The University of Sheffield

ACS6121

Mobile Robotics and Autonomous Systems

Real-Robot Lab: World Exploration

Due: 11:59 pm, Sunday May 22, 2022

Dr Lin Cao

l.cao@sheffield.ac.uk

Department of Automatic Control and Systems Engineering

May 12, 2022

Version 3

Revision Notes

Version 1:

- Initial release.

Version 2:

- Fixed a few typos.
- Corrected that the launch folder should be in the root of your team's package directory rather than in the src directory of the package (Section 1.4).
- Provided detailed instructions regarding how to transfer your package among multiple devices (Section 2.3).

Version 3:

- Fixed a few typos.
- Specified the initial orientation of the robot during assessment. Basically, for each run during assessment, at the initial position (center of the arena), the robot will be facing perpendicularly to one of the four walls of the robot arena. The revisions are highlighted in red in pages 9 and 10.
- Specified how the robot will be stopped when the 90 seconds elapse or the robot collides with any objects in the arena. The revisions are highlighted in red in pages 9 and 10.
- Added that the assessment will be conducted by the teaching team during the exam weeks. The revision is highlighted in red in page 10.

Contents

1 Lab Overview

1.1 Introduction

In this Real-Robot Lab, you will put into practice what you have learnt in the Simulation Lab to control a TurtleBot3 Waffle robot inside the robot arena in Diamond Computer Room 3 (DIA-CR3). **You will work together with your team to develop ROS node(s) to allow the robot to autonomously explore the robot arena containing obstacles. The robot 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 **(11:59pm, Sunday, 22nd May 2022)**. 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 and Lab Sessions

All students are divided into two groups, Group 1 and Group 2. Teams 1 to 11 are in Group 1, and Teams 12 to 21 are in Group 2. For this Real-Robot Lab, you are expected to work together with your team members to complete the lab task. The team list is available in the Real-Robot Lab area on Blackboard, so please refer to this to find out which team you have been assigned to and who your fellow team members are. The lab time slots for both groups are as follows:

- Group 1: Weeks 10-12, Tuesdays 4-6 pm
- Group 2: Week 10: Tuesday 1-3 pm; Weeks 11-12: Mondays 1-3 pm

You can also check your timetable to find out what time you should attend the Real-Robot Lab. During your lab sessions of Weeks 10-12, you will have full access to the robot and its associated laptop.

On Week 9, Wednesday 27 April at 1-3 pm, there is also an optional session in DIA-CR3 for all students to try out the real robots. Since we may not have enough space or robots for so many students and teams at one time, this session is made optional. However, you are still encouraged to come to have a go with the real robots by following the guidance on Working with the Real TurtleBot3 Waffles. Attending this session may save you some time for the Real-robot Lab starting on Week 10. Just to give you a heads-up: you may find it a bit crowd in this session because it is open to all students.

This lab is not only designed to give you 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.

Remember to use the GitHub Wiki 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, Sunday May 22, 2022**. The Blackboard submission portal will become open since Week 11. 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 acs6121_team{}.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

During the lab sessions, your team will be given a Turlebot3 Waffle robot and a laptop. To help you get started controlling the real robot, we have put together a tutorial of Working with the Real TurtleBot3 Waffles for your reference. There are a few exercises that you can have a go at there too. 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 3. At the end of each lab session, you need follow the below procedures to turn off the robot rather than simply turning off the power:

- First, close down any active processes that are running on the robot by checking through any active robot terminals and stopping any processes that are running using Ctrl+C.
- Then, shut down the robot by entering the following command in an ssh session running on the robot:

  $ off

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 a 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.git

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). Note that the colors of the obstacles do not matter for this lab.

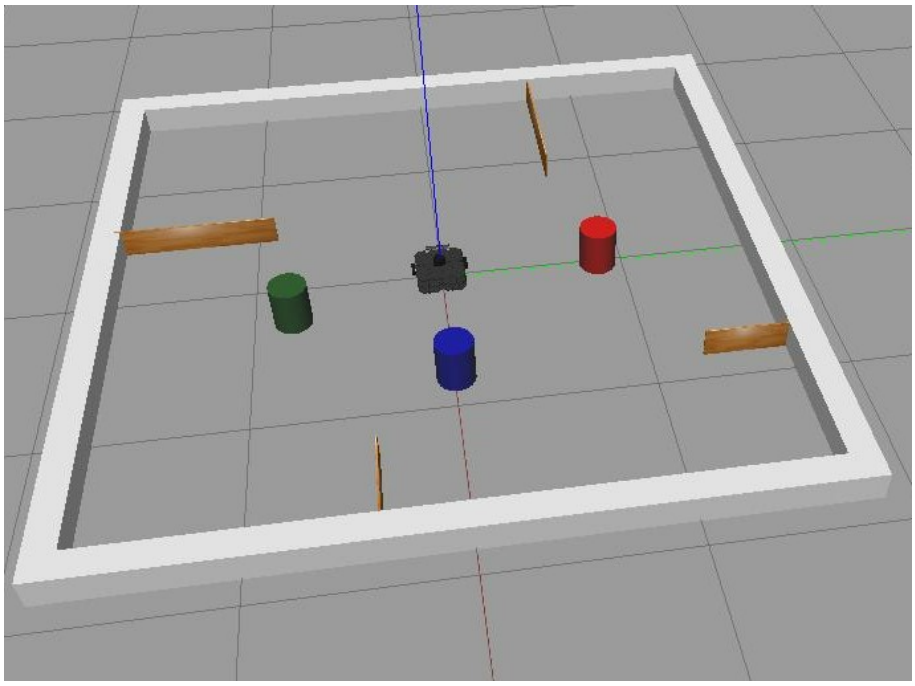

Figure 1: Real-Robot Lab simulated exploration world.

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.

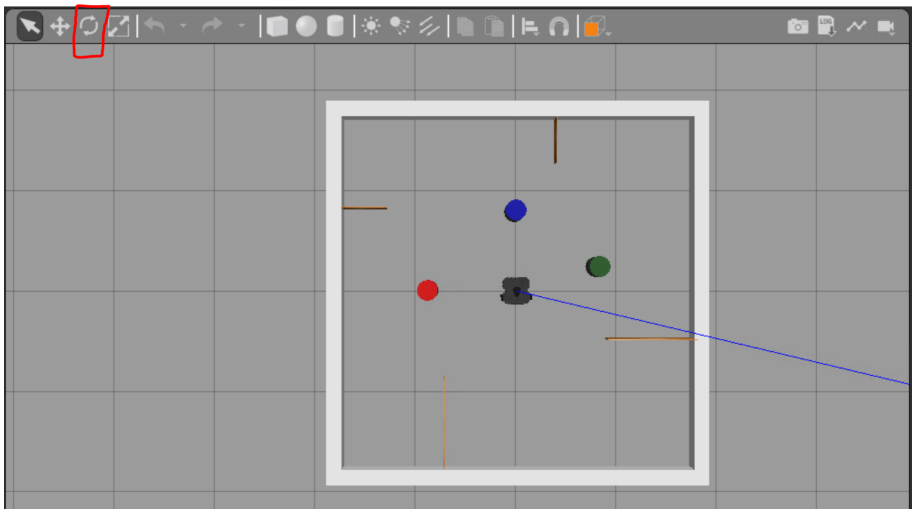

Figure 2: Click the the tool highlighted in red.

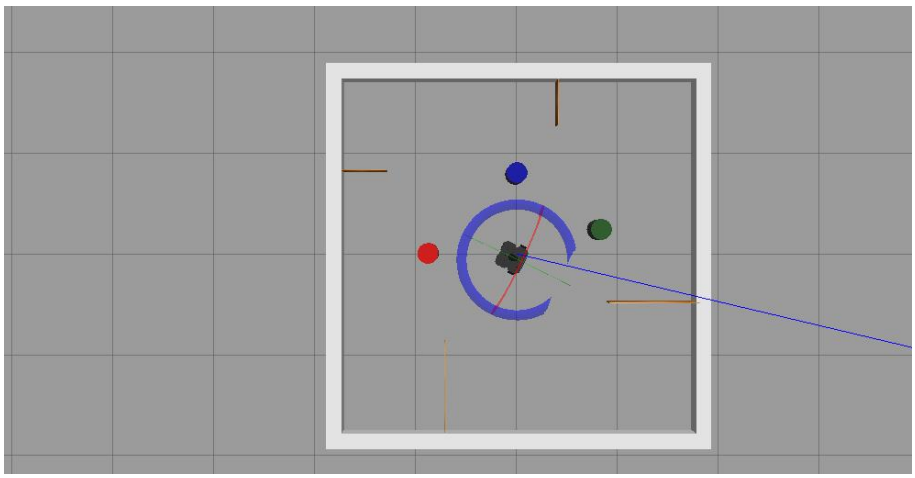

Figure 3: Click hold and drag the blue ring to change the orientation of the robot.

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 (in the robot or robot's laptop) to another directory (e.g., USB drive) 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. If the package is to be locally deployed on the robot, you must be located in the root of the Catkin Workspace and then compile the package using

```
cd  /catkin_ws && catkin_make
```

otherwise, use

```
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. Nine equal-sized zones will be evenly marked (3X3 grid) out on the arena floor and the robot must enter as many of these 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!

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' Wiki page 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 Wiki 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.

Criteria		**Marks**	**Details**
A	Launchable 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 zones of the arena that your robot manages to enter (excluding the one it starts in). The robot only needs to enter each zone once, but its full body must be inside the zone. The more 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).

Table 1: Marking Criteria.

5 Peer Assessment

Your final mark will not only dependent on your team's mark but also your individual contribution to your team. So, the final step for this lab is for you, as a team, to submit a 'Real-Robot Lab Peer Assessment Form.' The template can be found in the Real Robot Lab section on Blackboard. It is important that all teams complete and submit this form. Download the form from Blackboard and provide your team number, the names of all of your team members, as well as each team member's individual contribution to the Real-Robot Lab as a % (adding up to 100% in total for your team). Save the completed form using the format team{}.pdf (replacing {} with your team number), then submit it to the Blackboard submission portal which will become open on Week 11. This should be completed by no later than 11:59pm, Sunday, 22nd May 2022. Each team needs to submit only one form for peer assessment.

Good luck and have fun with the real robot!

The End