Tutorial 3: Environmental Planning and Robot Navigation

RoboCup@Home Practical Course WS17/18

Institute for Cognitive Systems

Department of Electrical and Computer Engineering

Technische Universität München

Dr. Karinne Ramirez-Amaro

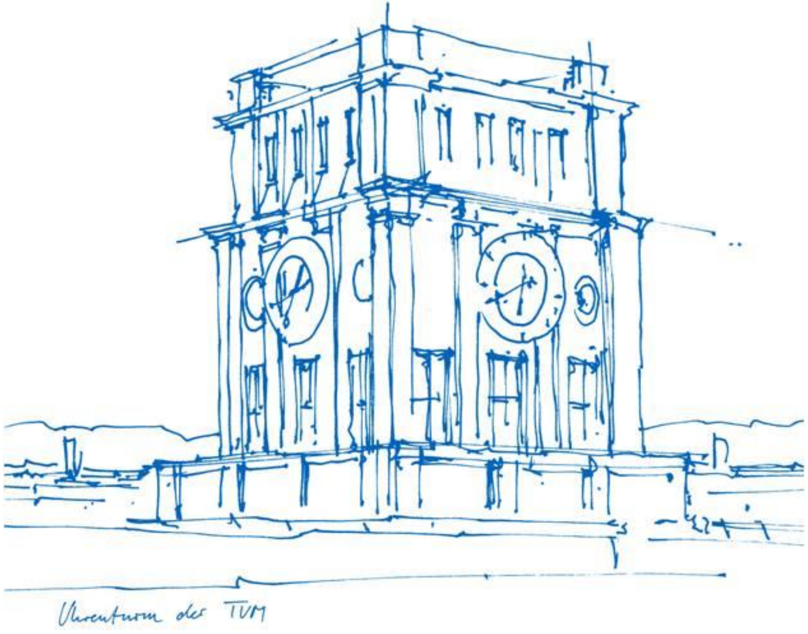
Dr. Emmanuel Dean

Dr. Pablo Lanillos

Rogelio Guadarrama

Constantin Uhde

Prof. Gordon Cheng

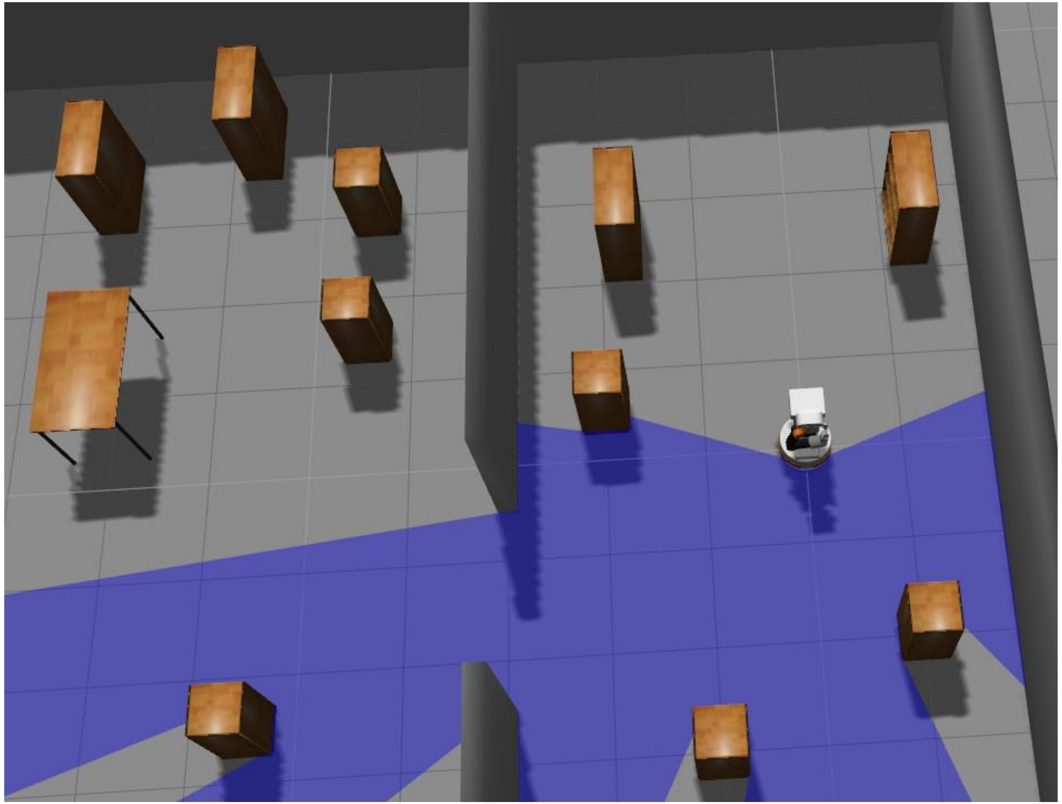


Munich, 15. November 2017

Navigation, Localization, Mapping

Three distinct problems:

- Get from A to B
- Locate the Robot
- Represent the Environment

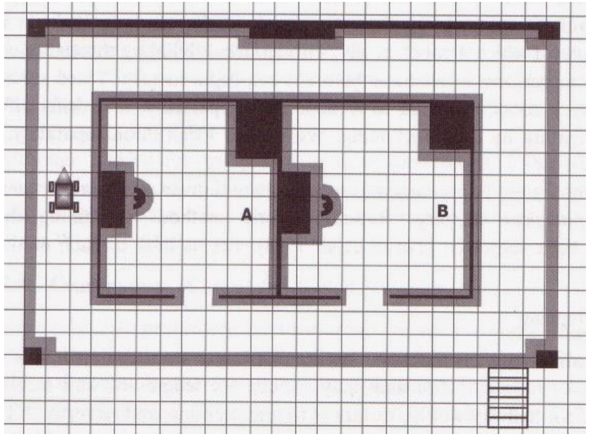


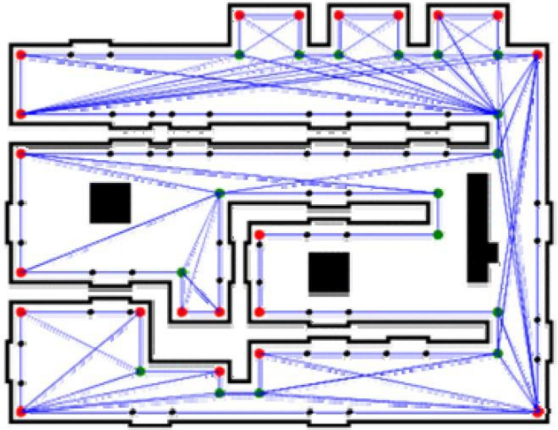
Mapping

Represent the robot environment by means of a Map.

Possible representations:

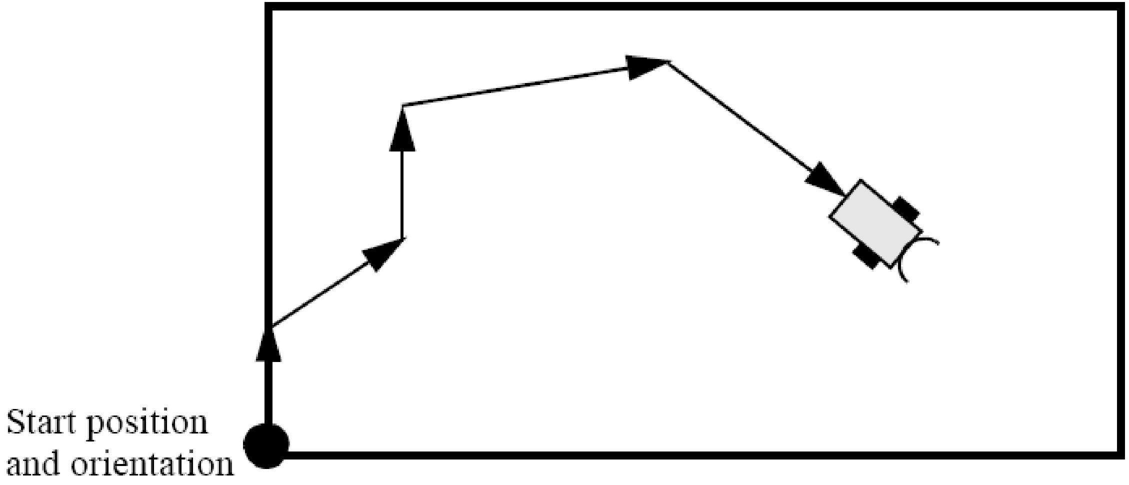
- Grid based / metric map: Environment is mapped with an evenly spaced grid, with each grid cell storing its probability of occupancy
- Topological map: depicts landmarks and their relations. The map is represented by a graph with nodes corresponding to landmarks and edges representing valid paths between these locations.





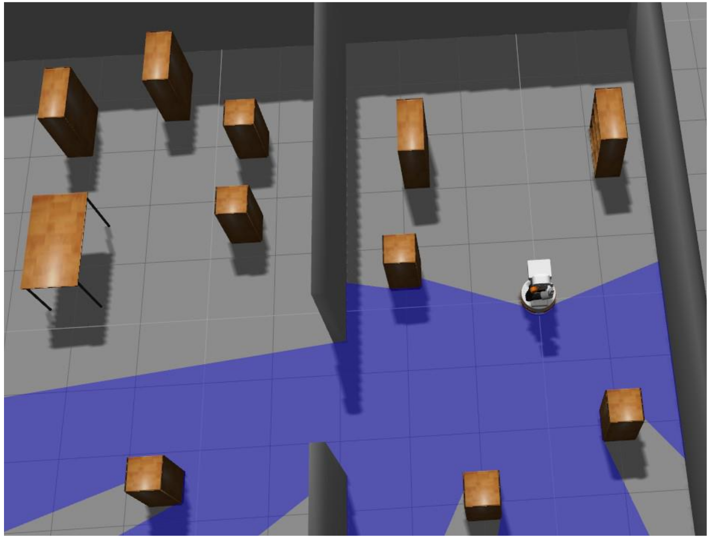
Localization

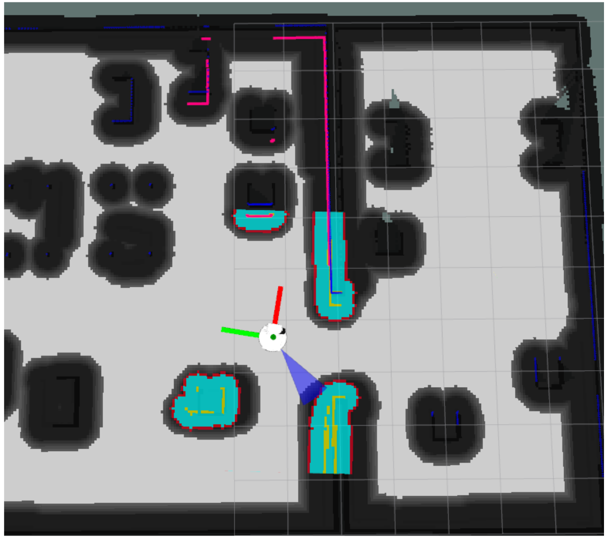
Robot position with respect to the environment



Navigation

Find a path to a goal location given a localized robot and a map of the environment



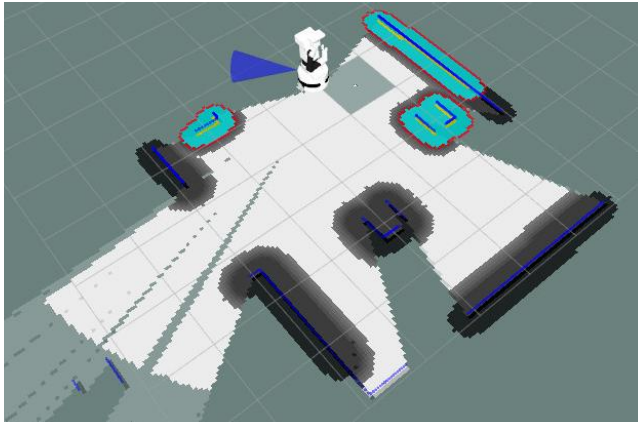


SLAM – Simultaneous Localization and Mapping

In practice, localization and mapping cannot be solved independently.

SLAM builds a map and localizes the robot in it at the same time.

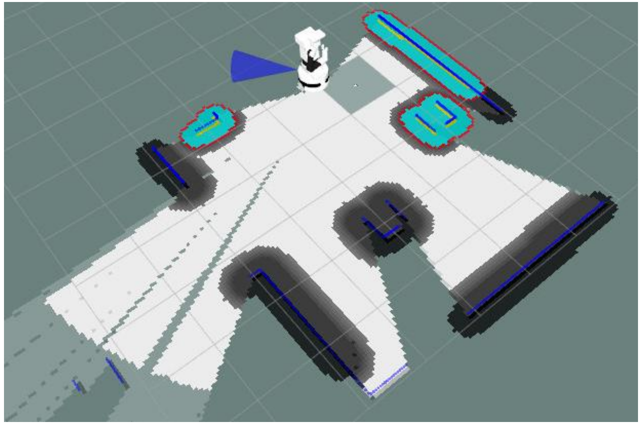
Luckily “gmapping” provides this capability for us



Mapping in ROS

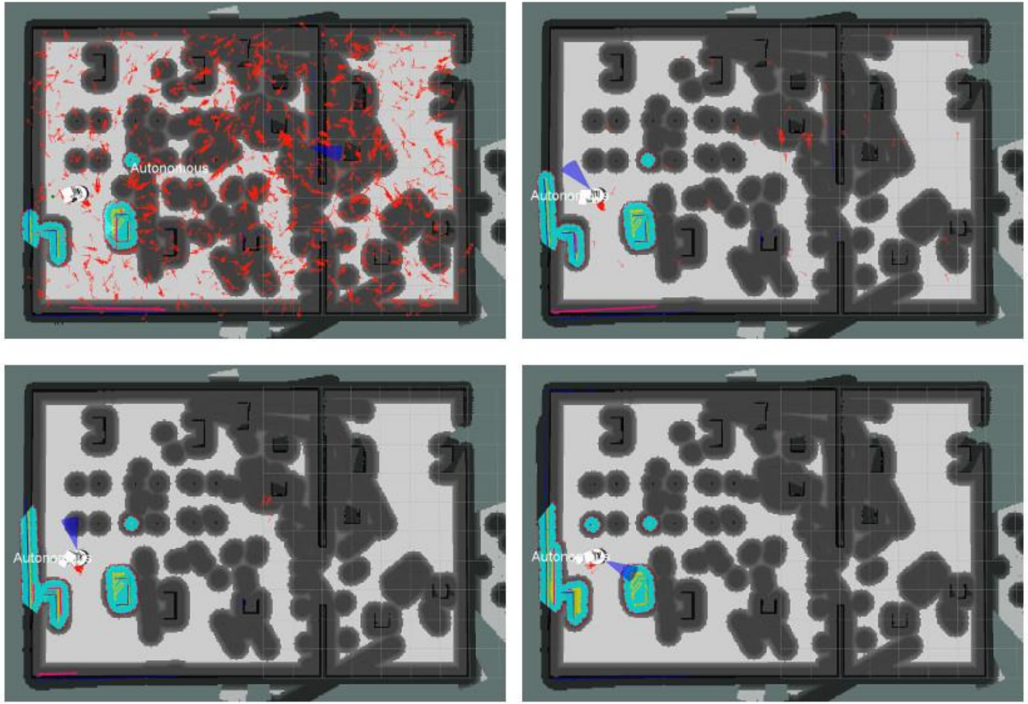
“gmapping” package allows to build 2D occupancy grid maps with laser sensor data and robot position
wiki.ros.org/gmapping

But:
If the position of the recording sensor is not precisely known,
Recorded data can’t be put into place on a map.
Therefore we need an approach that simultaneously localizes the robot (for the sensor position) and maps the environment

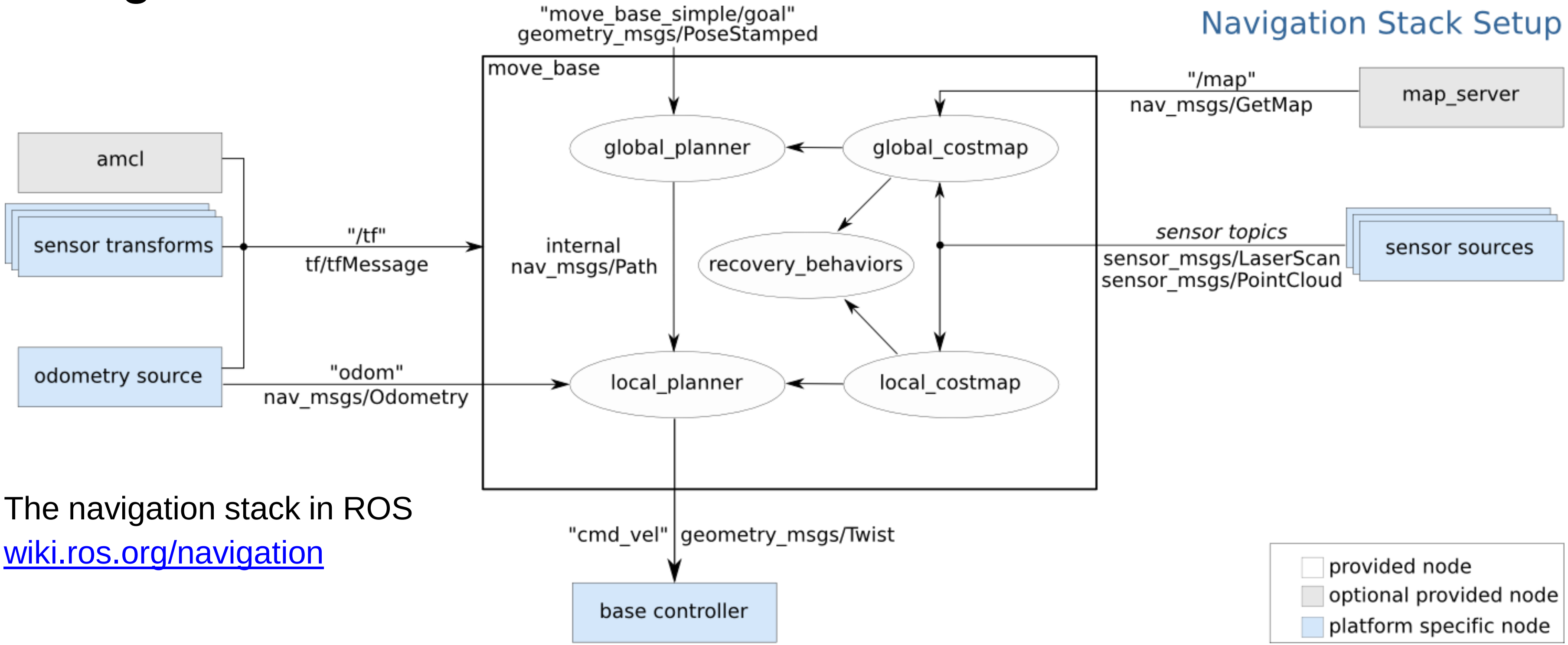


Localization in ROS

“amcl” package implements probabilistic localization. The employed technique is called Monte Carlo Localization.
wiki.ros.org/amcl



Navigation in ROS



The navigation stack in ROS

wiki.ros.org/navigation

Move robot to goal using C++

Check:
wiki.ros.org/navigation/Tutorials/SendingSimpleGoals

In order to move robot 1 meter forward from his current location use :

```
move_base_msgs::MoveBaseGoal goal;
goal.target_pose.header.stamp = ros::Time::now();
//reference frame is current location of the robot
goal.target_pose.header.frame_id = "base_link";
//moving distance is 1 meter
goal.target_pose.pose.position.x = 1.0;
//robot is moving forward
goal.target_pose.pose.orientation.w = 1.0;
```

Move robot to goal using C++

To move the robot to a specific goal on the map do:

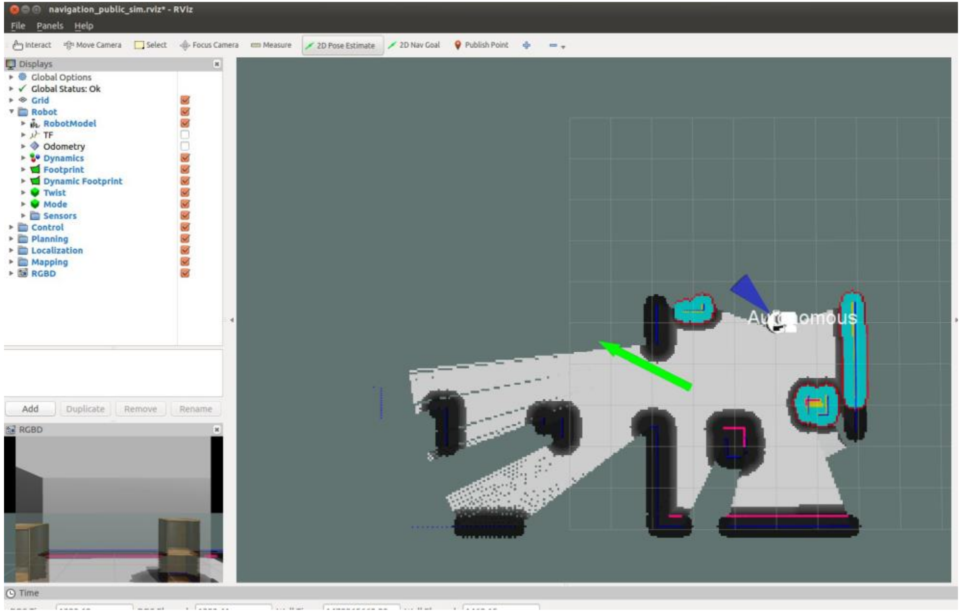
```
move_base_msgs::MoveBaseGoal goal;
goal.target_pose.header.stamp = ros::Time::now();
//reference frame is map
goal.target_pose.header.frame_id = "map";
//set goal as 2D coordinate
goal.target_pose.pose.position.x = 1.22;
goal.target_pose.pose.position.y = -0.56;
//set goal orientation
goal.target_pose.pose.orientation.x = 0.0;
goal.target_pose.pose.orientation.y = 0.0;
goal.target_pose.pose.orientation.z = 0.91;
goal.target_pose.pose.orientation.w = 0.43;
```

Find goal coordinates on the map

Use rviz button 2D Pose Estimate to select desired position. Hold your mouse, and select desired orientation by rotating the arrow.

The resulted position will be published to topic:
/initialpose

Make sure you subscribed to it before using 2D Pose Estimate



Exercises

Before doing the exercises, follow along these two tutorials:

- wiki.ros.org/Robots/TIAGo/Tutorials/Navigation/Mapping
- wiki.ros.org/Robots/TIAGo/Tutorials/Navigation/Localization

There are 2 distinct tasks:

1. Use gmapping to generate a map
2. Create a package and use amcl to navigate Tiago in said map (http://wiki.ros.org/ROS/Tutorials/CreatingPackage)

Create one tar file with subfolders "task_1" and "task_2" for the two tasks and the following naming scheme:

- "Name_lastName_roboCupHome_tutorial3.zip"

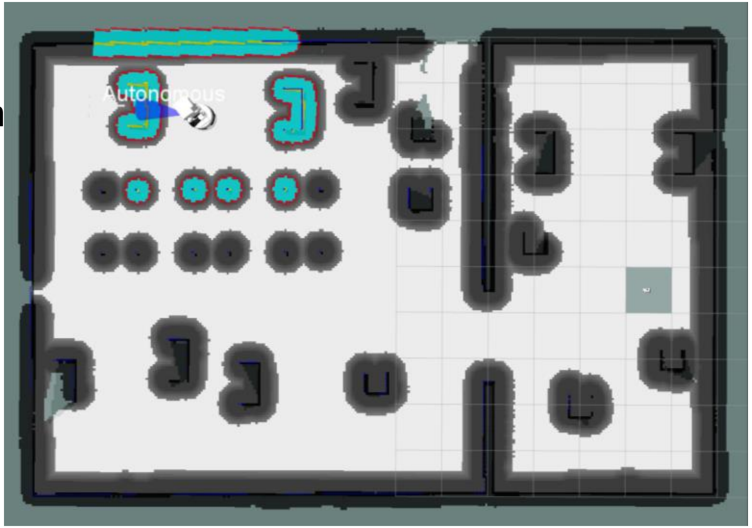
Send the files to: robocup.atHome.ics@gmail.com

Exercises

Task 1:

Generate two maps using Tiago:

- *"small_office"* which is the default world for Tiago navigation
- *"tutorial_office"* which is contained in the tiago_gazebo package



Deliverables:

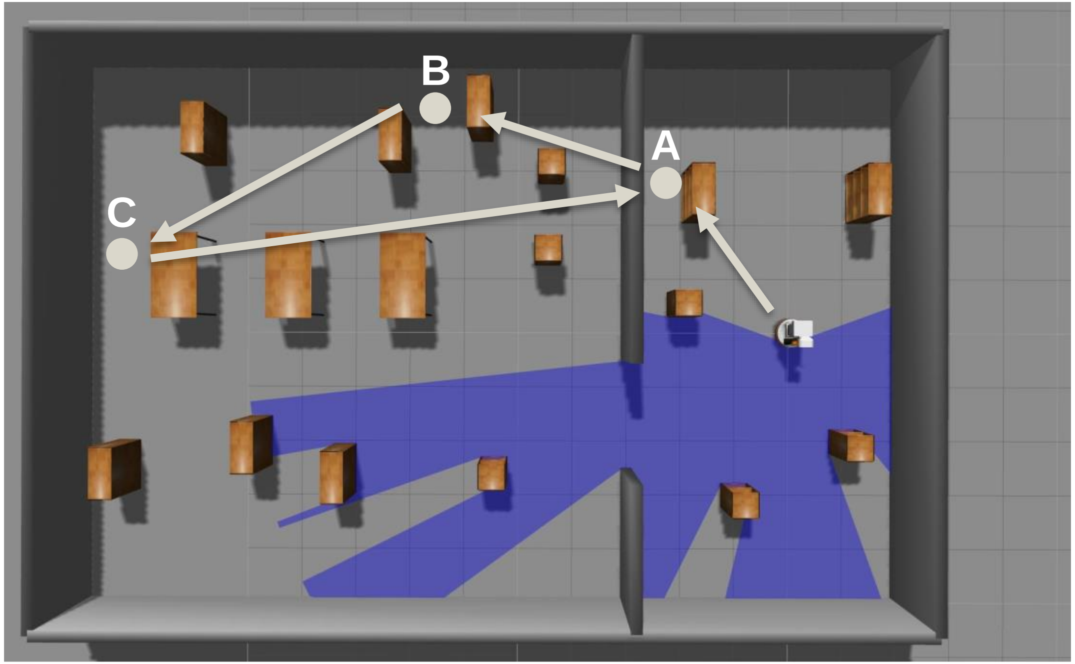
Create one subfolder *"task_1"* containing the following files:

- *map.pgm*
- *map.yaml*
- *mmap.yaml*
- *submap_0.pgm*
- *transformation.xml*

Exercises

Task 2:
Create a package to navigate Tiago in the *“small_office”* world. The robot should continuously and **autonomously** move between points A,B,C and loop as shown in the picture. Use the TIAGo and navigation_stack tutorials for reference

Deliverables:
The complete package in folder *“task_2”.* The package should contain a .launch file *“navigation.launch”* which automatically starts the navigation after launching.



Further Reading

Mapping:

- Grisetti, Giorgio, Cyrill Stachniss, and Wolfram Burgard. "Improved techniques for grid mapping with rao-blackwellized particle filters." *IEEE transactions on Robotics* 23.1 (2007): 34-46.

Localization:

- Fox, Dieter, et al. "Monte carlo localization: Efficient position estimation for mobile robots." *AAAI/IAAI* 1999.343-349 (1999): 2-2.

SLAM:

- Thrun, Sebastian, and John J. Leonard. "Simultaneous localization and mapping." *Springer handbook of robotics*. Springer Berlin Heidelberg, 2008. 871-889.