WALL FOLLOWING ROBOT USING AWS ROBOMAKER.

1.**INTRODUCTION.**

1.1.OVERVIEW

AWS RoboMaker is a service that makes it easy to create robotics applications at scale. AWS RoboMaker extends the Robot Operating System (ROS) framework with cloud services. This includes AWS machine learning services. It includes monitoring services. It even includes analytics services. These combine to enable a robot to do several things on its own. Stream data, navigate, communicate, comprehend, and learn. AWS RoboMaker provides a robotics application development environment. It provides a robotics simulation service, which speeds application testing. You can easily create hundreds of new worlds from templates you define using Simulation WorldForge It provides a fleet management service so you can deploy and manage applications remotely.

1.2.PURPOSE

Technologies for autonomous robots and self-driving cars have been rapidly advancing. Their development often relies on building application-specific simulation environments, and using the information to train reinforcement learning (RL) models.

A wall following robot is designed to move along a wall without hitting it. It has obstacle detection sensors mounted on the body which detects wall and drive DC motors attached to the wheels such that the robot keeps moving along the wall.

2.**LITERATURE SURVEY**

The basic concept of this project report is to design robots which can move and avoid any obstacle on its way without human guidance or control.

It deals with the geometric relationships that govern the system. It develops a relationship between control parameters and the behavior of a system in space.

The robot travels along contour of the object with a certain distance. This strategy can be very helpful when a robot is stuck in a deadlock.

2.1.EXISTING SYSTEM

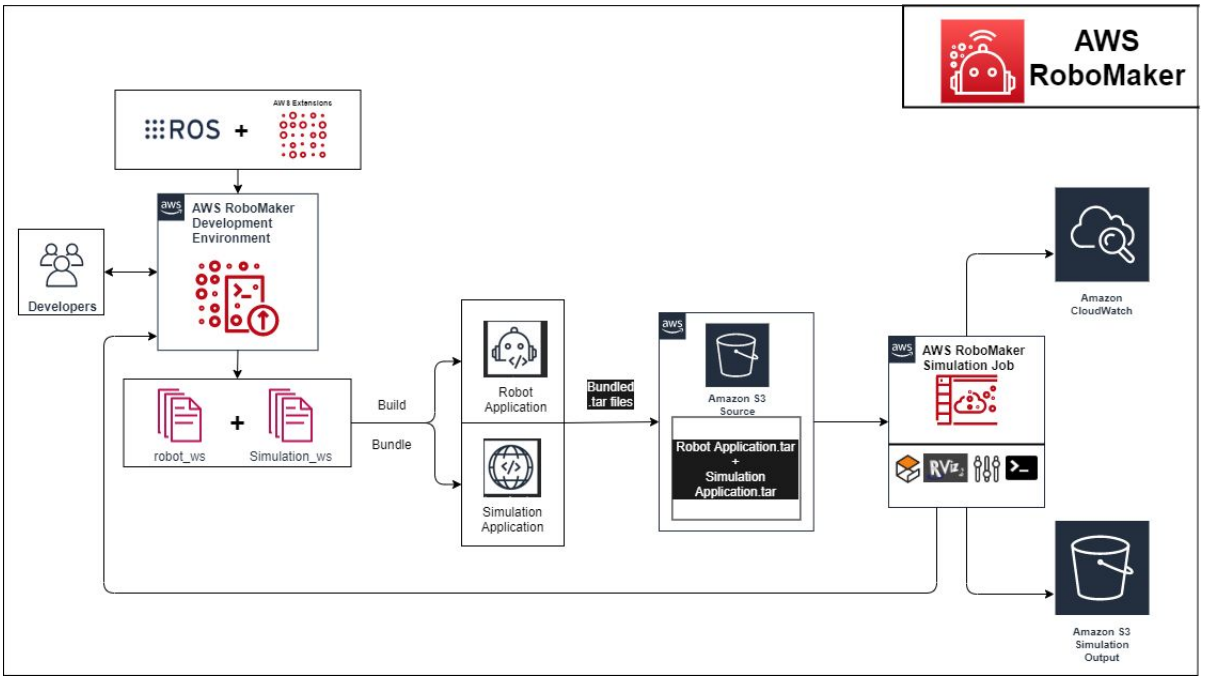
In the existing system, they are used in a robot which follows color line neglecting the white area it is called a Line Follower Robot. This robot consists 2 DC motors which are connected in form of H-bridge connection with Driver IC. IR sensor is used to detect that Color if it is black or Blue means it is taken as 1 otherwise it takes like 0. It follows that dark color lines only.

2.2.PROPOSED SYSTEM

In the existing system, they use IR Sensor to find out that dark region robot travels on that dark region only. In this system, we change that sensor Instead of using IR sensor we go for Ultrasonic sensor Range detection sensor or Laser based sensor. It gives distance values based on the distance value robot will move. We only fix the condition for robot traveling if the distance value is less than some preset value it will follow the wall unless it changes its position by itself. This is a basic prototype which can be further developed for applications like automatic window glass cleaner, floor cleaner and for some surveillance systems.

3.**THEORETICAL ANALYSIS**

_**3.1.**BLOCK DIAGRAM



3.2.HARDWARE / SOFTWARE DESIGNING.

AWS RoboMaker is a service that allows you to quickly develop, test, and deploy robot applications. The Steps to be followed are :

1.Robotics Development with AWS RoboMaker
2.Create a ROS Development Environment
3.Create a Robot Application
4.Develop Simulation and Testing Data
5.Fleet Management and Deployment.

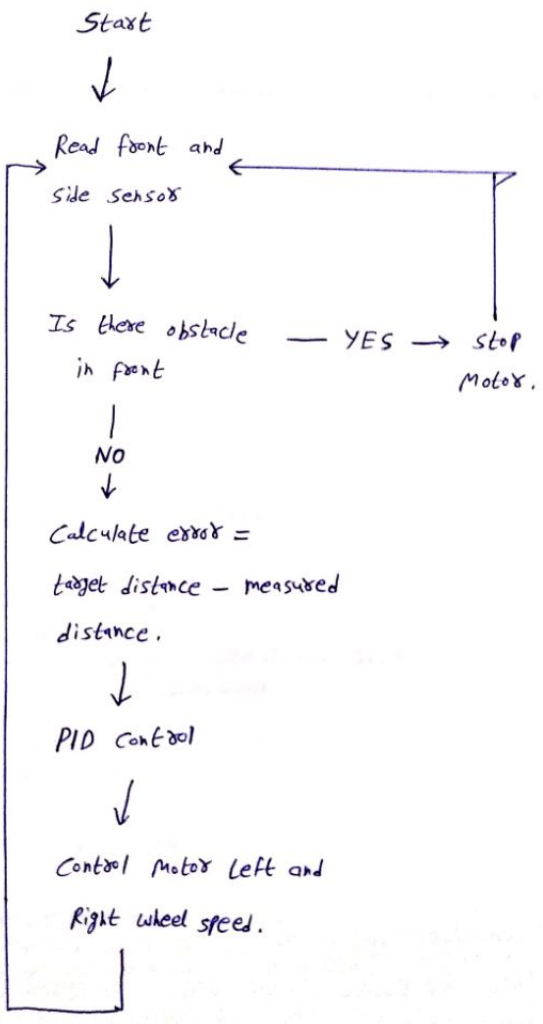
An RL algorithm will help navigate the agent to reach to the GOAL without bumping into a wall. The agent has a 360-degree surround lidar scanner (360 points x 5 fps) so that it monitors the distance from the surrounding walls all around. The lidar data are used to describe a state at a given step within an episode. The agent makes a decision out of 5 different actions i.e. turn left, turn right, move straight, steer to the left, and steer to the right.

4.**EXPERIMANTAL INVESTIGATION.**

When the robot is launched in gazebo simulator, in a maze world, it starts moving in an independent path. Until it's sensors detects a wall.

Once the sensors detects the wall, the robot starts following the wall. Here the robot creates a path in parallel to the wall i.e moving along the wall, so it does not collide the wall in it's pathway.

5.**FLOW CHART.**



6.**RESULT.**

The result is when the robot is placed in a maze environment it is able to solve the maze by following the wall.

7.**ADVANTAGES AND DISADVANTAGES**

In any robotics application, the deployed robot has to navigate from source to a destination for performing tasks.

This helps the robot to navigate to the target site.

In some rooms you could set a bot in the middle and it could estimate the dimensions without moving at all.

But, only a single wall following robot could not establish anything, it has to be integrated with working robots, to help them navigate.

8.**APPLICATIONS**

This is a basic prototype which can be further developed for applications like automatic window glass cleaner, floor cleaner and for some surveillance systems.

One of the application is automatic navigation employed in wall following robot. It is used for parallel parking of vehicles and auto detection of obstacles

9.**CONCLUSION.**

By using the simple components like sensor, driver circuit and DC motor we can design one robot it performs wall following operations. By using these basic components, we can design various robots like Obstacle Avoidance. This is the basic Robot employed in a wide variety of applications.

10.**FUTURE SCOPE.**

In any robotics application, the deployed robot has to navigate from source to a destination for performing tasks.
Efficient control of this navigation is a major research challenge in the field. In this paper, an attempt has been made to develop a neural network (NN) based controller for navigation of wall following robot.
The primary focus is to control the robot to take decision of changing direction based on a set of sensor readings, where the sensors are fit around of the waist of the robot (SCITOS G5 robot in this work).
The NN is trained by these sensor readings dataset (a collection of multiple such instances) and predicts the future control strategy. The NN is trained with gradient descent algorithm.
An extensive parametric study has been conducted to set the optimal number of nodes in the hidden layer and the learning rate. The experimental result shows that the proposed algorithm can control the robot with 92.67% accuracy and can take decision within 1 second.

11.**BIBLOGRAPHY**

1. D.M. Helmick, S.I. Roumeliotis, "Slip-compensated path following for planetary exploration rovers," Advanced Robotics 20 (2006) 1257-1280.

2. R. Manduchi, A. Castano, A. Talukder, L. Matthies, "Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation," Autonomous Robots 18, 81-102, 2005.

3. P. Fabiani, V. Fuertes, "Autonomous flight and navigation of VTOL UAVs: from autonomy demonstrations to out-of-sight flights," Aerospace Science and Technology 11 (2007) 183-193.

4. M. Knudson, K. Tumer, "Adaptive navigation for autonomous robots", Robotics and Autonomous Systems 59 (2011) 410-420.

5. F. Gomez, R. M11kkulainen, "Solving non-Markovian control tasks with neuro-evolution," in: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI-99, Stockholm, Sweden, 1999, pp. 1356-1361.

6. A.K. Agogino, K. Tumer, "Efficient evaluation functions for evolving coordination," Evolutionary Computation 16 (2) (2008) 257-288.

APPENDIX

SOURCE CODE:

The source code files are in this following link:
https://1drv.ms/f/s!AuQVeXxRjMAPiDftHAUixjUN71b8.