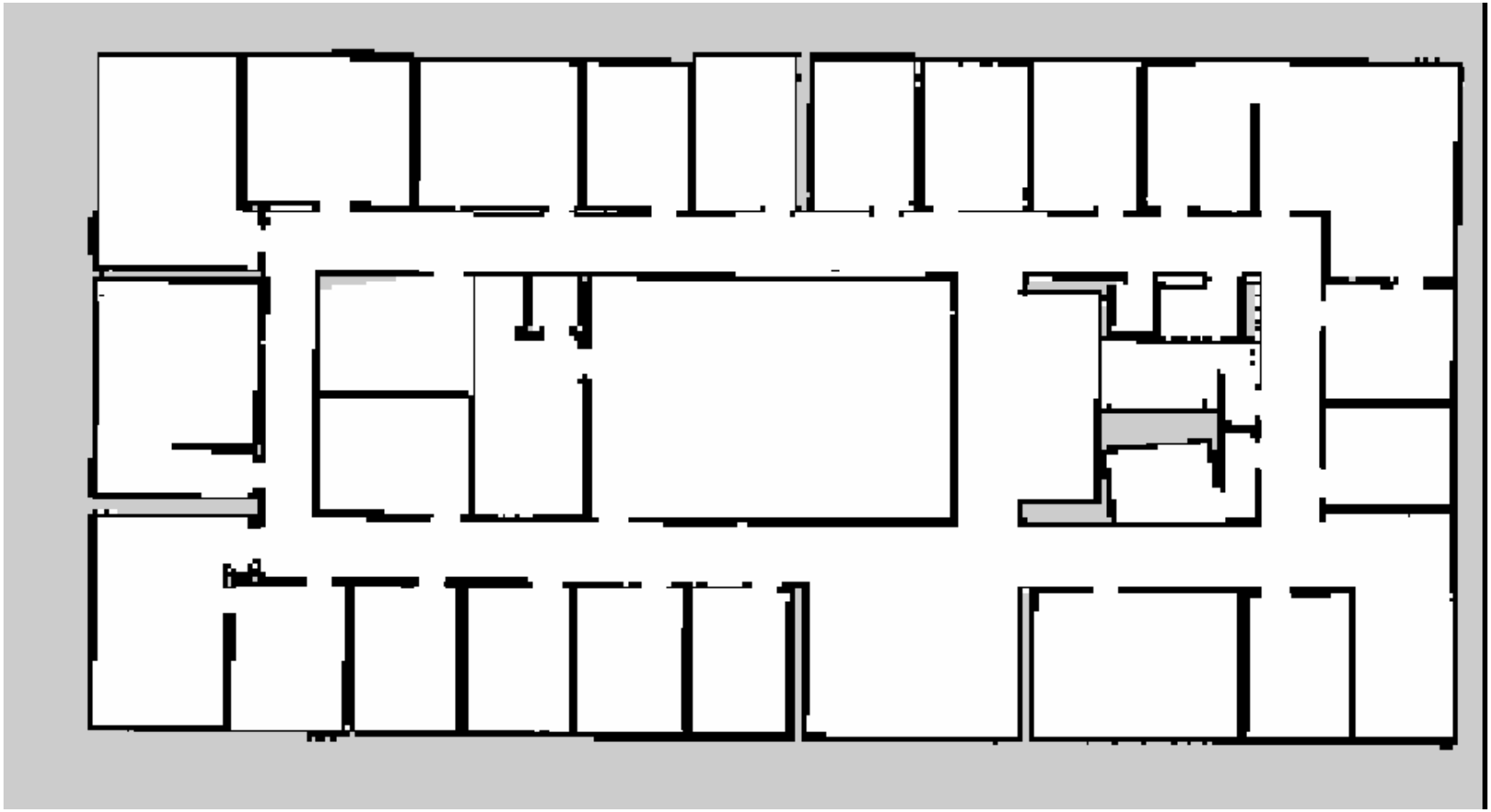
- Is a ROS wrapper for Gmapping C++ library.
- Implements a SLAM algorithm.
- odometry + laser scan $\rightarrow$ occupancy grid map

Occupancy Grids




```
map data = [0, 0, 0, 0, ...  , 0, 100 ,100 ,100 ,0 ,0 , ...  ]
```

Occupancy Grids



Gmapping

```
sudo apt install ros-noetic-gmapping
```

Gmapping

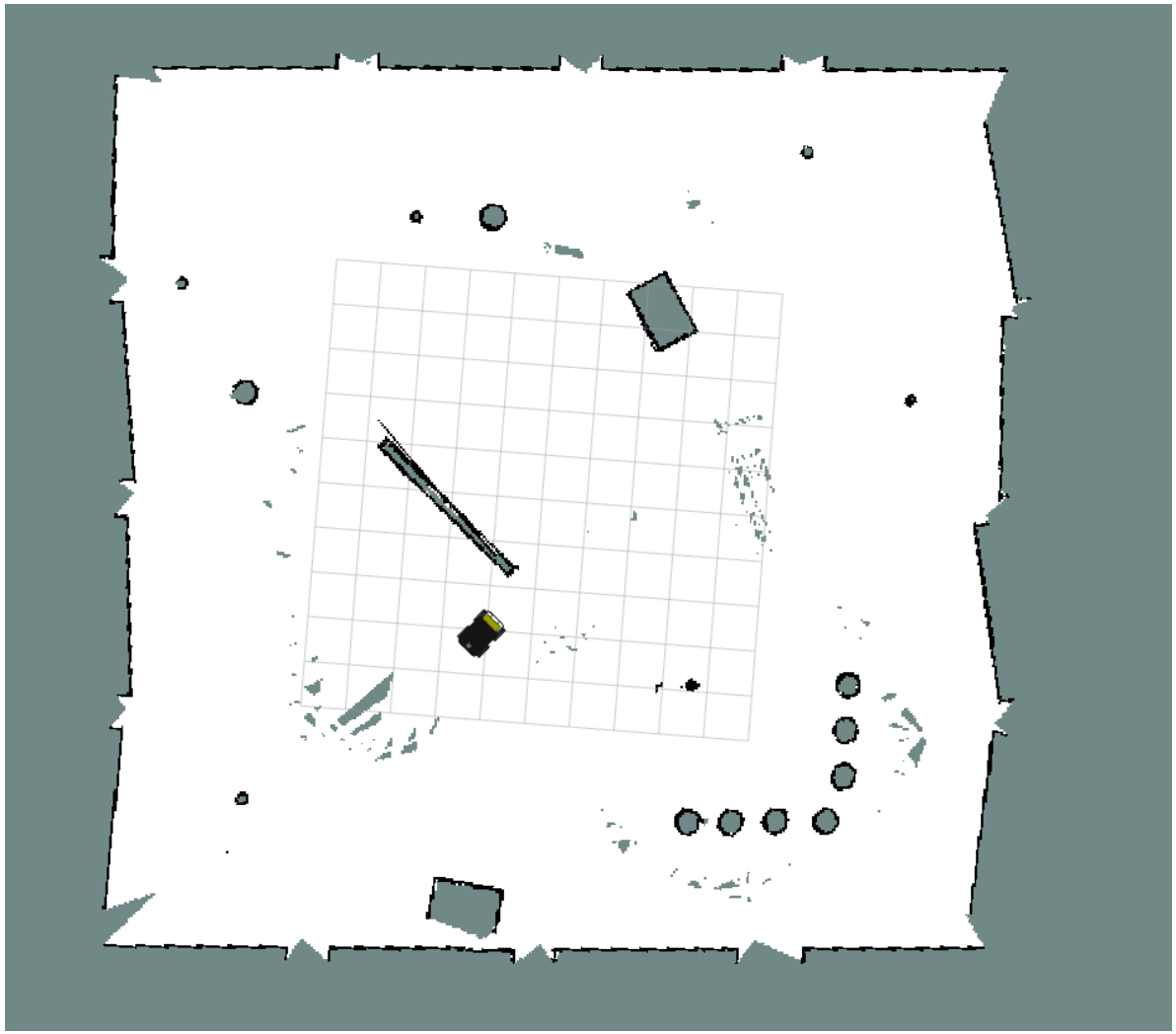
Navigation Stack

- A collection of packages for mobile robot **navigation** (**2D**)
- map + odometry + laser scan + goal $\rightarrow$ velocity commands

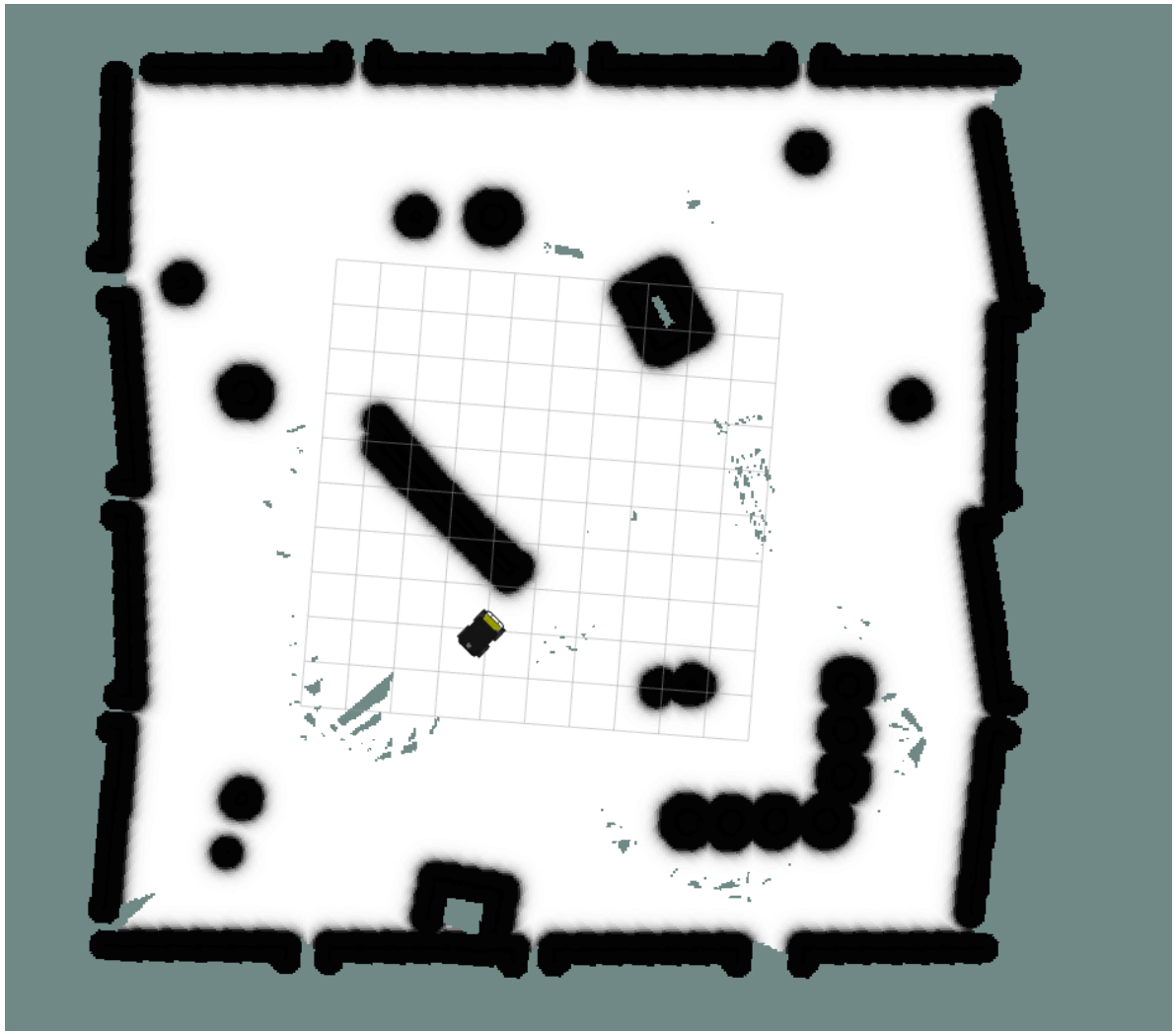
Navigation Stack

- Costmap
- Global planner
- Local planner

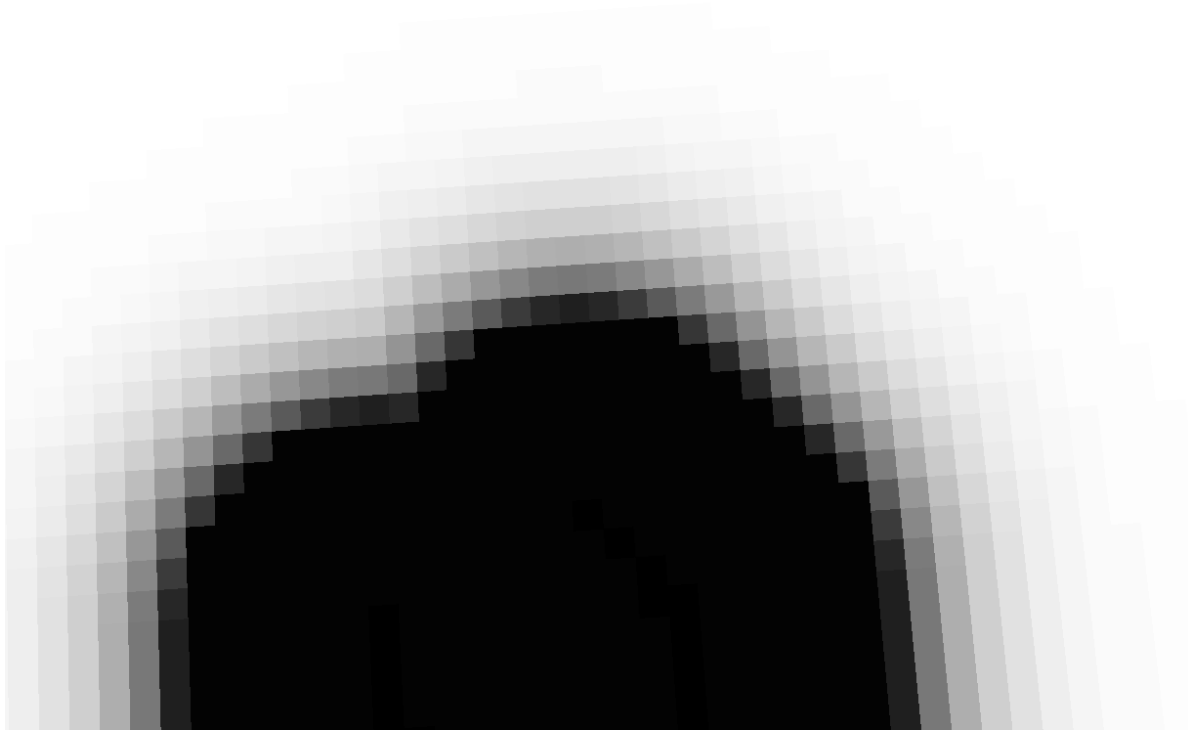
Costmap



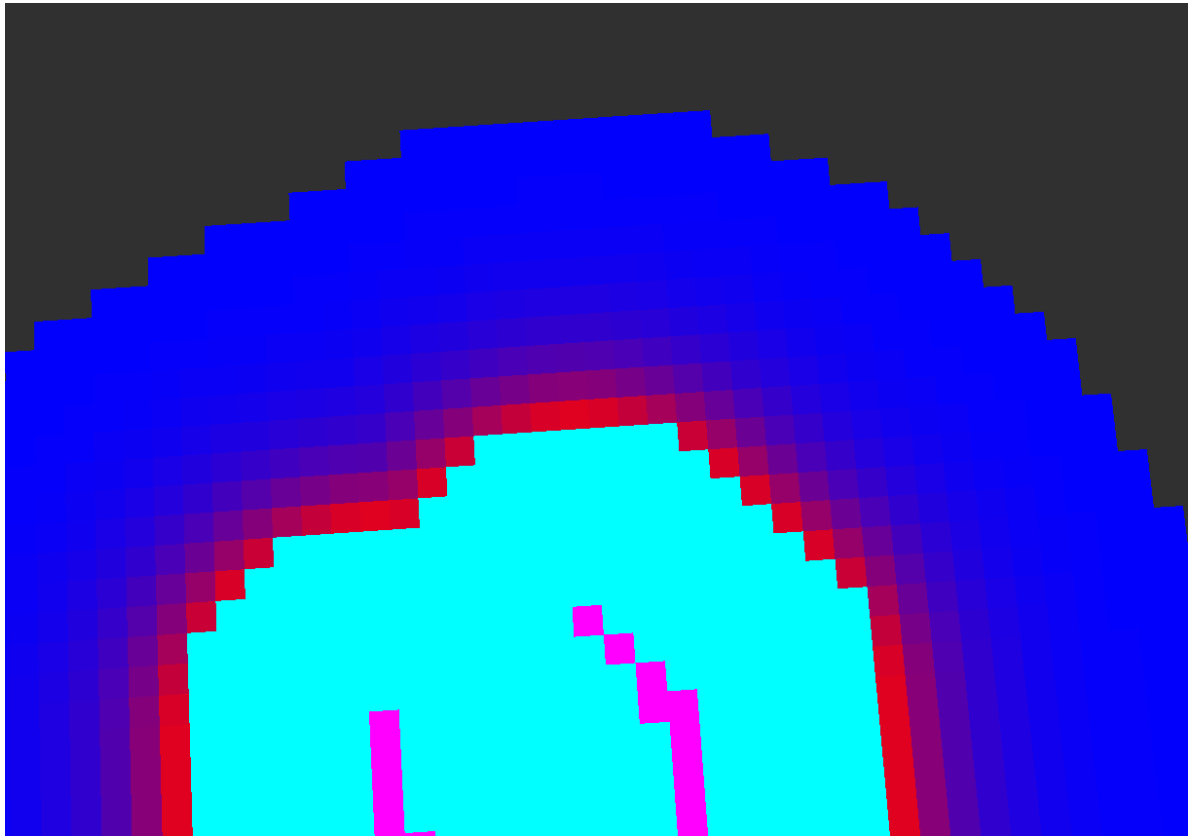
Costmap



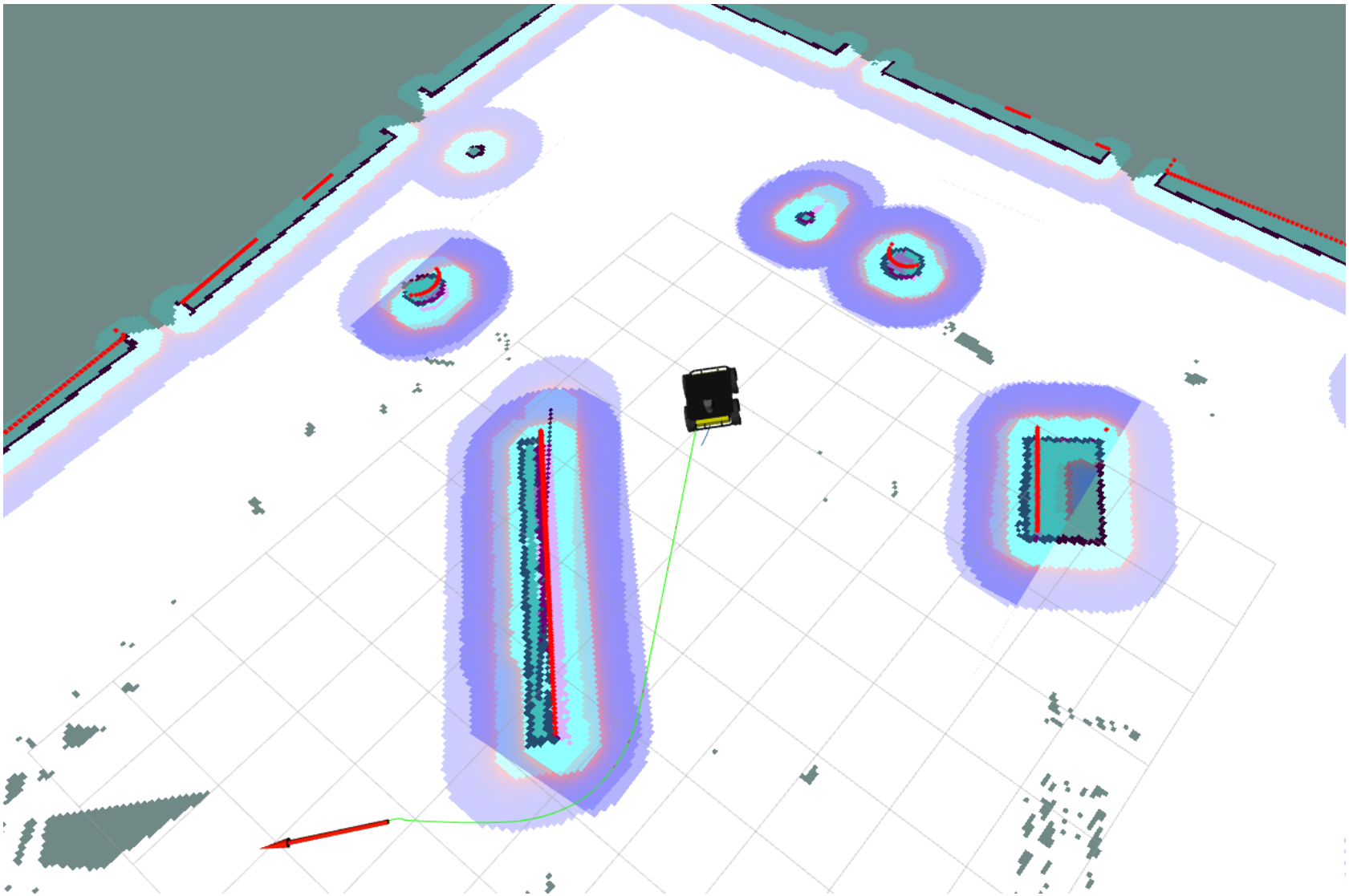
Costmap



Costmap



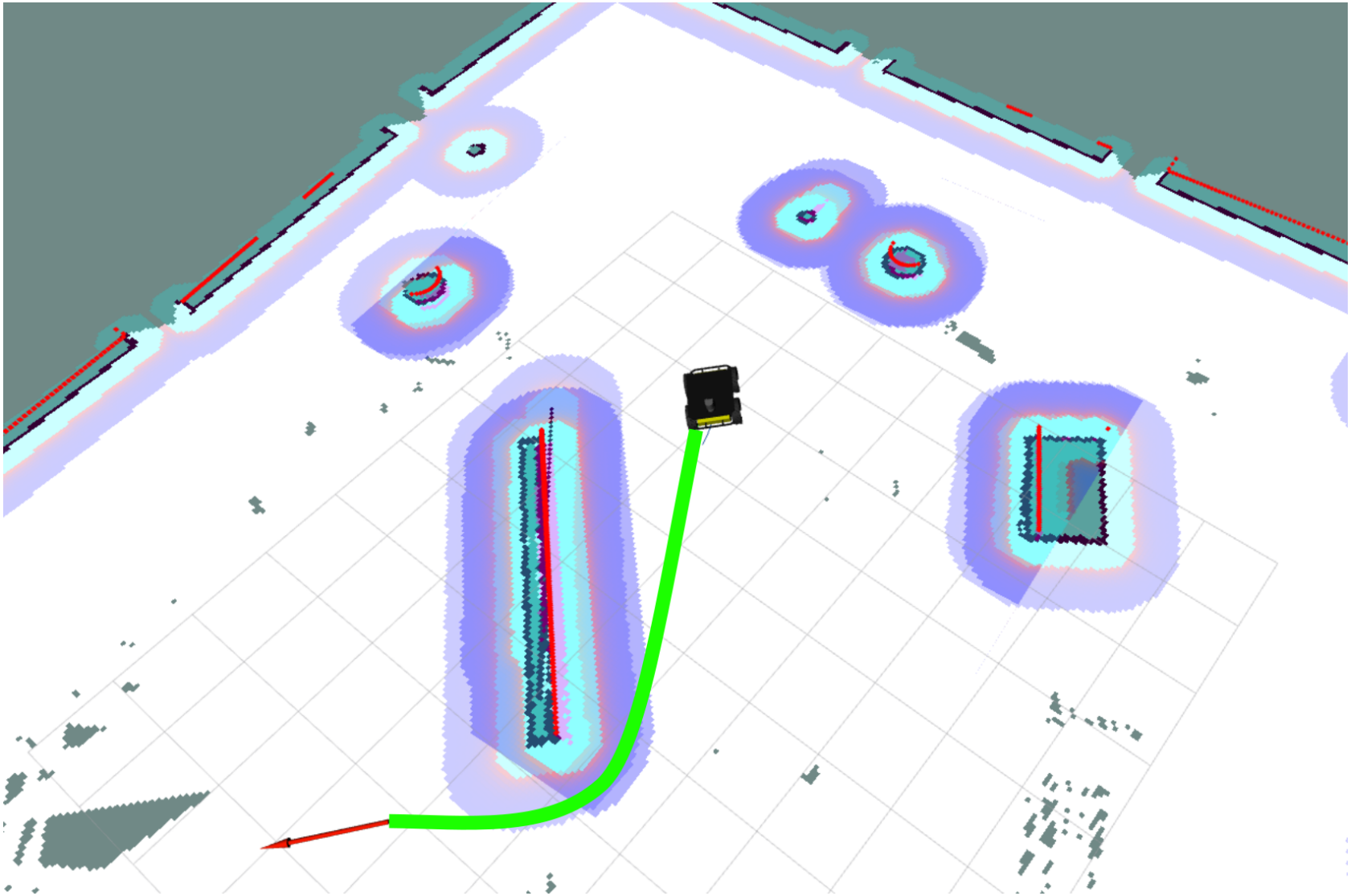
Costmap



Global Planner

- Uses the **global costmap**
- **Search** algorithm to find a safe path in the global costmap

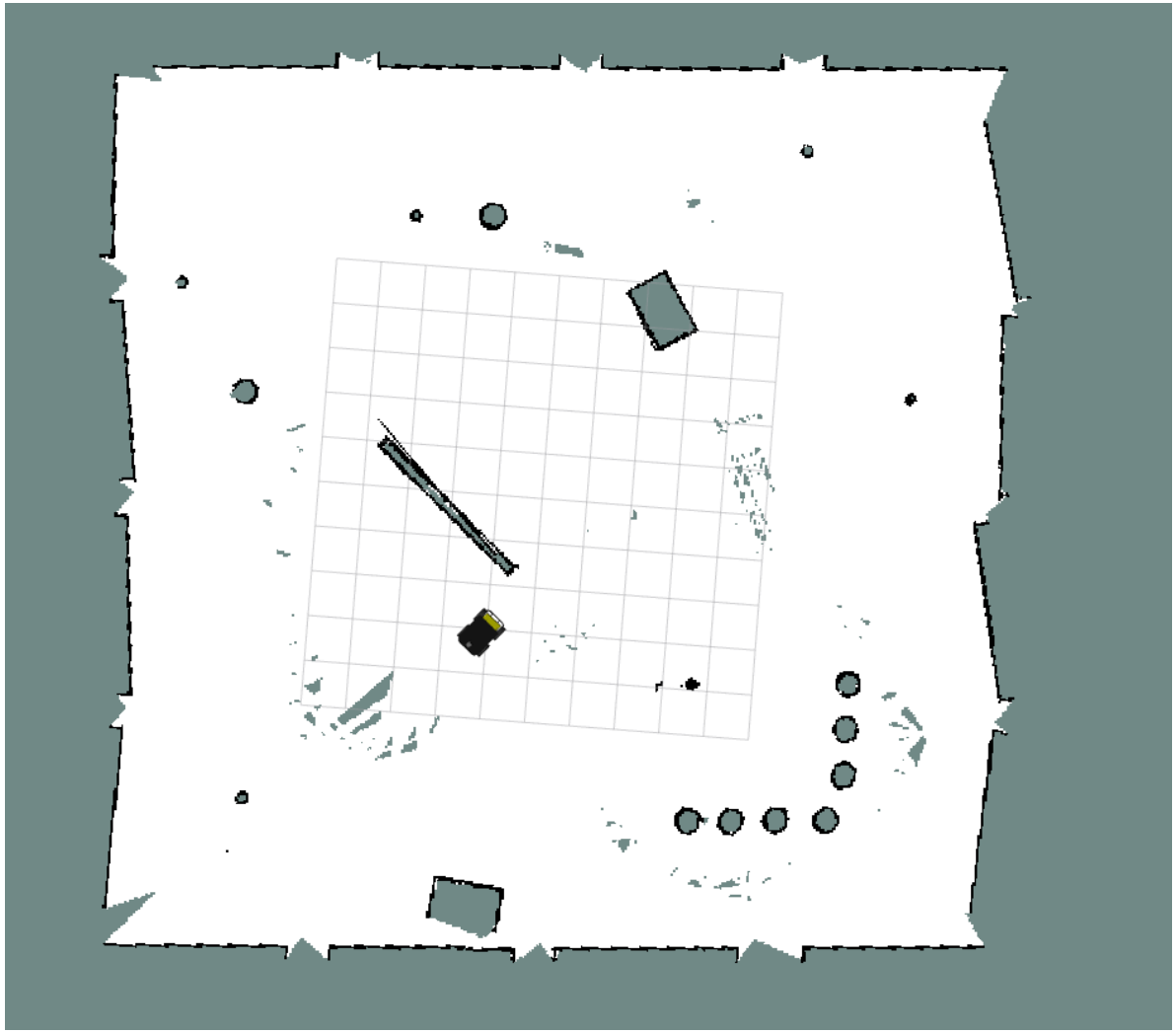
Global Planner



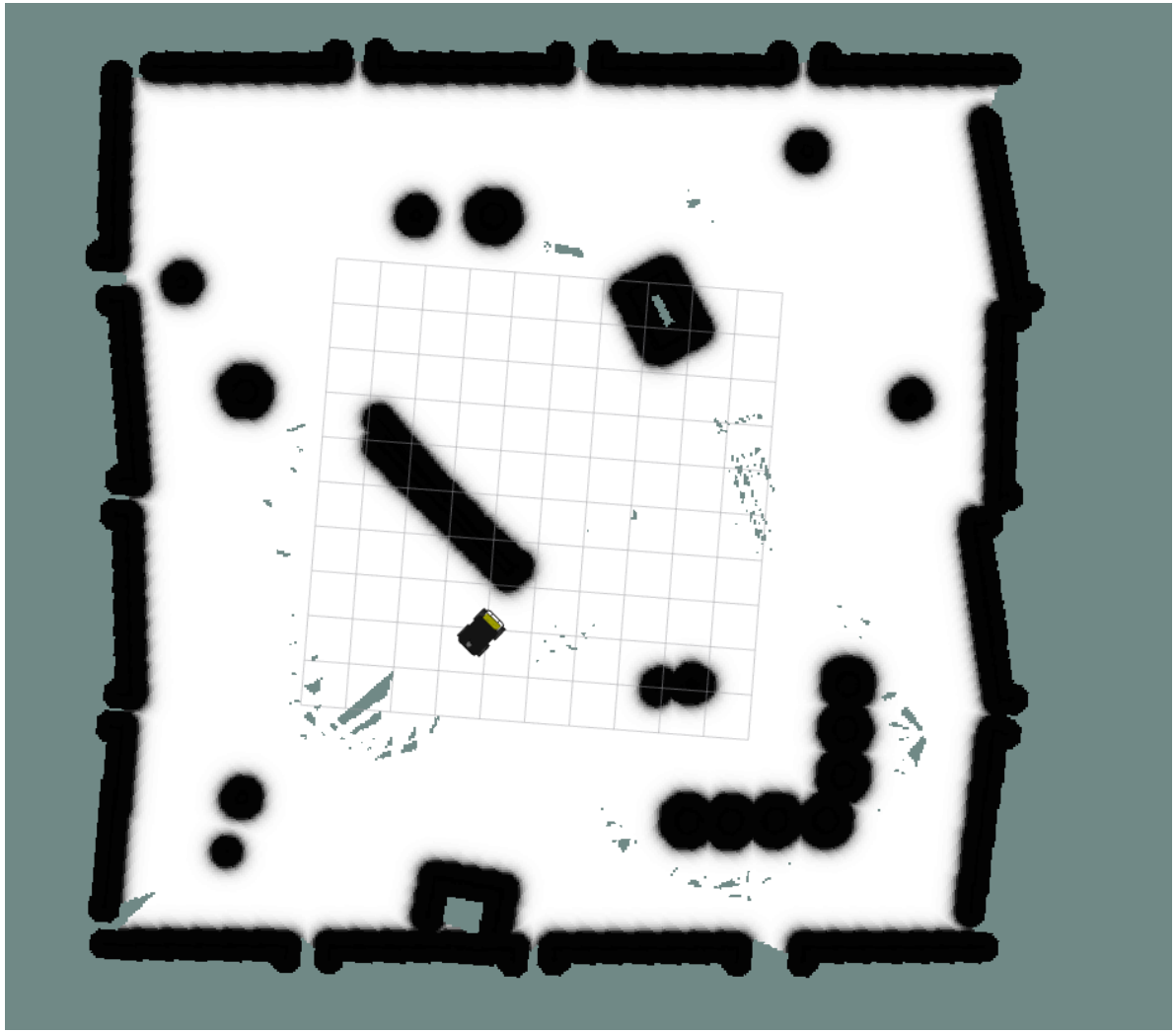
Local Planner

- Generates **velocity commands** to make the robot **follow** the global plan
- It uses a **local costmap**
- A local path is generated based on the global plan

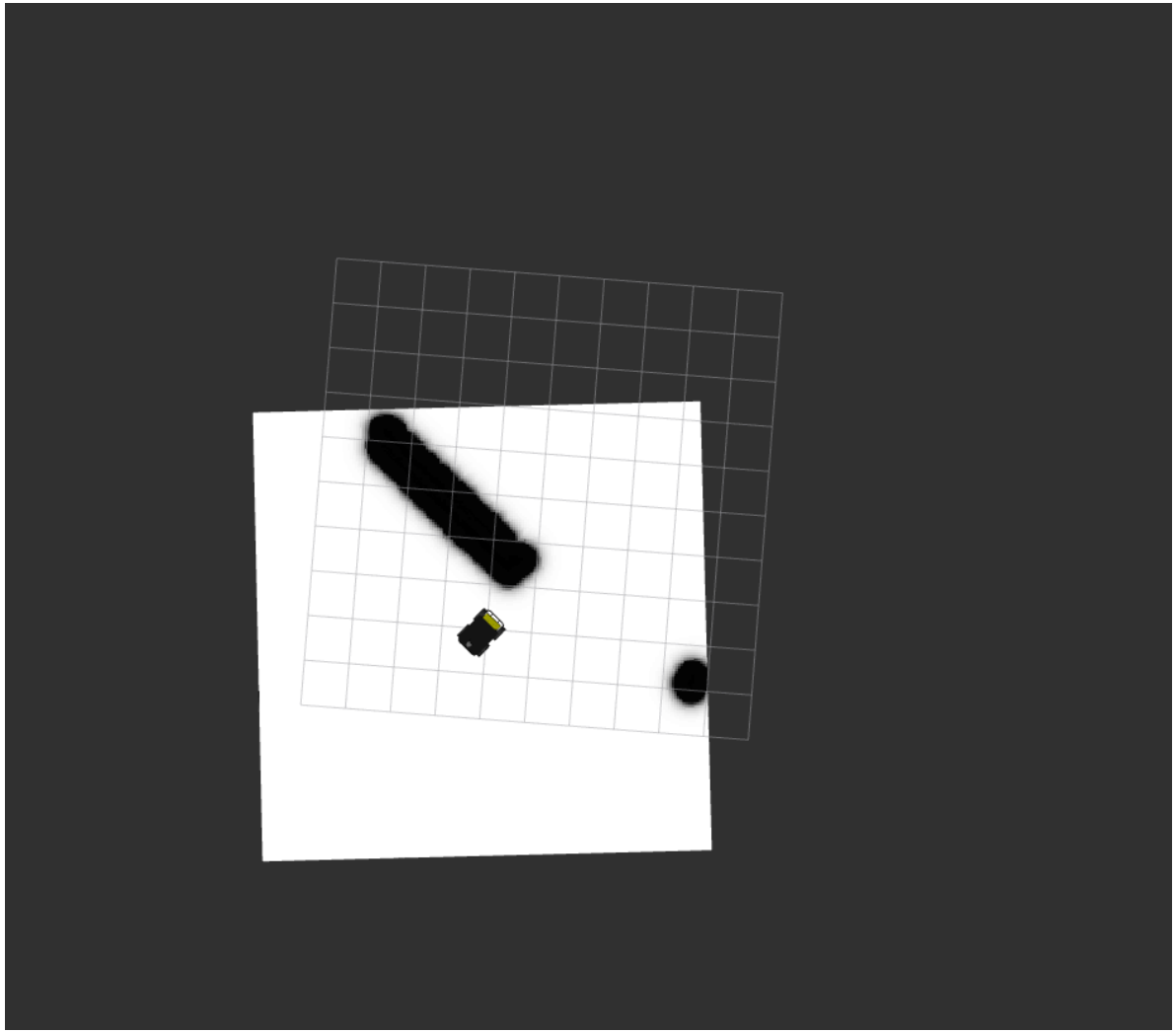
Local Planner



Local Planner



Local Planner



Demo

Thank you
Questions..?