

Labyrinth Simulation - Coppliasim

Project Report

Eklavya mentorship programme

At

**SOCIETY OF ROBOTICS AND AUTOMATION,
VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE.**

MUMBAI

JUNE 2020

ACKNOWLEDGMENT

We are really thankful that we got vedant and saharsh as our mentors for this programme.

Luckily we got one of the smartest mentors of SRA , who helped us every moment whenever we got stuck at any point during our tedious journey of this beautiful project .

We would like to especially thank Vedant as despite his extremely packed and busy schedule , he was always present to help us out . Many times there were doubts and errors which were not known to our mentors as well, still they took an extra mile , learned from their seniors or from some other source and made sure we were clear with the idea and concept of whatever is going on.

At the end we feel extremely fortunate to have them as our mentors and hope to receive help from them , even during our future quests and help us grow .

Pratam Jain

8830942773

pratamjain1234@gmail.com

Bhumika Kothwal

9653394644

kothwalbhumika@gmail.com

TABLE OF CONTENTS

NO.	Labyrinth Simulation - Coppliasim	PAGE NO.
	Project overview	ii
1.	INTRODUCTION	i
		1
	1.1 General	1
	1.2 Mathematical formulation	2
	1.3	45
	1.4	58
2.	PROPOSED METHOD	75
	2.1 General	75
	2.2	80
3.	IMPLIED PROCESS	85
	3.1 General	85
	3.2	
4.	EXPERIMENTAL RESULTS AND ANALYSIS	
	4.1 General	
	4.2	
5.	CONCLUSION AND FUTURE WORK	
	REFERENCES	

PROJECT OVERVIEW

- Technology used in this project is V-REP(VIRTUAL ROBOT EXPERIMENTAL PLATFORM).
- This helps us create a virtual robot as the name suggests and helps us in testing and experimenting variations of code.
- This project is to create a robot that follows a line and reaches the end of the maze by traversing the minimum distance possible to reach its destination.

INTRODUCTION

- The project includes VREP and knowledge of basic C language with some basic knowledge of data structures and algorithms and their mechanisms.
- Knowledge ->
Theory of algorithms such as dijkstra algorithm , BFS , DFS and some data structures such as stacks or queues , Graphs and arrays.

METHODS

- This project can be done in VREP simulation .
- Other options available are by using Gazebo . The disadvantage of using gazebo is it is a bit lengthy and complex to understand and also requires us to know about ROSS.
- If someone wants to learn about ROSS then he/she can use gazebo , whereas if someone does not want

to learn ROSS , VREP would be a preferred option.

- Workflow of the project

EXPERIMENTS AND RESULTS

- Procedure ->

- 1] Create a robot in VREP
- 2] write code for line following
- 3] tune in PID for better efficiency
- 4] write code for exploration of maze
- 5] write code for shortest path of maze

6] make sure the robot has decent speed as VREP has a time limit else it stops working (unless this bug has been solved by VREP)(For more info check VREP-copliasim discussion forum).

- It requires around 1000 or more testing cases for tuning in PID (P->proportion ; I->integration ; D->Differentiation) required so that the robot follows the line properly and does not deviate away easily.
- No. of test cases and trials required may vary from person to person depending upon its respective judgements and approximation.
- <https://drive.google.com/drive/folders/1eSP-3xXszwSAsnhzS7qDY1ISUakNhVcd?usp=sharing>
- The above shared link of google drive is the live working of our robot as to how it explores and covers the shortest path respectively.

CONCLUSION AND FUTURE WORK

- Efficiency for shortest path after exploration is 100% , while efficiency while exploration is not very effective yet and can be improved by a margin of around 15-20 % and accuracy while travelling the shortest path is again 100% and while exploration is around 65-70%.
- We achieved our final goal of making sure our robot travels the shortest path of the maze.
As this project also requires a lot of testing for getting appropriate values at various points , it also helped us in being more patient towards bugs as it took us around 10/12 hours on average to debug everytime.
- Future aspects -> to make our robot more efficient and accurate while exploring and also to use wheel encoder in it.

REFERENCES

- PDF shared by SRA (LABYRINTH 2019 FINAL.pdf)
- Research paper by Shadman sakib(BANGLADESH)
- VREP tutorial on you-tube by Leopoldo armesto
<https://www.youtube.com/playlist?list=PLjzuoBhdtaXOoqkJUqhYQletLLnJP8vjZ>

- Learn how to make a graph.
https://www.youtube.com/watch?v=5hPfm_uqXmw
- Learn BFS(Breadth-First search Algorithm) , Dijkstra Algorithm
[.https://www.youtube.com/watch?v=pcKY4hjDrxk](https://www.youtube.com/watch?v=pcKY4hjDrxk)
(BFS)
<https://www.youtube.com/watch?v=XB4MlexjvY0>
(Dijkstra)(Shortest path algorithm)