The University of Sheffield

ACS6121

Mobile Robotics and Autonomous Systems

Real-Robot Lab: World Exploration

Due: 11:59 pm, Friday, May 10, 2024

Dr Lin Cao

l.cao@sheffield.ac.uk

Department of Automatic Control and Systems Engineering

March 3, 2024

Version 1

Revision Notes

Version 1:

- Initial release.

Contents

1 Lab Overview

1.1 Introduction

In Weeks 5-6, you will work with your team members to develop a ROS package to control a Turtlebot3 for a World Exploration task in the simulation environment. The lab location for weeks 5-6 will still be North Campus Modular Village Computer Room. In weeks 8-10, you will test and tune the developed ROS package on a real robot in a physical robot arena in Computer Room 5, Diamond Building. **The World Exploration task is to let the robot autonomously explore the robot arena containing obstacles, and it must explore as much of the arena as possible in 90 seconds without crashing into anything! Whilst exploring the environment, the robot is expected to generate a map of the arena in the background.**

Unlike the Simulation Lab, there won't be any taught content or weekly exercises for you to work through for the Real-Robot Lab; you are now expected to work more autonomously in your team. Your team will need to submit a ROS package via Blackboard before the deadline **(11:59 pm, Friday, 10 May 2024)**. The submitted ROS package will then be assessed by the teaching team (Dr Lin Cao and the GTAs). The teaching team will run your team's submitted ROS package for three times, and the performance of the robot during the three runs will all be considered in the marking. The lab is worth 20% of the overall mark for the ACS6121 course.

Note that you only need to make one submission per team to Blackboard for this lab. Peer Assessment will be implemented so that your individual contribution to your team's project will be considered in your own final mark.

1.2 Teams, Lab Sessions, and Prerequisites

All students are divided into two large groups, Group 1 and Group 2. Based on the feedback from previous years, this year, we allow you to form your own teams to boost your team collaboration and enable a better teamwork experience (although finding your team members may be a bit pain!). Teams 1-17 are for Group 1, and Teams 18-33 are for Group 2.

For this Real-Robot Lab, you are expected to actively work together with your team members to complete the lab task. Your lab time slots can be found on your timetables and in the "Teaching Plan" section on Blackboard.

Some important notes:

- Before week 8, you are expected to use both your lab hours and out-of-lab time to complete a fully working ROS package in the simulation environment, and it is ready to be tested on a real robot. Otherwise, you may not have sufficient time to tune the package with the real robot during the scheduled three lab sessions in Weeks 8-10. As such, **it is extremely important that you have a fully working ROS package ready to be tested before week 8.** Proper project management is key!
- Before week 8, each member of your team needs to complete a health & safety quiz on Blackboard. You can find this quiz and its associated tasks via the following path:

Blackboard/ACS6121/RealRobot Lab/Real Robot Health and Safety Quiz and Resources

- During your lab sessions of Weeks 8-10, you will have full access to the robot and its associated laptop (with Linux system and ROS). Each robot has only one laptop; if your team needs more laptops, you can use your own laptops and install WSL-ROS on them by following the installation guide. With your personal laptops, you can also remotely access WSL-ROS through the university-managed remote computers.
- In week 8, follow this guide Working with the Real TurtleBot3 Waffles to work with the real-robot. You are advised to go through this guide before coming to the real-robot lab session on week 8 to save your team's time. **Pay special attention to the out of range LIDAR data (Mission 4) mentioned in the the above guide because the out of range LIDAR data may bring errors your coding!** You can skip Missions 2, 3, and 5 as you will not need them to complete your real-robot lab.

This lab is not only designed to give you an experience of programming robots, but also gives you the chance to develop and demonstrate essential team working and project management skills: planning and scheduling, assigning roles and responsibilities, delegating tasks, distributing workload appropriately and communicating effectively within your teams to complete this task to the best of your abilities. Certainly, it is up to you to decide on the best way to work effectively within your teams, but you should try to communicate regularly, not only during the scheduled lab sessions, but at other times as well.

1.3 Getting Support

The best way for you to get support is to attend the weekly lab sessions, where the course GTAs will be present to answer questions that you may have. Although GTAs are more than happy to discuss/help, do not expect the GTAs to debug the entire code on your behalf on the spot.

Remember to use the Simulation Lab webpage as a reference throughout this work: taking the things that we worked on in the Simulation Lab and applying them to your team's ROS packages.

Finally, don't forget to use the ACS6121 Discussion Board on Blackboard to post any questions you may have regarding this Lab.

1.4 Submission Requirements

The submission deadline for your team's ROS package is **11:59 pm, Friday, May 10, 2024**. The Blackboard submission portal will become open on Week 10. In order to be assessed, you must ensure that the following requirements are met with regard to the ROS package that your team submits (as well as any additional requirements described in the later sections of this document):

- As mentioned earlier, everything your team submits for this lab must be contained within a single ROS package. Inside this you will develop all the necessary nodes that allow your robot to complete the exploration task.

- In the root of your team's package directory , there should be a launch folder; in the launch folder, there is a launch file with the following naming convention:

  explore.launch

  The task will be assessed by the teaching team using this launch file.
- Your team's ROS package must work 'out-of-the-box', i.e. the module leader and GTAs will not fix errors or modify your codes for you during the assessment.
- Your package must be submitted to Blackboard as a .tar file with the following naming convention:

  acs6121_team{}.tar

  Where the {} must be replaced with your own team number. Detailed instructions on how to convert your ROS package to .tar file are provided below.

1.5 Exporting your ROS Package for Submission

When it comes to submission time, it's important that you follow the steps below carefully to create a .tar archive of your ROS package correctly. We recommend that you do this from WSL-ROS on a University Managed Desktop Computer (rather than one of the robot laptops), so that you can save it to your University `U:` Drive.

1. First, navigate to the `catkin_ws/src` directory in a WSL-ROS terminal instance:

   ```
   cd ~/catkin_ws/src/
   ```

2. Then, use the `tar` command to create an archive of your package:

   ```
   tar -cvf /mnt/u/wsl-ros/acs6121_team{}.tar acs6121_team{}
   ```

   ... replacing `{}` with your own team number again.

   This will create the `.tar` archive in your own personal University `U:` Drive, which you can access using the *Windows File Explorer...*

3. In *Windows*, open up Windows Explorer, click "This PC" in the left-hand toolbar and locate your own personal `U:` Drive in the "Network locations" area.
4. In here there should be a `wsl-ros` folder, which should contain the `.tar` file that you have just created.
5. Submit this `.tar` file to Blackboard via the appropriate submission portal.

Now, let's get started!

2 Getting Started

2.1 Working with the Real Robots

Before Week 8, your team must have developed a fully working ROS package for testing on the real-robot and all your team members must have finished the health & safety quiz and its associated tasks before being allocated with the robot and its laptop.

During the lab sessions on weeks 8-10, your team will be given a Turlebot3 Waffle robot and a laptop. As mentioned earlier, to get started controlling the real robot, you can refer to Working with the Real TurtleBot3 Waffles for your reference (pay special attention to the out-of-the-range LIDAR data). There are a few exercises that you can have a go there too (skip Missions 2, 3, and 5 as they are not relevant.). Note that these robots are far less robust than your phones, you need to handle them carefully (e.g., always run it in the robot arena instead of places with people walking nearby)!

The robot and laptop can only be used in the Diamond Computer Room 5. At the end of each lab session, you need to follow proper procedures described here Safe Shutdown to safely turn off the robot rather than simply turning off the power. **Failing to turn the robot off properly may result in data loss or issues when you use it next time.** At the end of each lab session, return the robot and laptop to the GTAs and sign out on the equipment record form (to be provided by the GTAs).

2.2 Simulation resource

While you'll need to do this lab on a real robot, we have put together a ROS package called "acs6121" which can provide you with a simulation world (Figure 1). The simulation world is similar to the real robot arena; so you can firstly develop/test your codes in the world, and then test them on the real robot in the arena. Note that, however, the assessment will be made only in the arena using the real robot.

Follow the steps below to get the acs6121 package to your WSL-ROS environment:

1. In a WSL-ROS terminal instance, navigate to the catkin_ws/src directory:

   ```
   cd ~/catkin_ws/src/
   ```

2. Then, clone the package from GitHub to your local directory:

   ```
   git clone https://github.com/lincaolab/acs6121_realrobot.git
   ```

   You may notice that the repository name (acs6121_realrobot) is not necessary the same as the ROS package name (acs6121)!

3. From the same terminal location, run catkin build to compile the new package:

   ```
   catkin build acs6121
   ```

4. Finally, re-source your environment:

   ```
   src
   ```

5. You can then launch the simulation world from the acs6121 package with the following roslaunch command:

```
roslaunch acs6121 arena.launch
```

You shall see the simulation world as shown in Figure 1.

The robot arena used for assessment might look something like the simulation environment as well as the real robot arena you see in the lab sessions. Obstacles include four wooden walls 160mm tall, 10mm thick and either 440 mm or 880 mm in length, and three cylindrical beacons of 200mm diameter and 250mm height. However, **the robot arena for assessment will be different from those in the simulation and lab sessions**; that is, the wooden walls and the beacons will be at different locations (but the robot will always have access to any free space in the arena) and the physical arena will be slightly larger (4-by-4 square meters). Note that the colors of the obstacles do not matter for this lab.

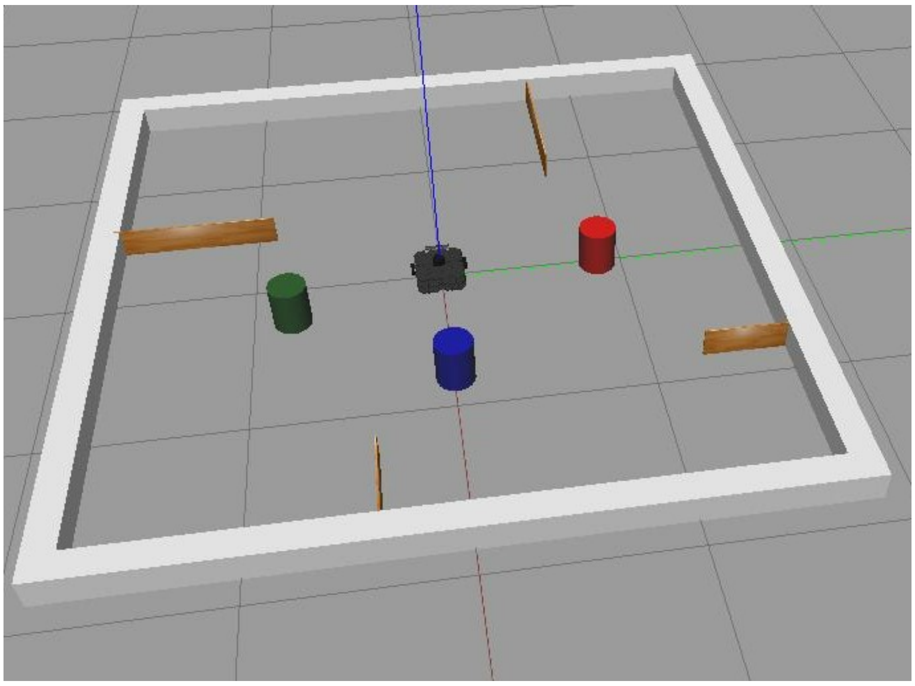

Figure 1: Real-Robot Lab simulated exploration world.

During assessment, your package will be run for three times by the teaching team. The robot will always be placed in the center of the arena, but its orientation will be different for each running. So you may wish to test this out also in your simulation environment. You can do this by changing the orientation of the robot once the simulation environment is launched. See the instructions in Figure 2 and Figure 3.

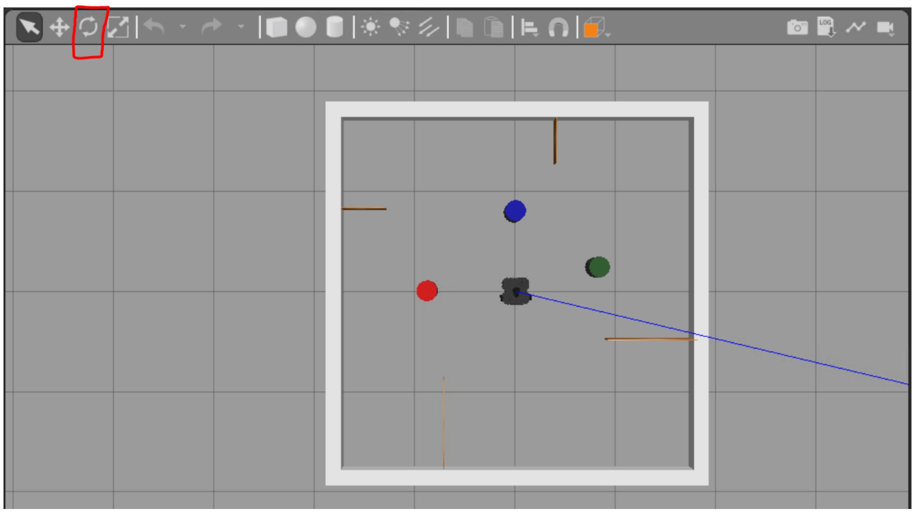

Figure 2: Click the tool highlighted in red.

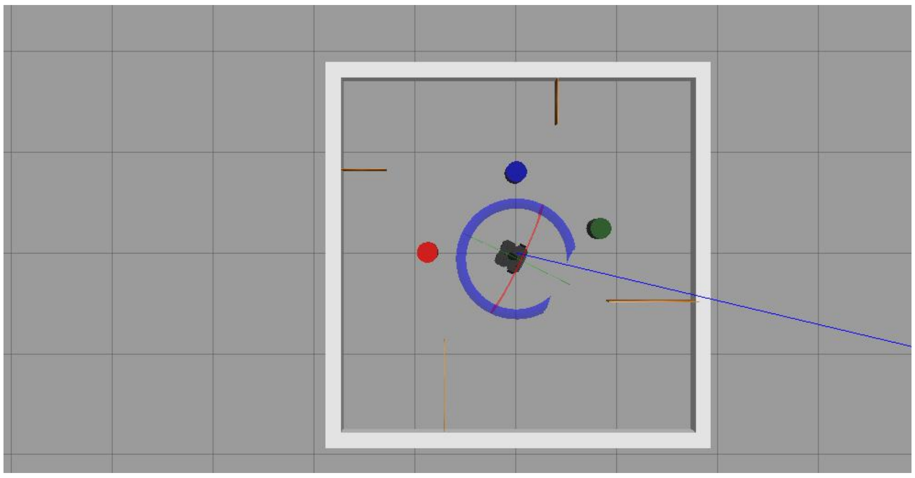

Figure 3: Click hold and drag the blue ring to change the orientation of the robot.

2.3 Transferring Your Team's ROS Package

Your team will need to develop a ROS package to control the robot to navigate in the arena. Once you have a working package developed using the simulation world, you may wish to test on in the real robot. In this case, you will need to transfer your package to the robot laptop whenever you want to work on a real robot and test things out in the real robot arena during one of the lab sessions! There are many ways to transferring you team's package between WSL-ROS and the robot's laptop, using a USB flash drive, google drive, U drive, or GitHub. It is up to you to decide which way is to use.

There is a Catkin Workspace on each of the robot laptops (much the same as in the WSL-ROS environment), and your package must reside within the catkin_ws/src directory when working with the real robot! To transfer your package between devices, you can copy one directory (your U drive) to another directory (e.g., the robot's laptop) using the following command:

```
cp -r path_to_source/ path_to_destination/
```

Check this link to find more details about copying directories in Linux system.

If you are using Visual Studio Code, you can simply copy and paste.

Whatever approach you use to transfer your package, you will need to compile the package to the device. So make sure you are located in the root of the Catkin Workspace and then compile the package using

```
catkin build {package name}
```

After that, resource the environment using `src`, like what you would do for a newly created package.

And now turn to the next Section for the details of the task...

3 Task Details

The robot arena consists of boundary walls and obstacles. Your robot will need to be able to detect these walls and obstacles and navigate around them in order to fully explore the space.

1. The robot will start in the centre of the arena, perpendicularly facing to one of the four walls of the robot arena.

2. It must explore the environment for 90 seconds without touching any of the arena walls or the obstacles within it.

3. If the robot makes contact with *anything* before the time has elapsed, then "Ctrl C" will be pressed in the terminal to stop the attempt and the navigation time will be recorded for marking. If your robot runs for more than 90 seconds, "Ctrl C" will also be pressed in the terminal to stop the attempt. So, make sure that your robot can be stopped when "Ctrl C" is pressed in the terminal.

4. Sixteen equal-sized zones (4 X 4 grid) will be evenly marked out on the arena floor. The robot must enter as many of the outer 12 zones as possible during the attempt.

5. Your robot must maintain a linear velocity of at least 0.1m/s throughout the entire duration of the task except for brief periods (of no more than a few seconds) where the robot need to turn on the spot.

6. The location, orientation of the wooden walls and cylinders in the arena during the assessment will be different from those in the simulation world and the arena you see in the lab sessions, so the ROS package that you develop will need to be able to accommodate an unknown environment.

7. As mentioned, your package must contain a launch file called explore.launch, which will be used by the teaching team to execute the functionality from within your team's package. This functionality must be launch-able via the command:

   ```
   roslaunch acs6121_team{} explore.launch
   ```

   Replace `{}` with your team number. Test this out in WSL-ROS to make sure that it works!

Some hints: Some concepts we introduced in the lectures can be adapted. For instance, the Finite State Machine which explicitly defines the actions to take for certain inputs and current state; or, the Artificial Potential Field which directly maps the robot's movements to its relative position with the obstacles. The solution for the last Tutorial Question of Unit 1 (Swarm Robotics) may also be considered. We designed a random walk finite-state machine, and this might be used as an obstacle avoidance and exploration strategy. These could also be further integrated with other strategies for improved performance, e.g., let the robot follow the wall of the environment or make a spiral search from the center to the boundary of the arena... Your call!

LIDAR may also generate out-of-the-range data. How do you filter out these data? LIDAR generates distance data all around the robot. You may also want to segment the distance data so that you could focus on a few key zones around the robot (e.g., forward, forward-left, forward-right). Then, based on the segmentation, what inputs/states you may have for different scenarios in the arena and what actions to take?

Advanced Feature: Mapping with SLAM

Marks will be available for an advanced feature: whilst your robot is exploring the arena, it also runs SLAM to generate a map of the arena in the background. In Session 3, Exercise 3 of the Simulation Lab we launched SLAM using the following roslaunch command:

```
$ roslaunch turtlebot3_slam turtlebot3_slam.launch
```

Once you've had a look at the 'Launch Files' on the lab webpage, it should be clear how to launch other launch files from within your own. Alternatively, you could navigate to the turtlebot3_slam package in WSL-ROS and have a look at the content of the launch file that we executed from within this package for Exercise 3, to see how you might be able launch the same (or similar) functionality within your own explore.launch file. When it comes to saving the map that has been generated by SLAM, think about how we did this in the Week 3 Exercise. This involved calling a map_saver node from the command-line. It is possible, however, to call nodes (and indeed launch files too) from within other ROS nodes using the roslaunch Python API, which we introduce at the end of the 'Launch Files' page of the lab webpage too. The root of your package directory must contain a directory called 'maps', and the map files must be saved into this directory with the name: explore_map.

4 Marking

There are **20 marks** available for this lab in total. As mentioned, during the assessment, your team's ROS package will be run for three times in the arena. The robot will always start in the center of the arena. The orientation of the robot will be different for each run, but it will always perpendicularly face to one of the four walls of the robot arena. Each run will be marked based on the criteria outlined in Table 1. There will be three markings, and the final mark for your team will be the weighted sum of these three markings. The percentage weights for the three markings are as follows: 60% for the highest marking, 30% for the medium marking, and 10% for the lowest marking. For instance, if the markings for the three runs of your team's ROS package are 18, 15, and 8 respectively, then the final mark for your team will be $60\% \times 18 + 30\% \times 15 + 10\% \times 8 = 16.1$.

The assessment will be conducted by the teaching team during the exam weeks.

5 Self and Peer Assessment

As the final step of the lab, each team member needs to do a **Slef and Peer Assessment**. Your final mark will depend on not only your team's mark but also your individual contributions to the team as assessed by other members of the team. It is a key opportunity for you to assess your team members' contributions and reflect on yourself's contributions as well as learning. The submission portal will be available in the Real Robot Lab section on Blackboard on Week 10. It is important that all team members complete this assessment individually. Failing to submit this assessment will result into a 10% penalty to your individual mark of the lab.

Table 1: Marking Criteria.

Criteria		**Marks**	**Details**
A	Launch-able package	2/20	You will be awarded the 2 marks if your team's package can be launched with the following command: `$ roslaunch acs6121_team{} explore.launch` The `{}` will be replaced with your team number.
B	Run time	7/20	You will be awarded marks for the amount of time that your robot spends exploring the environment before 90 seconds has elapsed, or the robot makes contact with anything in its environment. The longer the robot explores the arena without contacting with anything, the higher the mark will be.
C	Exploration	8/20	You will be awarded marks for the number of the outer zones of the arena that your robot manages to enter (not including the four central zones). The robot only needs to enter each zone once, but its full body must be inside the zone. The more outer zones the robot enter, the higher the mark will be.
D	Mapping	3/20	If, at the end of the attempt (e.g., "Ctrl C" is pressed in the terminal), there is a maps folder in the root of your package directory, within this there exists both a explore_map.pgm and explore_map.yaml file and the explore_map.pgm is indeed a map of the real robot arena in its current form (it is still acceptable if the map does not show the complete arena).

While working on the project, try to define your team's sub-objectives or tasks for all members to contribute more or less equally. Some members may contribute far less than others due to the lack of related skills or poor engagement; these members may end up with relatively low peer assessment marks. While working on the project, the team must take actions to inclusively run the project so that all members have been given the opportunities to actively contribute; for instances, these actions may include (but not limited to) reasonable objectives and task allocation, accessible materials (e.g., codes) of the team's progress, and open discussions.

This Self and Peer Assessment must be completed by no later than 11:59pm, Friday, 10th May 2024. Each member of the team needs to submit an individual assessment.

Good luck and have fun with the real robot!

The End