Homework 4
Gazebo Exploration

Robotics 2026 - ShanghaiTech University

I Overview

The goal of this homework is to learn how to use the Gazebo http://gazebosim.org/ robot simulator and learn about robotics and ROS at the same time. You will create your own simulated world, you will create a robot model with a laser and a camera and you will use ROS and rviz to visualize the simulated data. Additionally you will use gmapping and navigation2 for mapping and simple autonomy.

Recommended tutorial to follow

Here we recommend a 5-chapter tutorial for you. By following it you can acquire some essential skills for accomplishing the tasks in this homework. Feel free to use it as a framework for your homework. However, your submission should meet the requirement specified in the tasks below and should contain significant original contribution from you. Do NOT carry over the tutorial without any modifications. By doing that you would receive barely no points for this homework.

1. How to Create a Simulated Mobile Robot in ROS 2 Using URDF
2. Set Up the Odometry for a Simulated Mobile Robot in ROS 2
3. Sensor Fusion Using the Robot Localization Package – ROS 2
4. Set Up LIDAR for a Simulated Mobile Robot in ROS 2
5. Navigation and SLAM Using the ROS 2 Navigation Stack

Task 1: Create your own simulated world (20 %)

You should create your own world to simulate a robot in. It can be indoors, in which case it needs to have at least 2 different rooms. Or it can be outdoors, where there must be at least some narrow passages as well as a bigger place surrounded by some obstacles. You are of course free to model the STAR Center or the ShanghaiTech campus, for example.

In any case, the world must be closed: Starting inside the world, the robot should not be able to leave the world. So there should be walls or other obstacles all around.

Task 2: Create a simple robot model (20%)

In Gazebo you can also create your own robot. The robot to be simulated should be a differential drive robot. The robot has to have one camera facing forward and one laser scanner. The robot should move properly and the sensors should deliver data in the namespace 'camera' and 'lidar'. You may find it helpful to reduce the number of samples per beam of the laser in order to get a good performance.

Also make sure you create the according UDRF to be able to use the robot in ROS and visualize it in RVIZ (including the models).

Create a launchfile called 'gazebo.launch' to start your world with your robot.

Task 3: Autonomy! (15%)

Now you have a robot in a simulator in ROS. Time to give it some autonomy! First we need some mapping. Gmapping is a very well known 2D SLAM package available in ROS. Use it with your lidar to crate a 2D map.

Now launch your nodes to enable autonomy. When using rviz to set a goal point, the robot should drive to it, avoiding obstacles along the way.

You should create a launchfile called 'autonomy.launch' that launches both gmapping and all of the needed ROS navigation nodes.

Task 4: Visualize the data in rviz (5%)

Use rviz to visualize the data. The following data should be displayed:

- the grid
- the robot model
- the camera data in an imageview
- the laser data in the 3D view
- the 2D map in the 3D view
- the tf tree

Save the configuration of rviz. Create a launchfile called 'rviz.launch' to start rviz with this configuration file. Make a screen shot called "task4.png".

Task 5: Exploration (35%)

Finally you get to code some robot autonomy now! The goal is to have the robot explore the environment. For that the robot should go to areas that it hasn't seen yet and map them. Implement a frontier exploration algorithm for robot exploration. Suggested steps:

1. Get the 2D gridmap from HW3/ any SLAM you like.
2. In the map identify big frontiers:
   (a) Discretize all cells as either: free, occupied or unknown
   (b) Mark as "frontier" all cells that are unknown but have a free cell as neighbor
   (c) Create sets of connected frontier cells
   (d) Filter out sets with too few frontier cells
   (e) Calculate the centers of big frontier sets
   (f) **Visualize all big frontier sets in a certain color in rviz!**
3. Select one of those big frontiers as the new goal for navgiation2. This can be based on a number of criteria, e.g.: closeness; direction of the w.r.t. frontier location; what was the previously selected frontier; etc.
4. re-calculate the frontier goal on some of these conditions: reached within x meters of the goal; after y seconds; got stuck; ...

The above is a suggestion. You are free to make the algorithm better.

Submit your code (C++ or python) to your HW4 directory in a package called: HW4 and the file name should also start with hw4_exploration. Also make a screen capture video called 'exploration.mp4' (max 30MB) of the rviz view of your robot performing this exploration.

Create a launchfile 'exploration.launch' that will launch everything that is needed to run the exploration from scratch (you may just use the include tag to include launchfiles from previous tasks!).

Task 6: Submission (5%)

Your submission consists of all the files needed to run all five tasks. Everything has to be committed to a ROS package folder named 'HW4' to git and pushed to the gitlab server.

You should put all files to modify in this "HW4" package instead of modifying the files in the dependencies packages directly.

The following files are important:

- In the HW4 package folder a "launch" folder puts all your launch files including "gazebo.launch.py", "autonomy.launch.py", "rviz.launch.py", "exploration.launch.py" etc.
- In the HW4 package folder a "models" folder puts all your robot models.
- In the HW4 package folder a "worlds" folder puts all your world models.
- In the HW4 package folder a "rviz" folder put all your rviz configuration files.
- In the HW4 package folder a "src" folder put your exploration code.
- In the HW4 package folder put the media files "task4.png" and "exploration.mp4".