

READ ME

PART TWO

Given below are the instructions to execute a code written for the implementation of '2D Optimal Path Planning' for a mobile robot in a workspace space with obstacles, using A* Algorithm with non-holonomic constraints. Additionally, given below are the instructions to execute the A* path planner algorithm for a differential robot (TurtleBot Waffle) in ROS2.

- In the map below is the obstacle space for the robot to operate in.
- The 'Free Space' is represented by the white space.
- The 'Obstacle Space' is represented by the red.
- The 'Clearance Space' is represented by the black.
- The map dimensions are 600*200 cm with (0,0) being in the bottom left corner.



Obstacle Space in cm

Instructions to execute the code:

- The source code is a python file written using Visual Studio Code and I've attached the (a_star_Caleb_Dillon_Hamsaavarthan.py) format for execution.

- Run '.py' in the VS Code for execution.

-User Inputs

- Enter in the RPM of the left wheel and then the right wheel.
- Enter the desired clearance from the walls.
- Enter the (x, y), where x in range [0,600] and y in range [0,200], coordinates for the 'Start' node as requested by the code, with the respect to the origin (0,0) at the bottom-left corner of the map shown above. (recommended (50,100) as thats where the Gazebo simulation begins)

- Followed by the orientation of mobile robot at the source node

respectively, from the values (recommended 0 as that is the Gazebo starting orientation)

- If the given coordinates are not reachable (belongs to obstacle space), the code will prompt with “The given coordinates are not reachable. Try again with different Coordinates”.

- Provided with valid ‘Start’ and ‘Goal’ coordinates, the code will display the ‘Optimal Path’, as example shown below, for 5 seconds. (Press any ‘key’ to quit).

- The output of an animated video ‘A*_phase2.mp4’, will be created as a demonstration for node exploration and optimal path travelling for reference.

- The ‘Start’ and ‘Goal’ nodes are denoted by ‘Yellow’ and ‘Purple’ circles respectively.

- Lists of angular and linear velocity to be used in ROS Implementation.

Libraries used in the code:

- cv2

- heapq

- math

- numpy

- time

- matplotlib

Instructions to execute the code in ROS2:

1. Have all libraries downloaded and have ROS2 Humble installed.
2. Download the PROJECT3_WS to your computer from the github.
3. In your terminal enter the following commands:
 - a. `cd project3_ws`
 - b. `colcon build --packages-select turtlebot3_project3`
 - c. `source install/setup.bash`
 - d. `ros2 launch turtlebot3_project3 competition_world.launch`
4. After running the commands the Gazebo environment will open with the turtlebot in its initial position.
5. When you are ready to execute the path planer run these commands in another terminal.
 - a. `cd project3_ws`
 - b. `colcon build --packages-select turtlebot3_project3`
 - c. `source install/setup.bash`
 - d. `ros2 run turtlebot3_project3 path_plan.py`

6. After running the commands, the terminal will prompt you to enter the RPM1 and RPM2 values. We recommend entering RPM1=10 and RPM2 = 20.
7. Next, it will prompt the user to enter in the clearance value in cm. We recommend entering a value of 3 cm. After entering the map will begin to build.
8. After the map is built, the terminal will prompt the user to enter the x and y coordinates of the initial position and the angle of the bot. For the gazebo simulation, enter x = 50, y = 100, and orientation = 0.
9. Next enter your desired goal position on the other side of the map. Example: x = 570, y = 100.
10. After entering the values, the path planner will find the optimal path and execute it in the gazebo simulation environment. This is how to execute the code.

Libraries used in the code:

- cv2
- heapq
- math
- numpy
- time
- rclpy
- rclpy.node
- geometry_msgs.msg
- sys
- termios

Link to the GitHub Repository: https://github.com/calebmyersaz/Project3_Phase2_ENPM661

Videos:

Search Algorithm:

<https://drive.google.com/file/d/1mNJnJqtC2HCYlc3Og2lsrP6ltrsKOl8h/view?usp=drivesdk>

Team:

Name	Caleb Myers	Dillon Miller	Hamsaavarthan Ravichander
------	-------------	---------------	---------------------------

Directory ID	cmyers17	dmille19	rhamsaa
UID	120504440	121013316	120516979