**LAWN-MOVER ROBOT**

THE PROJECT REPORT

***Submitted by***

BL.EN. U4AIE19028 K.Vishnu Sainadh

BL.EN. U4AIE19034 K.Satwik

BL.EN. U4AIE19066 V.Ashrith

**For the course**

**19AIE213- Robotic Operating System and Robot Simulation**

**Guided and Evaluated by**

**Mr.Nippun Kumaar A. A.**

**Asst. Prof (SG),**

**Dept. of CSE**



AMRITA SCHOOL OF ENGINEERING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

**BANGALORE 560 035**

May-2021

**AMRITA VISHWA VIDYAPEETHAM**

Title of the project

**LAWN-MOVER ROBOT**

**Acknowledgement**

The satisfaction that accompanies successful completion of any task would be incomplete without mention of people whom made it possible, and whose constant encouragement and guidance have been source of inspiration throughout the course of this project work.

We offer our sincere pranamas at the lotus feet of “AMMA”, MATA AMRITANANDAMAYI DEVI who showered her blessing upon us throughout the course of this project work.

It is a great pleasure to express our gratitude and indebtedness to our project guide Mr. Nippun kumar A A . Assistant Professor, Department of Computer Science and Engineering, Amrita School of Engineering, Bangalore for your valuable guidance, encouragement, moral support and affection throughout the semester.

TABLE OF CONTENTS:

1) PROBLEM STATEMENT…………………………….5

2) EXPLANATION …………………………………….. 5

3) IMPLEMENTATION………………………………… 5

4) RESULT……………………………………………… 8

5)REFERENCES ………………………………………...9

TABLE OF FIGURES:

Fig 3.1 : Points Generated in python script ……………………………….6

Fig 3.2 : RQT Graph ……………………………………………………….7

Fig 3.3 : Block Diagram ……………………………………………………7

Fig 4.1 : Robot avoiding the obstacle ……………………………………..8

Fig 4.2 : Robot moving in a straight line …………………………………... 9

**Problem Statement:**

Design and implement an lawn mowing robot using a custom built differential drive robot. The robot should traverse to all the locations of the environment, the boundaries of the environment should be exclusively given as inputs in terms of boundary coordinated, also assume some static objects placed in the environments. The total number of laps should be taken as input. The type of sensors can be team’s choice but the reason to use the sensor should be clear in the report and the presentation.

**Explanation:**

We should design a differential drive robot which should traverse to all the locations of the lawn within the boundary limit by avoiding the obstacles placed in the lawn. We will use sensors for detecting various obstacles and go ahead by avoiding them. We will be using a node for detecting obstacles and it will be avoided by moving around the obstacle which will be done by another node and then we will be simulating the lawn moving robot in Gazebo using the urdf file which is used for building the differential-drive robot.

**IMPLEMENTATION**:

Differential drive is a control system that uses only two motorized wheels to control a robot. The word differential refers to the speed difference between the two wheels on either side of your robot, which determines the robot's turning speed. It consists of two wheels mounted on a common axis, each of which can be moved forward or backward independently. We used a differential drive robot with two wheels, one fixed base connection to support the wheels, and one caster wheel in front, and we used a hokuyo sensor that is fixed at the top because obstacles can collide with the robot if it is fixed at the front.

First we have written a python script which takes the boundaries and number of laps as input. It will generate the points through which the robot has to travel within the boundary such that it covers all locations of the environment. We have built our own environment with some custom obstacles in it. So, first the robot assumes an imaginary line between the robots position and the goal’s position and then it tries to move along the imaginary line. Robot checks whether there is an obstacle ahead of it.

If there is an obstacle it changes its state to obstacle avoidance, this service makes the robot turn left and it will try to keep obstacle to the right side of robot. And then again it checks for obstacle if the obstacle is present it will avoid as above else it will get back to the imaginary line.

If there is no obstacle the robot sets its yaw to the next desired position and then it moves ahead along the imaginary line. It always checks for the obstacle and whenever the obstacle is found it changes its state to obstacle avoidance.

This process will work for each point that has been generated initially. After covering all the points for given number of laps the robot will stop.

Here we have used hokuyo sensor for our robot to detect the obstacles. The Hokuyo sensor is a small, lightweight, and extremely accurate obstacle detection sensor developed specifically for indoor robotics. The Hokuyo laser range finder is one of the Japanese LiDAR manufacturer's longer-range versions. It can detect objects up to 120 meters away in ideal conditions. We have a 180-degree scan and a range of up to 30 meters in our project. We're working with LiDAR, which stands for light detection and ranging and has been around for quite some time. It measures distance by timing the movement, or flight, of laser pulses as they detect objects and return to the source of the laser.

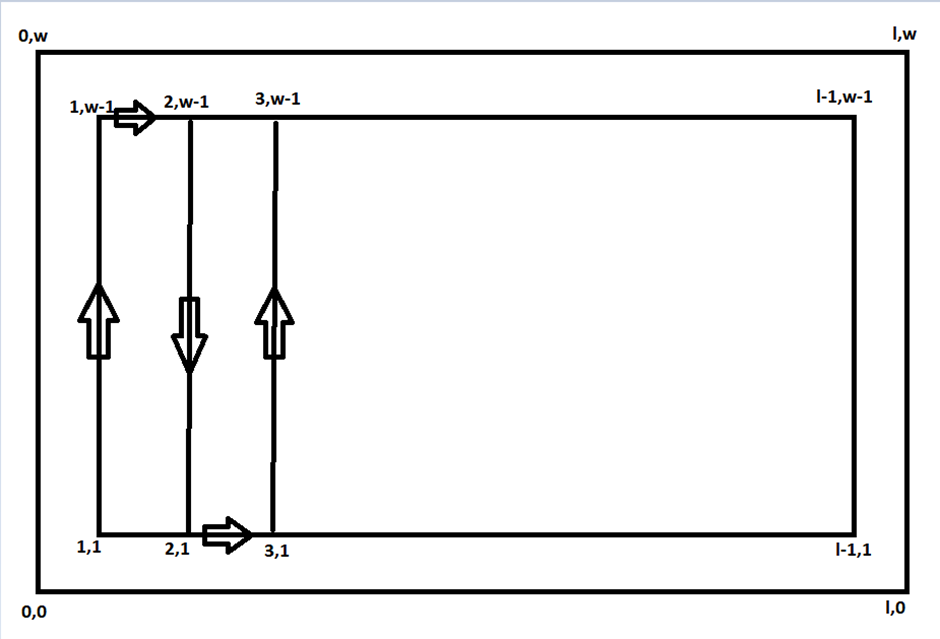


Fig 3.1 : Points generated in python script

The points will be generated depending on the boundary values and the robot will move along the direction as shown in figure and stop after covering all points for given number of laps.

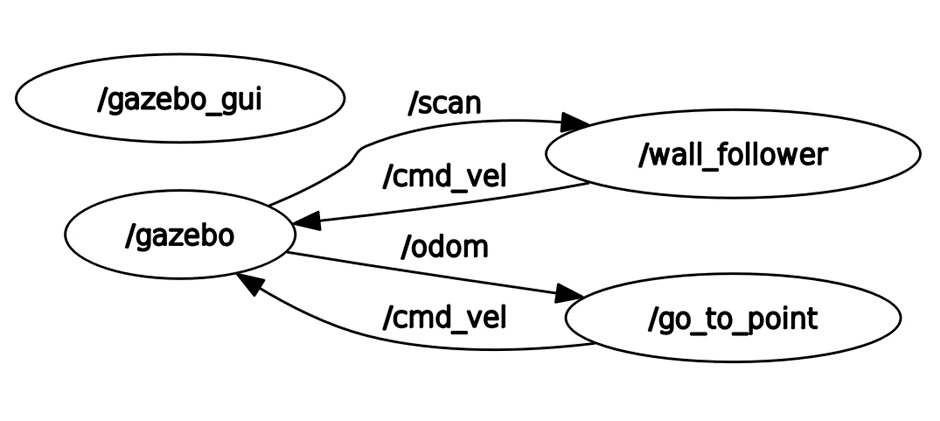


Fig 3.2 : RQT Graph

First Gazebo will be launched, wall\_follower subscribes scan from gazebo and publishes cmd\_vel . Same way, go\_to\_point subscribes odometry(current position) from gazebo and publishes cmd\_vel.

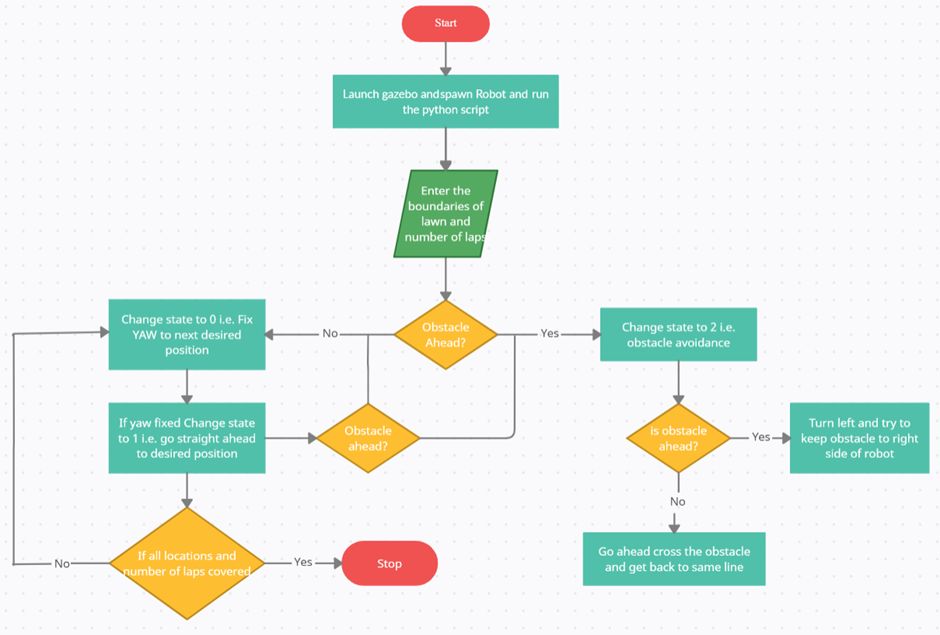


Fig 3.3 : Block Diagram

**RESULT :**

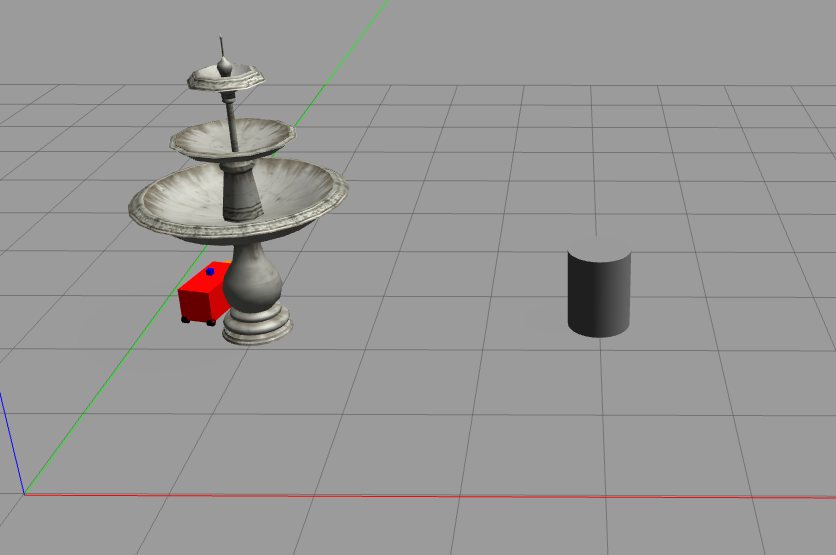


Fig 4.1 : Robot avoiding the obstacle

In this snapshot we can see that the robot always try to keep the obstacle to its right and moves around the obstacle and then comes back to its line to reach its desired position.



Fig 4.2 : Robot moving in a straight line

Here there is no obstacle in front of robot so it moves in a straight line to its desired end position.

**REFERENCES:**

1. [**https://cs.gmu.edu/~ashehu/sites/default/files/cs485\_Fall2013/ShehuLecture02.pdf**](https://cs.gmu.edu/~ashehu/sites/default/files/cs485_Fall2013/ShehuLecture02.pdf)
2. **https://www.allaboutcircuits.com/projects/how-to-build-a-robot-follow-walls/**