ROB-GY 6203 Final Exam (Fall 2022)

Rushi Bhavesh Shah

Submission Deadline: NYC Time 16:00, December 18, 2022
Late submission penalty: -3 points for every 5 minutes past the deadline
Submission URL (must use your NYU account): https://forms.gle/PnUB1HDdhuH4rpDU7

1. Please rename the **.pdf** generated by this LaTex file to **yourNetID.pdf** before submission. This .pdf file will be the main document for us to grade your exam. If you wrote any code, please zip all the **code** together and **submit a single yourNetID.zip file**. Name the code scripts clearly or/and make explicit references in your written answers. Do NOT submit very large data files along with your code!

2. Please typeset your final exam solution in LaTex/Overleaf using this template. You should have learned how to use LaTex/Overleaf by now via our homework! **Do NOT submit a hand-written report!** If you do, it will be rejected from grading.

3. Do not forget to update the variables "yourName" and "yourNetID".

4. Everyone **can submit multiple times** to the Google form. We will **only grade the last valid submission**, but be careful because a **penalty** might be applied if your last submission is after the deadline.

Contents

SLAM (65 points)

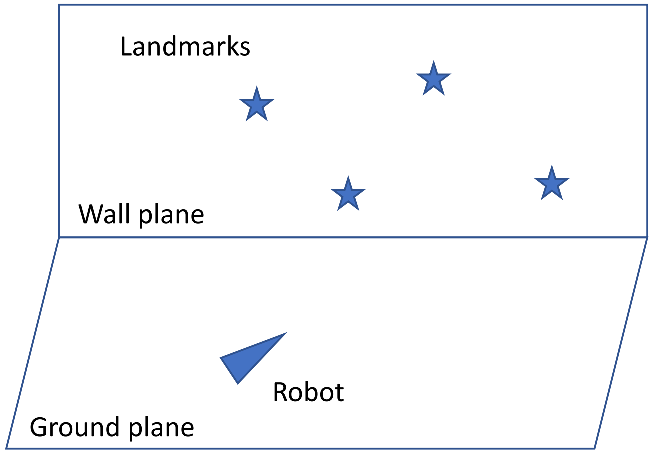

Figure 1: SLAM problem illustration.

A robot can move on a ground plane freely (see Figure 1). At any moment t, the robot can observe 4 unknown point landmarks on a 3D planar wall that is not necessarily perpendicular to the ground and far from the robot. The robot only has one sensor, which is an electronic distance-measuring device that can tell the distance from its current position to any landmark point on the wall with Gaussian error $N(0, 0.01m)$ in its returned distance readings (assuming the device always knows the correspondence no matter where the robot is). The robot's base coordinate frame at $t = 0$ defines the world coordinate frame. The goal of the robot is to localize itself in the world frame, and to measure distances between each pair of those unknown landmarks.

a Describe the scene formally by a set of mathematical symbols. Include a figure to label your notations (you can label the screenshot of Figure 1 directly). (5 points).

Using notations that you have defined above, answer the following:

b To localize the robot and estimate the distances between each pair of landmarks, we need to solve a SLAM problem. What are the states of this system? Explain the meaning of each part/dimension of the state, and express the states using the minimum number of necessary parameters. (10 points).

c Based on your answers above, what is the observation function that can predict measured values given the states of the system? (20 points).

d In order to solve this SLAM problem, what is the minimum number of different positions that the robot needs to be to make measurements? (5 points). Why? (10 points). What if there are only 2 (instead of 4) landmarks? (5 points).

e Based on your answer above, write a python or Matlab code to simulate this scenario numerically, and solve the SLAM problem (you can use **any** APIs in numpy or scipy or Matlab). Report the visualization of both the ground truth scene, and the estimated scene from your solution by matplotlib/Matlab. You can arbitrarily set up the ground truth states of the system. (10 points).

Solutions:

a

In the figure above, landmarks are represented as $[l1, l2, l3, l4]$ and X0 is the initial state of the robot.

b

In the above figure, w is the world/initial frame. The states of the robot are $X_0(x_0, y_0, \theta_0)$, representing x y coordinates of the robot along with the yaw angle. These states of the robot are obtained assuming the robot frame's $x - y$ plane is parallel to the ground plane. If that had not been the case, the $z-$coordinate would also have been

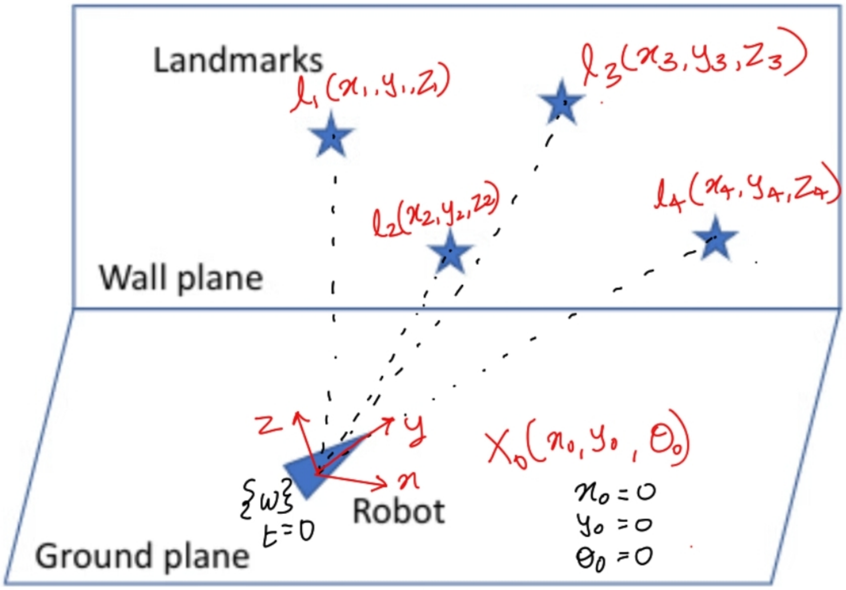

Figure 2: Annotated

considered in the robot's state. However, currently z is considered to be constant and hence 'zero'. In addition to that, the states of the landmarks are represented as x_i, y_i, z_i each corresponding to x, y z coordinates of each. Hence, total states will be **15**. Hence, states would be as follows :

$$\text{states} = [x_0\ y_0\ \theta_0\ x_1\ y_1\ z_1\ x_2\ y_2\ z_2\ x_3\ y_3\ z_3\ x_4\ y_4\ z_4]$$

However, for triangulation we only need 3 landmarks as we have need to solve for three variables. Hence the minimal no. of states required would include the pose of the robot and the 3D coordinates of any three landmarks. Hence, the number of minimal states required would be **12** with states shown as below.

$$\text{states} = [x_0\ y_0\ \theta_0\ x_1\ y_1\ z_1\ x_2\ y_2\ z_2\ x_3\ y_3\ z_3]$$

c

The observation model would include the Euclidean distance of all the landmarks from the robot (considering the origin is at robot). $d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}$ where $z_0 = 0$

d

In order to solve the SLAM problem, the robot needs to measure three distances in order to do the triangulation. For this, the robot need to move to three positions and perform the least-square optimization.

If there are only 2 landmarks, the robot will still have to do three distance measurements for triangulation. This will be done by measuring the distances between the robot and both the landmarks and the distance between both the landmarks. However, it needs to be noted that with just 2 landmarks, the accuracy will be less than that compared to 3 or more.

e

Visual Perception System Design (35 points)

Design the visual perception system for a mobile service robot in any typical indoor office space like 6MTC that can move around on the same floor to clean unlidded office trash cans in the hallways.

a In general, what algorithmic modules are necessary for this visual perception system? List each module with its functionality and its potential implementation as a bullet point (e.g., A 2D path-finding module implemented with the Dijkstra algorithm). Any assumption (such as computers and sensors on the robot) you make to design this perception system should be added after all the bullet points are listed (15 points).

b If you have the freedom to use AprilTag in this project, how would you implement the above system? List the modules among your answer in 1 that would be affected by the use of AprilTag, and explain briefly and concisely how to implement that module using AprilTag (10 points).

c What if AprilTags (and any other fiducial markers alike) are NOT allowed in the system? (10 points)

Solutions:

a

In general, the visual perception system for a mobile service robot in an indoor office space could include the following algorithmic modules:

2D path-finding module: This module would be responsible for generating a path for the robot to follow, based on the 2D occupancy grid map. The path should be optimized to minimize the distance traveled and avoid obstacles. This module could be implemented using algorithms such as Dijkstra's algorithm, A* search, or Rapidly-exploring Random Trees (RRTs or RRT*).

2D occupancy grid mapping module: This module would be responsible for constructing and maintaining a 2D map of the environment, using data from the robot's sensors (e.g., a LIDAR or a stereo camera). The map could be implemented using a grid of cells, where each cell represents the probability that it is occupied by an obstacle or free space. This module could be implemented using a variety of algorithms, such as SLAM (Simultaneous Localization and Mapping) or FastSLAM.

Object detection and classification module: This module would be responsible for identifying and labeling objects in the environment, such as trash cans, chairs, desks, etc. This could be achieved using machine learning algorithms, such as Convolutional Neural Networks (CNNs), that are trained on a dataset of labeled images.

Visual servoing module: This module would be responsible for using visual feedback to control the robot's motion, so that it can follow the path generated by the 2D path-finding module and avoid obstacles. This could be implemented using algorithms such as image-based visual servoing or feature-based visual servoing.

Assumptions: It is assumed that the robot is equipped with a camera and a LIDAR sensor, and has a computer on board for processing the data from these sensors and running the algorithmic modules.

b

If the use of AprilTag is allowed, it can be used to enhance the object detection and classification module and the 2D occupancy grid mapping module of the visual perception system for a mobile service robot in an indoor office space. The robot can be programmed to recognize AprilTag markers, that can be attached to objects in the environment, and use this information to classify objects as trash cans, for example. The robot can also use AprilTag markers to more accurately localize itself on the map as it moves through the environment, which would allow it to build a more accurate and detailed map of its environment. Using AprilTag in this way could improve the performance and accuracy of the visual perception system for the robot.

To implement the object detection and classification module using AprilTag, the robot could be equipped with a camera that is capable of capturing images of the environment. The robot could then use an AprilTag detection library, such as the one provided by the AprilRobotics group, to detect and recognize AprilTag markers in the images. The robot could use this information to classify the objects that are associated with the AprilTag markers.

c

If AprilTag (and other fiducial markers) are not allowed in the system, the object detection and classification module could be implemented using other machine learning algorithms. For example, the robot could be trained on a dataset of images that are labeled with the objects of interest (e.g., trash cans). The robot could then use a CNN or another type of machine learning algorithm to classify the objects in the environment based on their visual appearance. Alternatively, the robot could use other types of sensors, such as a LIDAR, to detect and classify objects in the environment.