VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CONTROL SYSTEMS – ECE2010

PROJECT FINAL REPORT

MAPPING ROBOT SIMULATOR

- **TEAM MEMBERS:**

1. T. DINESH RAM SAI - 18BEC0042
2. ABHISHEK HARIKRISHNAN - 18BEC0909
3. SRIKAR ANUDEEP REMANI - 18BEC0555

SUBMITTING TO: Prof. RAJESH R. Sir

SLOT: F1+TF1

SEMESTER: FALL SEMESTER 2020-21

Table of Contents

ABSTRACT

Mobile robots can be used for transportation tasks, surveillance, or cleaning. Increasingly, they play an economic and important role also in the every industry amidst this pandemic time and are known for to help combat the spread of the **novel coronavirus**. Mobile robots are suited to disinfecting facilities, providing surveillance, and delivering goods. In control of mobile robots, precision plays a key role in path tracking.

Currently, the path planning problem is one of the most researched topics and has any many advantages and disadvantages. That is why finding a safe path in a cluttered environment for a mobile robot is an important requirement for the success of any such mobile robot project. Hence The aim of this project is to search a safe path for the mobile robot, to make the robot moving from a starting position to a destination position without hitting obstacles.

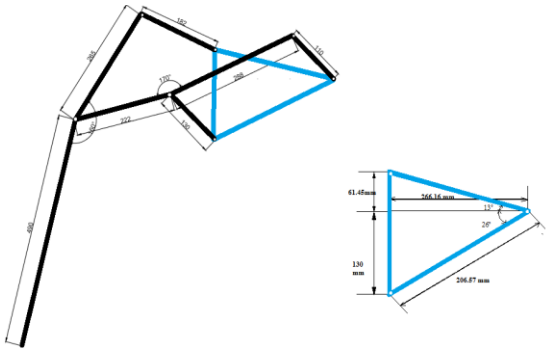
Finally, well simulate it and are going to show simulation result that the developed approach is a good alternative to obtain the adequate path and demonstrate the efficiency of the proposed control law for robust tracking of the mobile robot.

We have found that stepper motors are more accurate for path tracking than normal DC motors with wheel encoders and one can obtain the implicit coordinates of the robot in runtime more precisely. Getting the precise coordinates of the robot at runtime can be used in various SLAM and VSLAM techniques for more accurate 3D mapping of the environment.

INTRODUCTION

Robotics has become an indispensable part in our lives from strategic military operations to mundane applications of household and cleaning. Among the various applications of robotics, path tracking has been one of the most emerging fields in terms of projects. **Legged mechanism** can be used for most precise mapping of the trajectory.

A leg mechanism is a mechanical system designed to provide a propulsive force by intermittent frictional contact with the ground. This is in contrast with wheels or continuous tracks which are intended to maintain continuous frictional contact with the ground.However controlling and operating legged mechanism robots are much more complicated than wheeled robots. Apart from the control systems legged mechanism robots are relatively slower. Differential drive odometry techniques with DC motors and encoders are used for path tracking algorithms.



Wheel encoder output gives velocity in which the two wheels move. The bot can be moved along a certain curve by giving the two wheels different velocities. Such a bot will have problems moving along a straight line since both the wheels are independent. Moving both wheels with the same velocity is a challenge.

So our project is based on a better approach of using stepper motors rather than DC for precise rotation of the wheels and in turn the orientation of the bot. To keep track of orientation of the bot at a particular point a magnetometer is also used.

The error caused in the actual orientation of the bot given by magnetometer and the orientation achieved due to movement of the stepper motor is also taken into consideration. In traditional differential drive odometry readings from the encoders are sent as a feedback to the system to determine the motion of the wheels and the position of the bot. In our stepper model of the robot stepper motors are used for the precise motion of the bot.

The wheels can be independently controlled to change the orientation of the bot at any given instant of time.

LITERATURE REVIEW:

(Part of the Introduction Section)

The following are the Literature **Review Works** for this project. For more information, kindly refer to the Reference Papers given at the end of the Report.

1. In the reference Papers they have discussed about the Stepper Motors and Solidworks Modelling. From which we drew our knowledge and we have extended our gained knowledge by using DC Motors instead of Stepper Motors and using Trajectory Mapping instead of Traditional Solidworks Modelling in Matlab

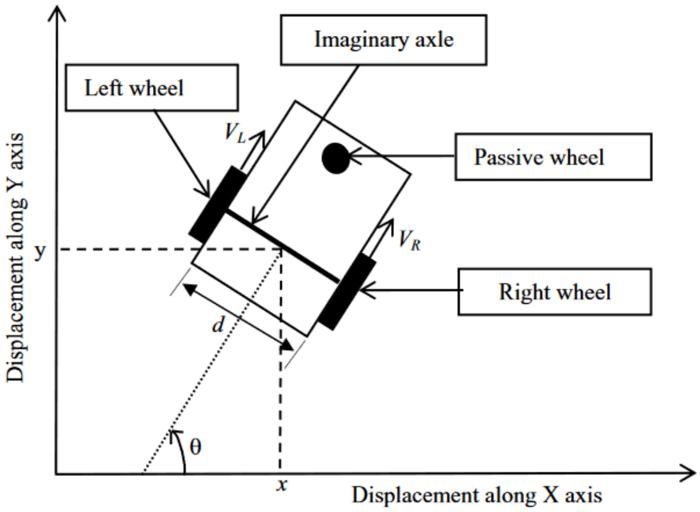
2. And they have used the Control Algorithm which is specific to each Trajectory. For example, If we use Control Algorithm for Triangular Trajectory, then we have to make some changes in it, while using for

Circular or Rectangular Trajectory. But instead of doing this, we have used a Map which can be used to define path and hence we developed a control algorithm for defining the path in a map, instead of sticking to any specific trajectory. We can even change the map, and still can define the path in the map without changing the main Control Algorithm.

SYSTEM MODEL

The following are the specs of our developed System Model:

- Our design is a simple three wheel configuration in which the rear wheels and front wheels are controlled using DC Motors. This configuration helps for smooth point rotation and lateral movement of the bot.

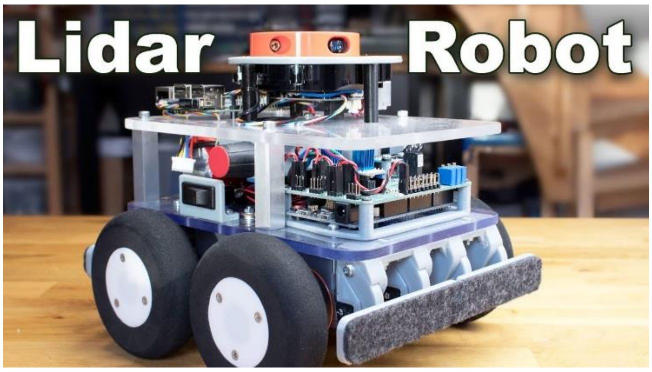


- These mapping robots have independently actuated right and left wheels. There may be an another one passive wheel in the front for smooth movement of the robot. The motion of the robot is controlled by controlling the velocities and accelerations of left and right rear wheels, VLandVR. Rolling of rear wheels without slipping will make sure that the velocity of the robot to be orthogonal with respect to the axle connecting the rear wheels.

- The weight ,dimensions and path for each part is manually entered so that the centre of gravity is automatically assembled by Matlab.
- In this, project we have considered continuous and differentiable trajectory functions. Trajectories/Paths expressed in terms of y=f(x) are sampled/divided along the x axis. Continuous curved trajectories can be sampled via simple commands in MATLAB using **Robot Operating System(ROS)** Interface.
- We also made sure that t every step the bot is programmed to change its orientation depending on the environment rather than just changing at the sample points.

METHODOLOGY

This system is split up into 2 parts – the LIDAR sensor which is used to detect the surroundings and the bot, which reads the signals which the LIDAR sensor passes to the bot. The bot takes the readings as a matrix containing data points and uses them to 3 calculate the distance between the simulated wall and the robot.



Using this the bot can then plot the map figure. In the simulation, we are using the ROS interface to simulate the full working of a mapping robot by simulating the LIDAR sensors as well as the bot's computer which calculates the distances and plots the new map. The signals acquired are in the form of a matrix containing the distances that each laser ray from the LIDAR sensor senses from the surroundings.

The MATLAB built-in functions calculate the distance of the bot from the surroundings by calculating the time taken by the laser ray to go and return. This time and consequent distance are then returned as a matrix which is then used as data points for the plotting. The analog signals received by the sensor in this case are the returning laser rays which were sent from the sensor itself.

PERFORMANCE & ANALYSIS

1. We have loaded the Map and Manually Defined the **PATH** inside the Map. We could also change the path inside map, by just changing the Coordinates

2. The **CURSOR** is the Mobile Robot and **BLUE LINES** are the sensing rays from **LIDAR** sensor.

3. Initially, the bot starts at the **STARTING POINT** and will go through the entire path. As the Bot is moving in the path, the **OCCUPANCY GRID** will be mapping.

4. **OCCUPANCY GRID** is being mapped, while the **UPDATE** is increasing. **UPDATE** is the interval, in which the bot will move from one interval to another interval.

5. We have defined the value of **UPDATE** as **50** which means, if bot moves from interval to another, the update reading will increase to 50, which is the number of **LIDAR READINGS** that can be taken during a **UPDATE**.

6. After the bot, reaches the destination point, the plotting/Mapping of the OCCUPANCY GRID is completed

7. In the completed occupancy Grid,

WHITE REGION – High Probability Region, that a BOT can go there.

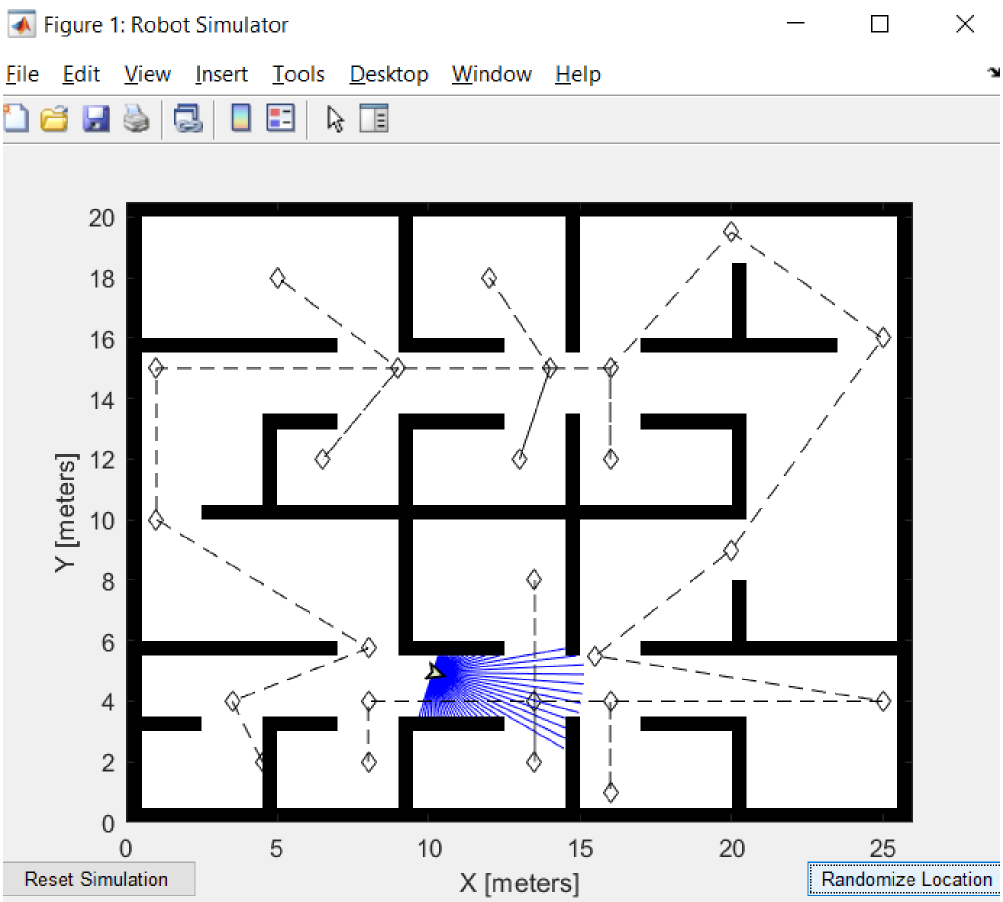
GREY REGION – Low Probability Region, that a BOT can never go there

8. We can verify the result, by clicking the **RANDOMIZE LOCATION** Button in the bot map graph. Which will always shifts to the White Regions, where the probability is high that a BOT can go there.

9. Similarily, it also never shifts the Bot to the Grey Regions, where the Probability is low that a BOT can never go to that regions.

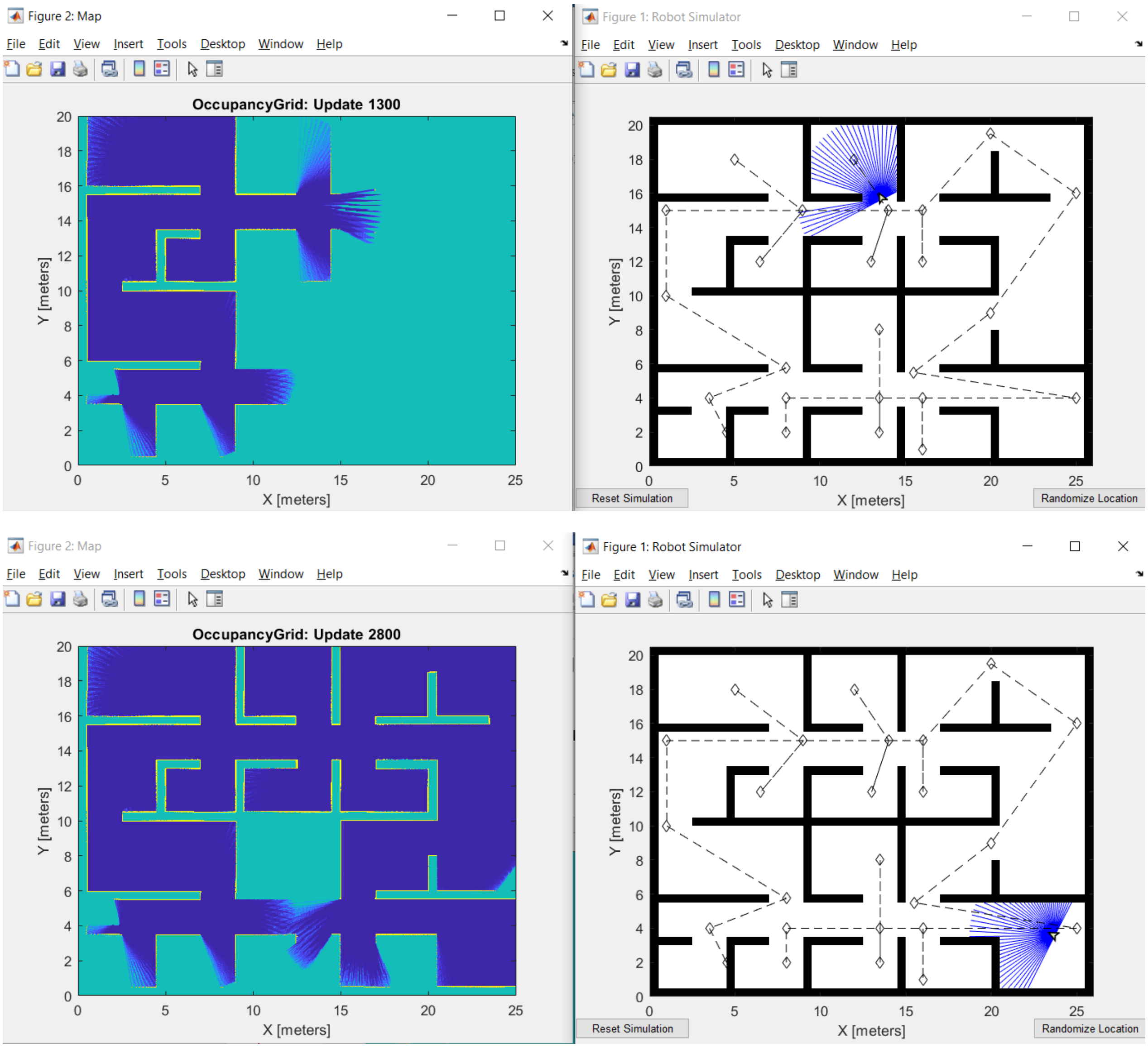
RESULT & DISCUSSION

The following are the outputs of the simulation:

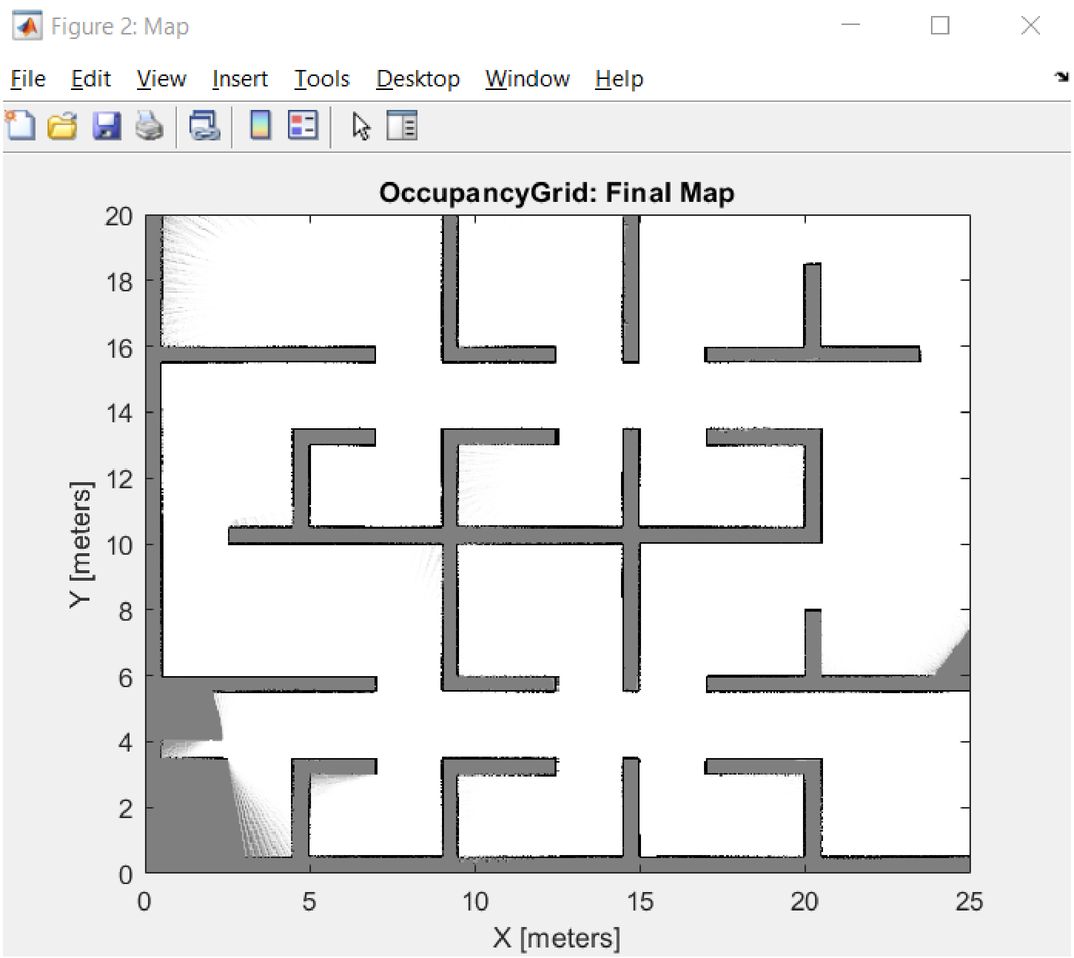


As we can see, we have loaded the Map and also defined a path in it.(**UPPER FIGURE)**

We can see that the Occupancy Grid is filling/Mapping while the Bot is moving through the path, by the following figure.



The Final OCCUPANCY GRID after the bot finishes travelling through Path is:



WHITE REGIONS – High probability

GREY REGIONS – Low Probability

PROJECT CODE:

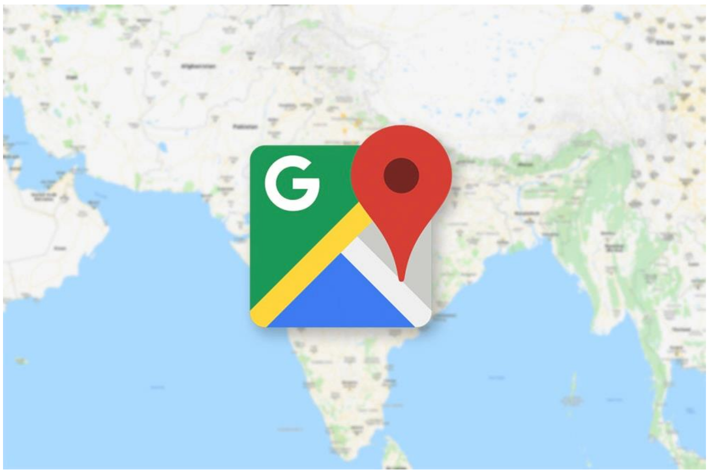
CONCLUSION

With this project, we were able to simulate the working of a mapping robot by simulating the movement of such a robot through a room and by using simulated LIDAR sensor values we were able to calculate distances and recreate the room in another map. By adjusting the speed and waypoints, we can simulate methods to efficiently scan any given room.

FUTURE SCOPE

1. In some applications, such as self-driving cars, precision localization is often performed by matching current sensor data to a high definition map of the environment that is created in advance. If the a priori map is accurate, then online maps are not required. Operations in highly dynamic environments, however, will require dynamic online map updates to deal with construction or major changes to road infrastructure.

2. Because of limited Hardware Resources, **we took the limits of X and Y coordinates as 25 and 20.** In real Life, with the help of high speed hardware resources, we can integrate this project with GOOGLE MAPS API along with Machine Learning, so that we could directly load Google Maps as the Map and path can be defined by using Machine Learning/AI



REFERENCES

1) Ankit Deo, Raaghavi Ravisankar. Differential Drive odometry for two wheeled mobile robots. IJRSET.

2) Trajectory tracking and control of differential drive robot for predefined regular geometrical path Global Colloquium in Recent Advancement and Effectual Researches in Engineering, Science and Technology (RAEREST 2016)