University West
Department of Engineering

SLAM Project 2

Matlab MAP builder

Author
SOLOMON GUGSA

January 8, 2021

Contents

0.1 Introduction

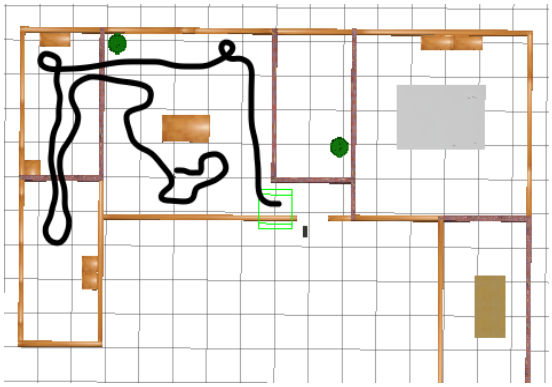
The goal of this lab project work is to have a deeper knowledge about SLAM (Simultaneous Localization and Mapping). The emphasis on this project is on creating a map of a virtual simulation of a house using SLAM builder in MAT-LAB.The layout of this report lists questions from the main lab document followed by answer and practical discussions that arises during the lab work.

0.2 Task 1

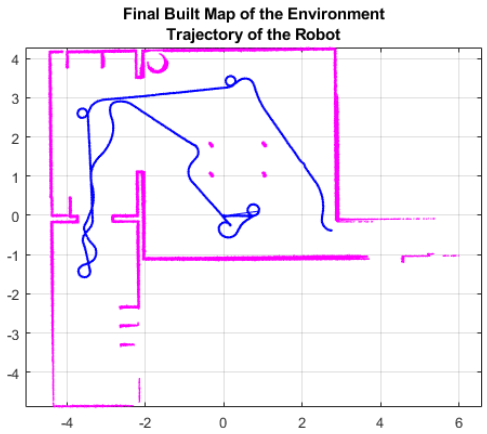
Create a map of the house world available in the virtual box supplied using MAT-LAB. Three different rooms are enough.

0.2.1 Sub-Task A and B

Record a Rosbag while you teleoperate the robot around the house and use the SLAM Map Builder in MAT-LAB to create an occupancy map based on the bag data.



(a) Approximate path



(b) Map using SLAMALGO

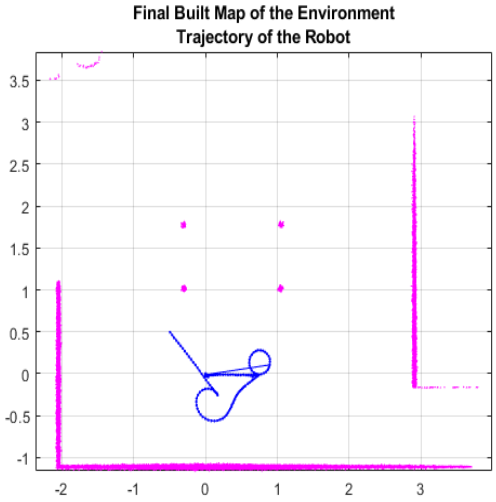
Figure 1: SLAM

Figure (a) shows relative environment being mapped and approximate route of the robot path in black line. The lines are hand-drawing and it shows an approximate path when the robot teleoperated from gazebo. The reason for using hand-plotting is that it's not possible to plot robot path in gazebo. Figure (b) shows the final built map using slamAlg and also including full path in blue line. It was possible to scan at-least the first three rooms and import scan-data using rosbag.

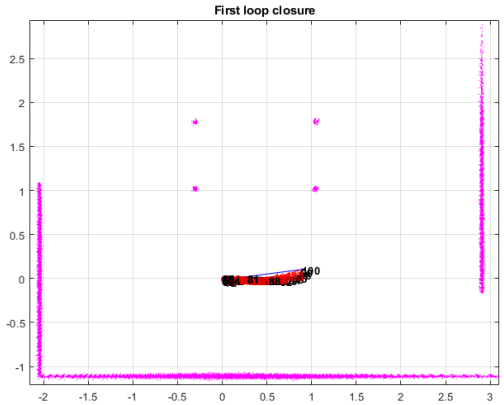
0.2.2 Sub-Task C

Explain what incremental match is and what loop closure is.
The goal of Task 1 is to build a map of the simulation environment imported through rosbag and eventually retrieve the trajectory of the robot.

Slam algorithm uses incremental matching to process and builds a pose graph that connects Cartesian coordinates retrieved from lidar scans. To prove this, the first 100 scans



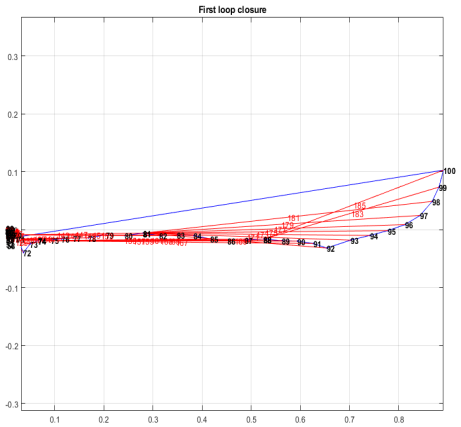
(a) Approximate path



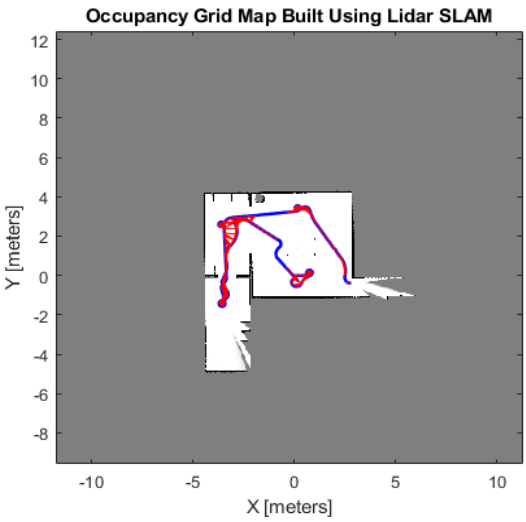
(b) Loop Closure

Figure 2: SLAM LoopClosure and Incremental Match

plotted in fig (a) above by avoiding scans which has too small distance between one another. Incremental Additions of scans is observed in the map building process during the simulation. The robot also recognise prior poses through evaluating identical scans:- a process called incremental matching. This process will follow a loop Closure method to identify previously visited place eventually optimizing the path on the map. Fig (a b) shows this process where in fig (a) shows the first 100 scans numbered, and Optimized path in blue using loop closure edges in red. Fig (b) shows a general picture of the map using occupancy grid mapping.



(a) First LoopClosure



(b) Over all Loop Closure Occgridmap

Figure 3: SLAM LoopClosure

0.3 Task 2

Create a probabilistic road-map in MATLAB and find a collision free path.

0.3.1 Sub-Task A and B

Use MATLAB to modify the map. Create a probabilistic roadmap in MATLAB based on the occupancy map. What is a probabilistic roadmap

A probabilistic roadmap (PRM) is a network graph of possible paths in a given map based on free and occupied spaces[1]. The idea behind PRM algorithm is basically, to explore feasible paths around a number of obstacles located in the map. This could be done using mobileRobotPRM object in matlab. For-example, a simple road-map with sixty nodes using previously generated Map created below in figure (4). The OccupancyMap is modified using binaryOccupancyMap and it resembles that of the original map.

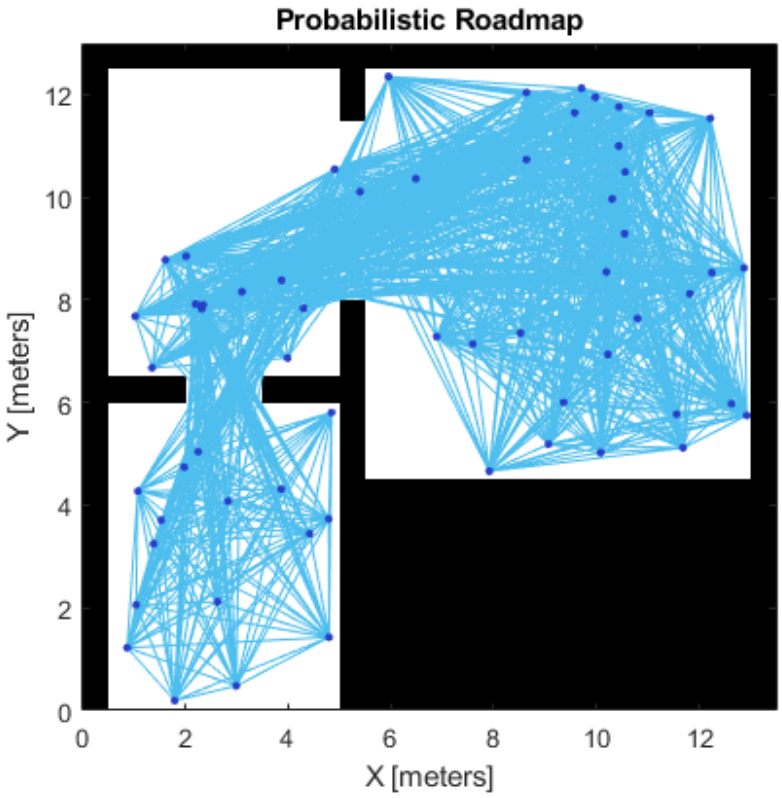
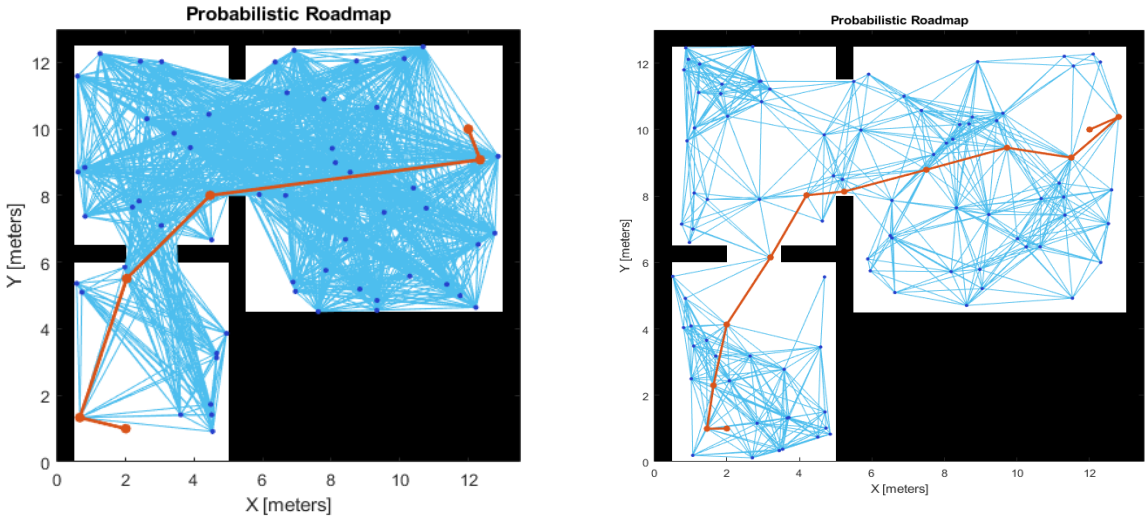


Figure 4: Probabilistic roadmap with 60 nodes

0.3.2 Sub-Task C

Create a MATLAB program that asks the user for an initial and final point and then calculates and plots the collision-free path in MATLAB.

The user is prompted a start and end location. The optimized path is calculated and shown in figure (a and b). Tuning the connection distance results a better path and more directed route for the robot. Adding more number of nodes also improve path calculation by increasing the number of feasible path but yields more computational complexity.



(a) Collusion free path between start and end location, connection distance set to inf

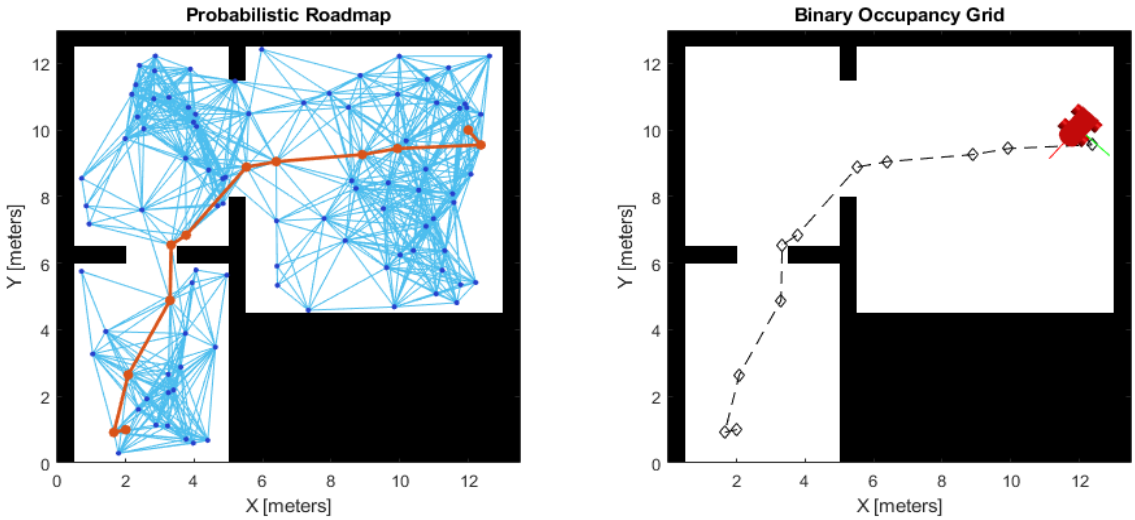
(b) Collusion free path between start and end location with a distance only 3

Figure 5: Collusion free path with a start and end location prompt by the user

0.3.3 Sub-Task D

Create a controller in MATLAB to move along the planned path.

The following result shows a differential robot simulator controlled using a simple matlab application. Way-points are defined using the predetermined path created in PRM (fig a). The simulator follows exactly similar path until it reaches predefined goal.



(a) Optimized Path

(b) Differential Robot Controller

Figure 6: Robot simulator

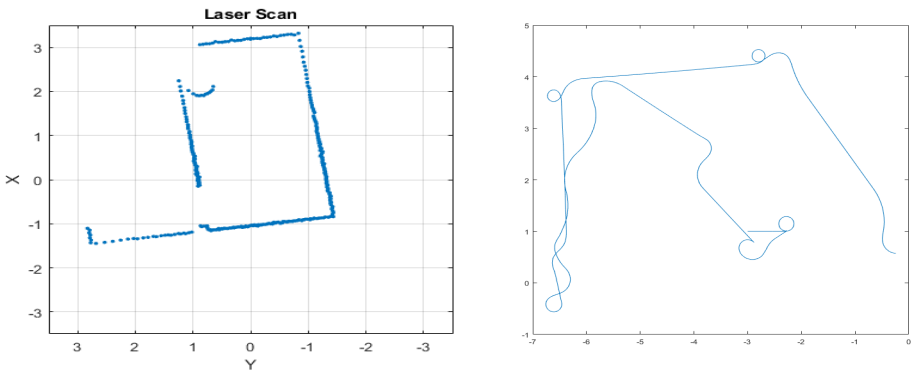
0.4 Task 4

Create the map in MATLAB by reading the scan and odometry topics in real-time, without using the SLAM Map Builder and rosbag data

The following line of code produce a real-time map without using SLAM. I was able to plot real-time lidar-scan data but retaining a final map wasn't possible. Figure B shows optimized trajectory followed by the robot in gazebo.

```
%-----------------------------Solomon Plot map without slam--------------%
clc
clear
close all

bag = rosbag('map.bag');
bSel = select(bag,'Topic','/odom');

%plot robot trajectory
ts = timeseries(bSel, 'Pose.Pose.Position.X', 'Pose.Pose.Position.Y');
plot(ts.Data(:,1), ts.Data(:,2));

%Connect to master
ip = "http://192.168.233.130:11311"; % Modify with your ROS Master URI Address
rosinit(ip);

%get scan data & plotting
sub = rossubscriber('/scan');
scan = receive(sub);
i = 1;
while i < 3

    scan = receive(sub);
    plot(scan)
    i = i - 1;
end
%--------------------end-------------------------------------------------%
```

Figure 7: LOC Real-time Map Plotting and trajectory



(a) Real-time Map Scan (b) Optimized Trajectory

Figure 8: Robot Path

0.5 Reference

1. Kavraki, L.E., P. Svestka, J.-C. Latombe, and M.H. Overmars. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Transactions on Robotics and Automation. Vol. 12, No. 4, Aug 1996 pp. 566—580. https://se.mathworks.com/help/robotics/ug/probabilistic-roadmaps-prm.html
2. Copyright 2014-2019 The MathWorks, Inc. https://se.mathworks.com/help/robotics/ug/path-following-for-differential-drive-robot.html