ECE4310/CIE6042 Programming for Robotics

# Project 1: Service Robot

School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen

2021-2022 Term 2

**Motivation:**

1. To integrate what you have learned and practiced in the previous labs to build a mobile service robot.

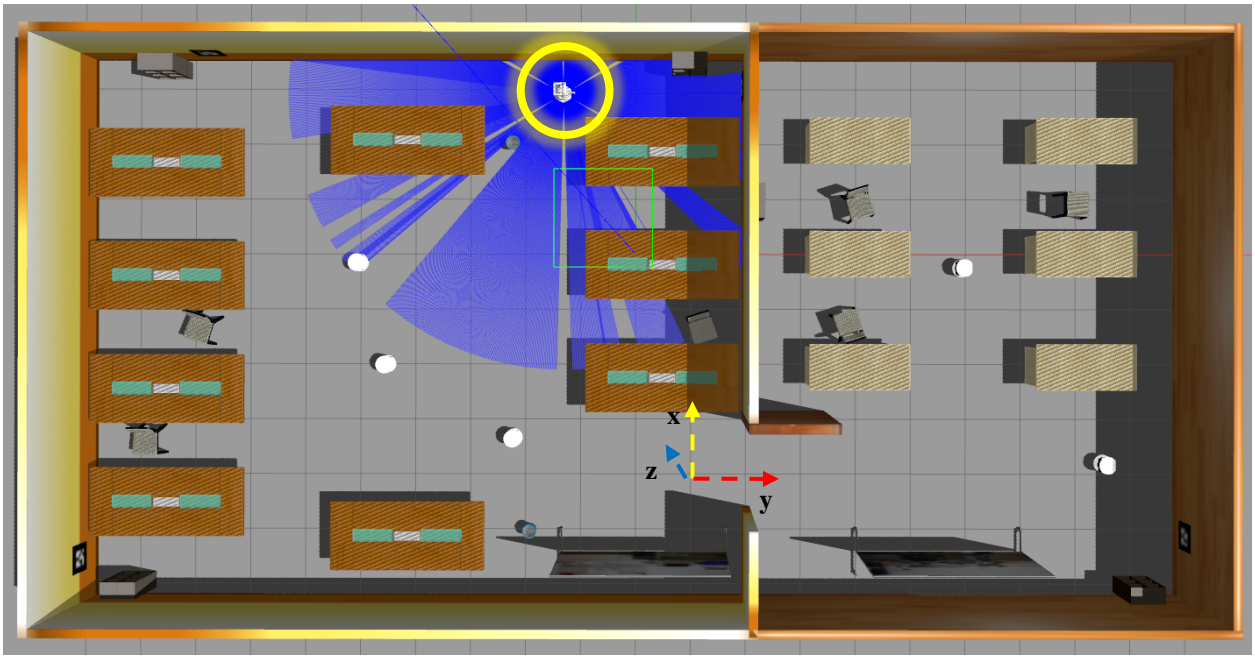



Figure 1: The RA317 virtual environment

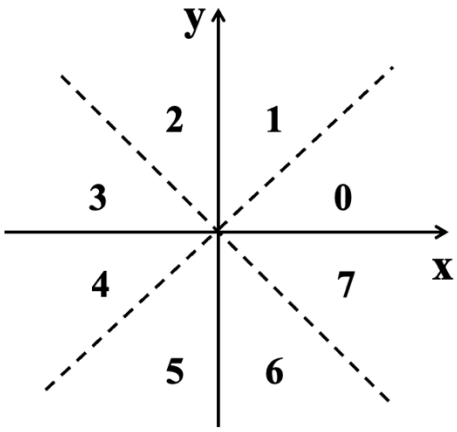
As shown in Figure 1, your Spark robot will be spawned at the yellow-circled area towards the first AR-tag of your delivery task. There are 5 AR-tags with storage cabinets in total to guide the procedure delivery task, each of which contains the relative direction from the current delivery location to the next delivery location. By using the *ar_track_alvar* package, you can get the pose and id of the detected AR-tag.

| Topic | Type | Bandwidth | Hz | Value |
|---|---|---|---|---|
| /ar_pose_marker | ar_track_alvar_msgs/AlvarMarkers | 1.03KB/s | 8.00 | |
| header | std_msgs/Header | | | |
| markers | ar_track_alvar_msgs/AlvarMarker[] | | | |
| [0] | ar_track_alvar_msgs/AlvarMarker | | | |
| confidence | uint32 | | | 0 |
| header | std_msgs/Header | | | |
| id | uint32 | | | 5 |
| pose | geometry_msgs/PoseStamped | | | |
| header | std_msgs/Header | | | |
| pose | geometry_msgs/Pose | | | |
| orientation | geometry_msgs/Quaternion | | | |
| position | geometry_msgs/Point | | | |
| x | float64 | | | 0.6030752496954835 |
| y | float64 | | | -3.2302620263747226 |
| z | float64 | | | -0.014442146200180792 |

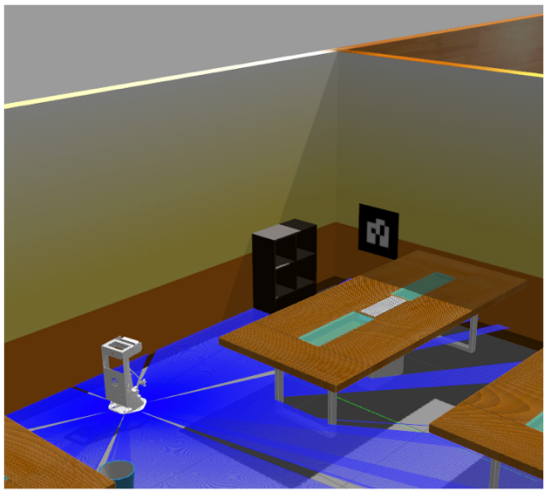
Figure 2: /ar_pose_marker topic information when detected AR-tag

As shown in Figure 2, there are two types of data you will need to accomplish the delivery task. The first one is *pose/position*, which is the rough location of the corresponding storage cabinet that helps the robot reach the delivery location.

The second one is *id* which is an integer ranging from 0 to 8 with relative direction to the next delivery location. The detailed directional encoding method is shown in Figure 3. For example, the initial AR-tag you will detect is *id=5*, which means that the next delivery location is within the range [-145°, -90°] with respect to the current AR-tag. Note that the last AR-tag is *id=8*, meaning that the delivery task is finished.




Encoded directional information (i.e. *id*)



Initial AR-tag with *id=5*

Figure 3: Encoded directional information in the AR-tag

**Tasks:**

1. You are going to build a service robot for delivery in an indoor environment.
2. The information about the delivery location is stored in an AR Tag and will appear in the environment. The first AR Tag is located at the starting position of the robot in the map. The next AR Tag will appear at the delivery location.
3. The simulated indoor environment (Gazebo) is provided, including the AR Tags in the environment, and the starting position of the robot. (Figure 1)
4. You are requested to make a robot that reaches all delivery locations automatically according to the information of the AR Tags.
5. The costmap is not provided. You may need to create one by yourself.

**Submission:**

**[Report]** In the report, include the methods and flow that you have used to achieve the task in detail.

**[Code]** Provide sufficient files and codes with adequate comments that we can replicate your result.

**[Demo Video]** Provide a video to show the delivery performance of the robot.

**[Deadline]** 9:00 am, April 28, 2022

**Gradings:**

- Report, Code, and Demo Video [50%]
- New methods [10%]: Bonus will be given if any effective library have been used which is not covered in the Labs.
- Presentation [20%]: Make a 5 minutes presentation in the class to explain how you achieve the task. (April 28, 2022 10:30am-11:30am)
- Competition [20%]: To show the effectiveness and the adaptiveness of your methods. We will reimplement your methods in the same environment but add some obstacles and change the location or order of the AR Tags. The faster the task is completed, the higher the score. (April 28, 2022 11:30am-12:30pm)

## Project Hint:

You may first follow the procedure below to check if the provided codes can progress well:

1. Build the source code

   $catkin_make

2. Start Gazebo simulation:

   $export GAZEBO_MODEL_PATH=$GAZEBO_MODEL_PATH:<your-ws-path>/src/spark1_description/model

   $roslaunch spark1_description spark_gazebo.launch

3. Test *ar_track_alvar*, then check detection of initial AR-tag correctly (i.e. /ar_pose_marker topic):

   $roslaunch ar_track_alvar ar_track_depth.launch

4. Test move_base without building static map and amcl, then publish 2d nav_goal in rviz and see the planning results:

   $roslaunch spark_navigation move_base_mapless_demo.launch

***5**. Alternative to 4., you can follow what you've learnt on Lab5 and built a map of the environment with any methods,then use amcl and move_base to navigate in the room. Refer to Lab5 handout for this alternative.