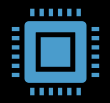
PATH PLANNING FOR MOBILE ROBOTS

Adarsh Kumar

22cd3003

INTRODUCTION

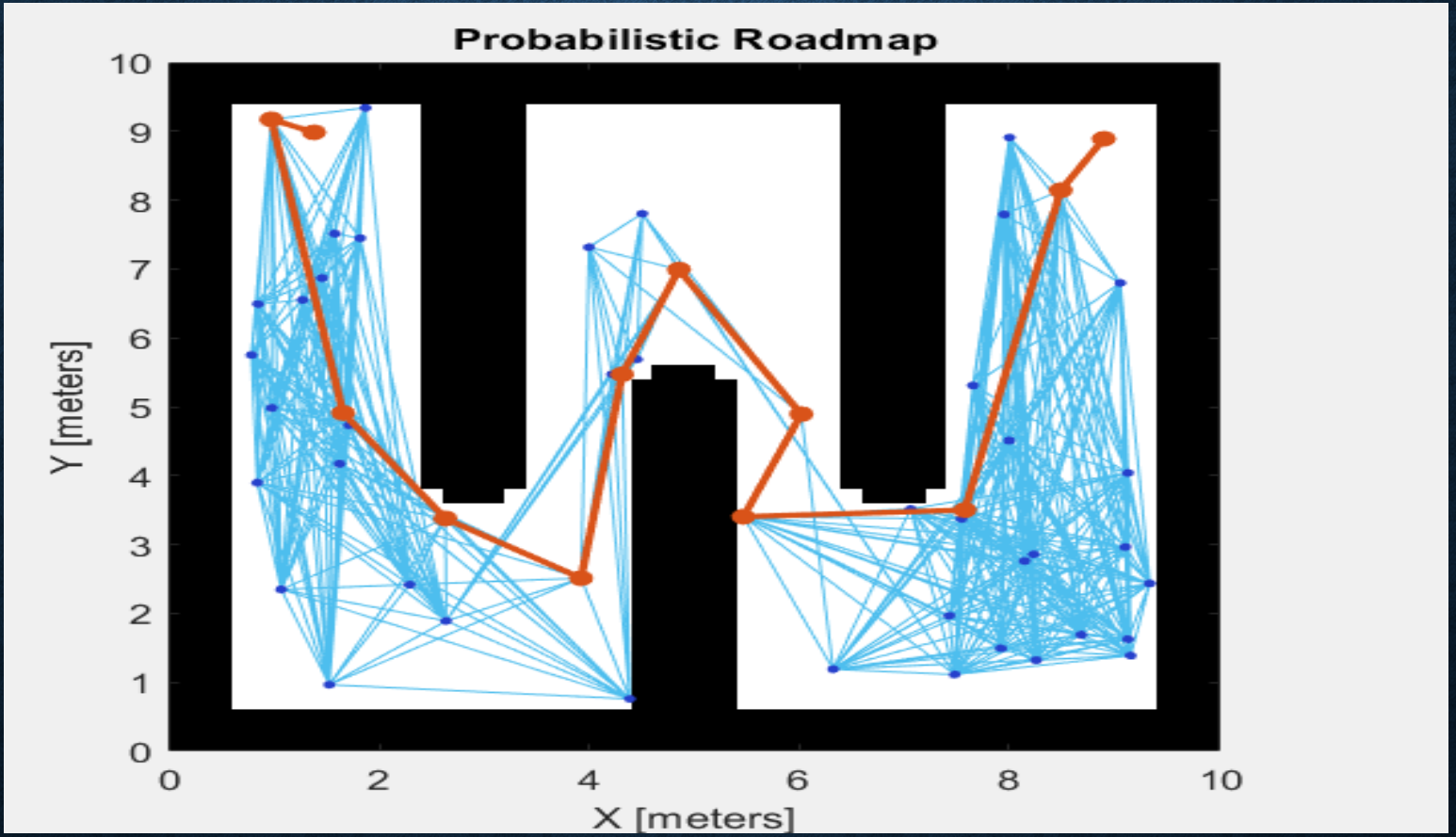
My project on "Path Planning for Mobile Robots," where we will make a setup for a simulation environment to enable efficient navigation for mobile robots. It is made by Binary Occupancy Maps and Probabilistic Roadmaps (PRM).



Our primary goal is to establish a robust framework for guiding mobile robots through complex environments from which we can plan a precise and efficient path for mobile robots.

We've employed a combination of Binary Occupancy Maps and PRM to create a dynamic and adaptable simulation environment.

OVERVIEW


Probabilistic Roadmap
Y [meters]
X [meters]
0
1
2
3
4
5
6
7
8
9
10
0
2
4
6
8
10

CREATION OF MAP

For the creation of map we created a binary occupancy map named myMap with a size of 10x10 and a resolution of 5 cells per meter.

Now we initialized a 50x50 matrix called walls to represent obstacles. Set the top, bottom, left, and right edges of the walls matrix to 1, representing the walls of the environment.

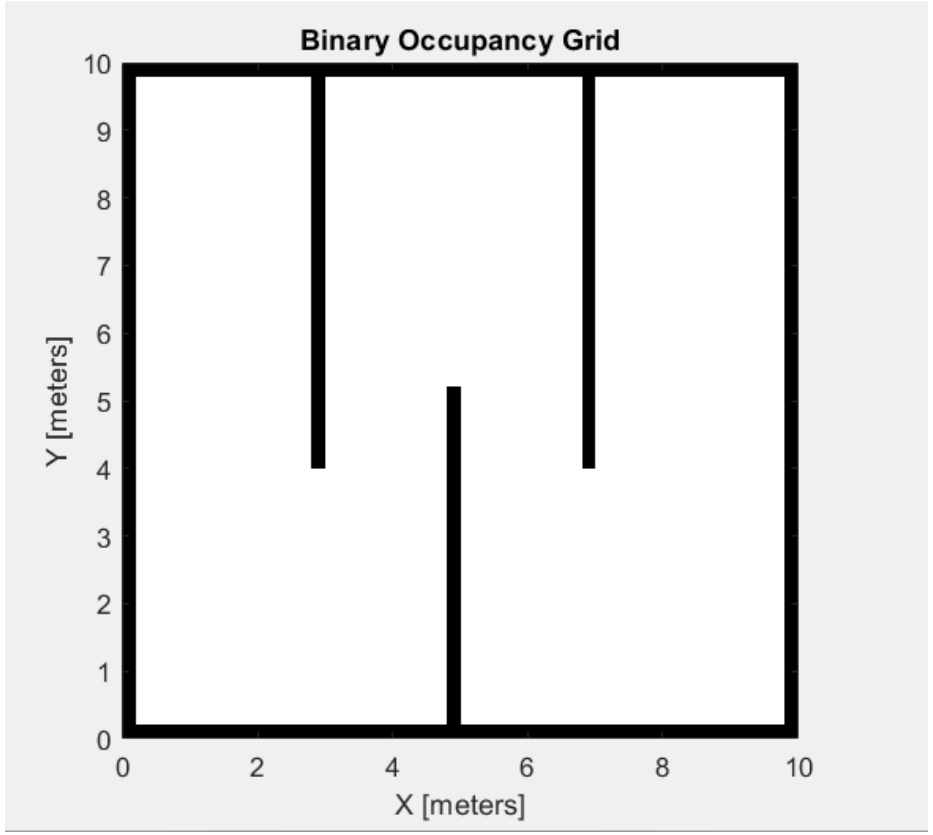
We Created three vertical divisions in the environment by setting specific columns in the walls matrix to 1.

Now we Set the occupancy of the cells specified by the (1,1) coordinate in the myMap using the walls matrix.

So, by these techniques we created a binary occupancy map .

Matlab code and the map is in the next slide.

```
myMap = binaryOccupancyMap(10,10,5);
walls = zeros(50,50);
walls(1,:) = 1; % Top wall
walls(end,:) = 1; % Bottom wall
walls(:,1) = 1; % Left wall
walls(:,end) = 1; % Right wall
walls(1:30,15) = 1; % Left division
walls(25:end,25) = 1; % Middle division
walls(1:30,35) = 1; % Right division
setOccupancy(myMap,[1 1],walls,"grid")
show(myMap)
```

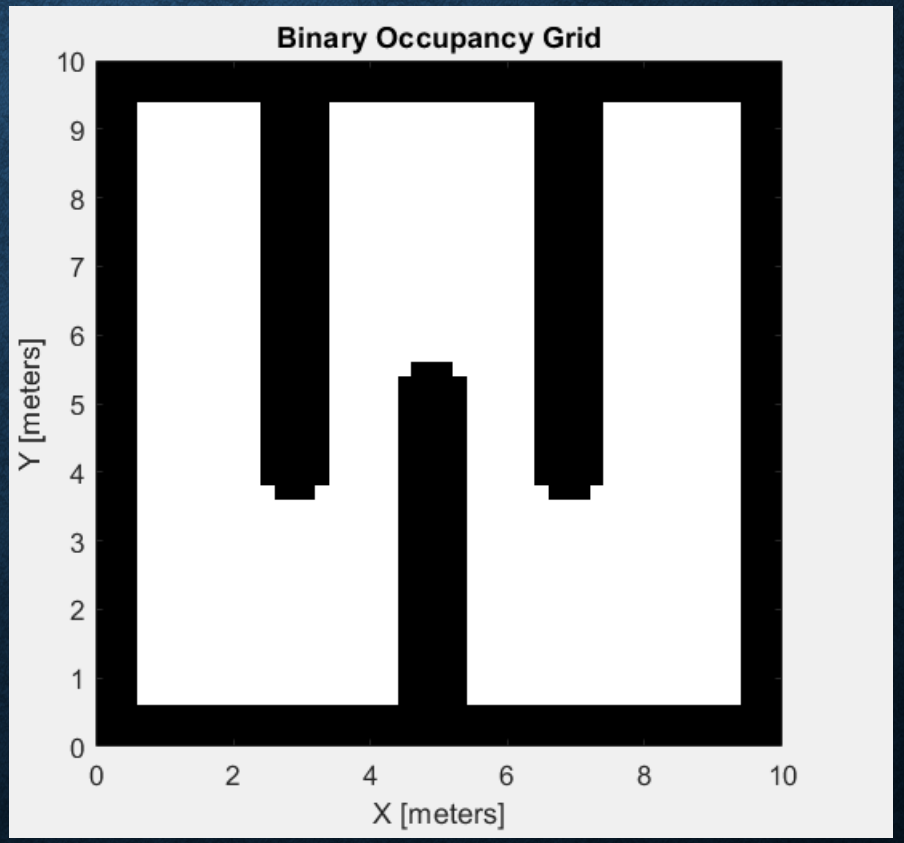


Now we will Inflate the obstacles in the map by a radius of 0.3 meters. This is done to avoid the damages in the drones due to errors.

MATLAB Code-

```
inflateRadius = 0.3;
inflate(myMap,inflateRadius)
show(myMap)
```

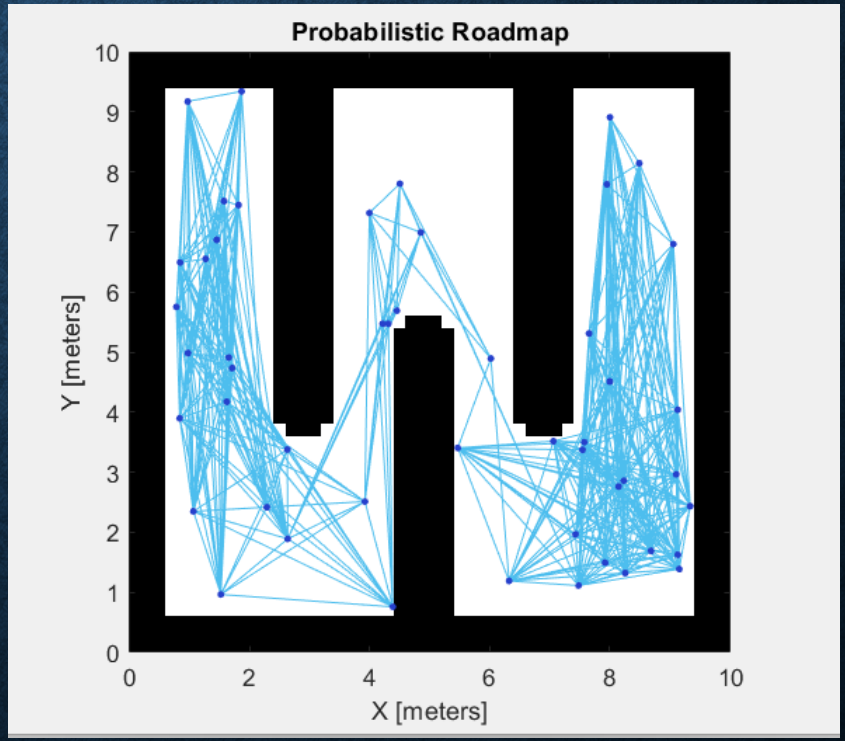
After inflating we got a map like this-



PROBABILISTIC ROADMAP(PRM)

Set the number of nodes in the PRM, the maximum connection distance between nodes, and the occupancy map for the PRM.

```
prm=mobileRobotPRM;
prm.NumNodes = 50;
prm.ConnectionDistance = 80;
prm.Map = myMap;
show(prm)
```

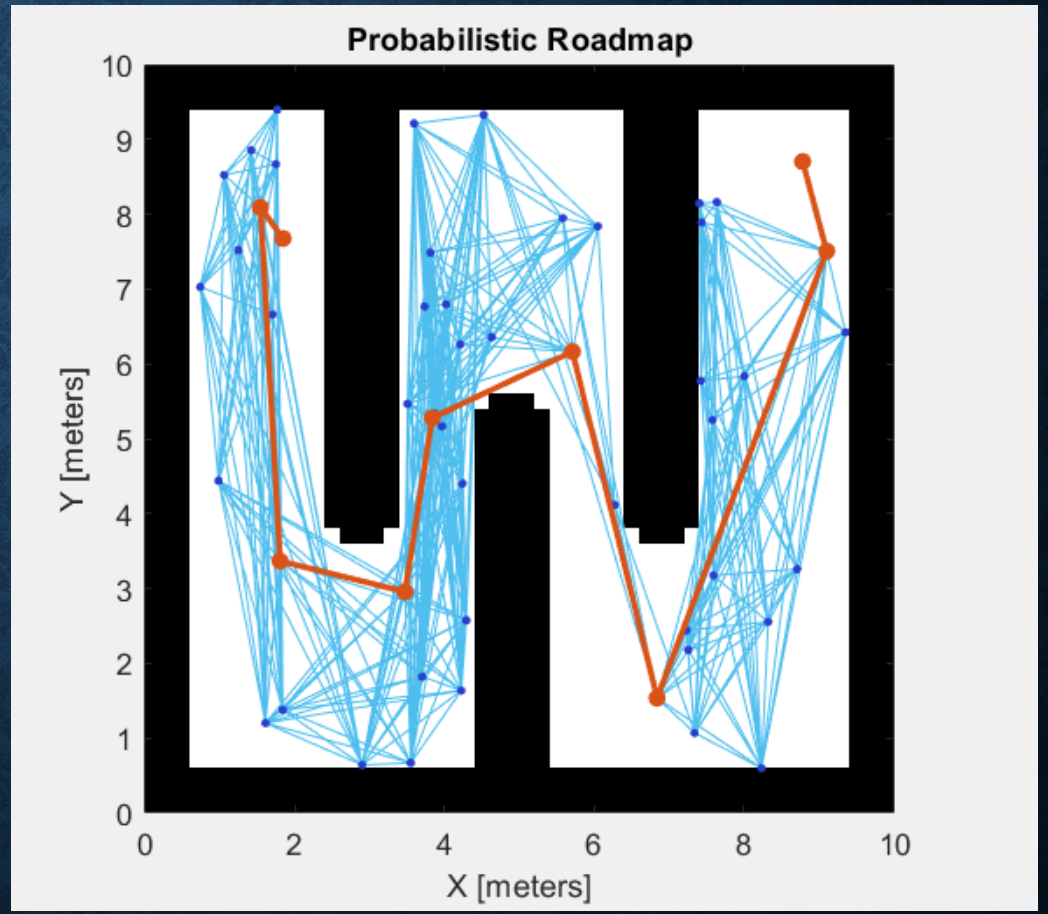


- prompt the user to click on the figure to specify the start location for the robot.
- prompt the user to click on the figure to specify the start location for the robot.

```
disp('Cick for start location');
startLocation = ginput(1);
disp('Click for goal postion')
endLocation = ginput(1);
path = findpath(prm,startLocation,endLocation);
```

- Now attempt to find a path between the specified start and end locations using the PRM and then display the final PRM with the found path.

```
path = findpath(prm,startLocation,endLocation);
show(prm)
```

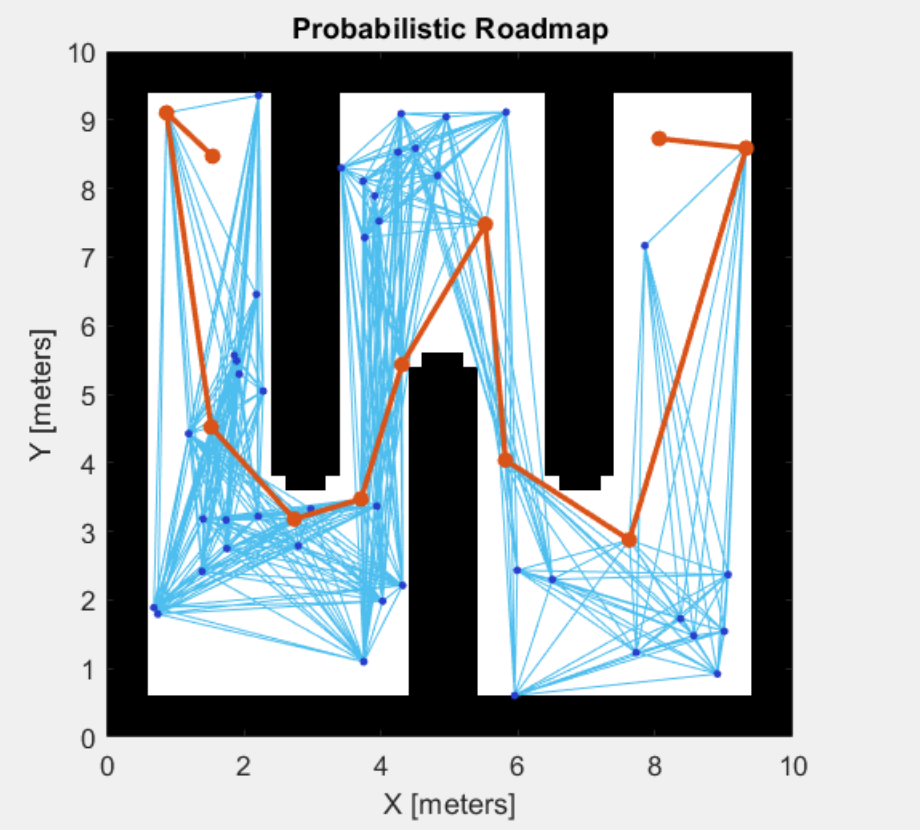


```
while isempty(path)
    prm.NumNodes = prm.NumNodes + 20;
    update(prm);
    path=findpath(prm,startLocation,endLocation);
    show(prm)
    pause(1);
end

show(prm)
```

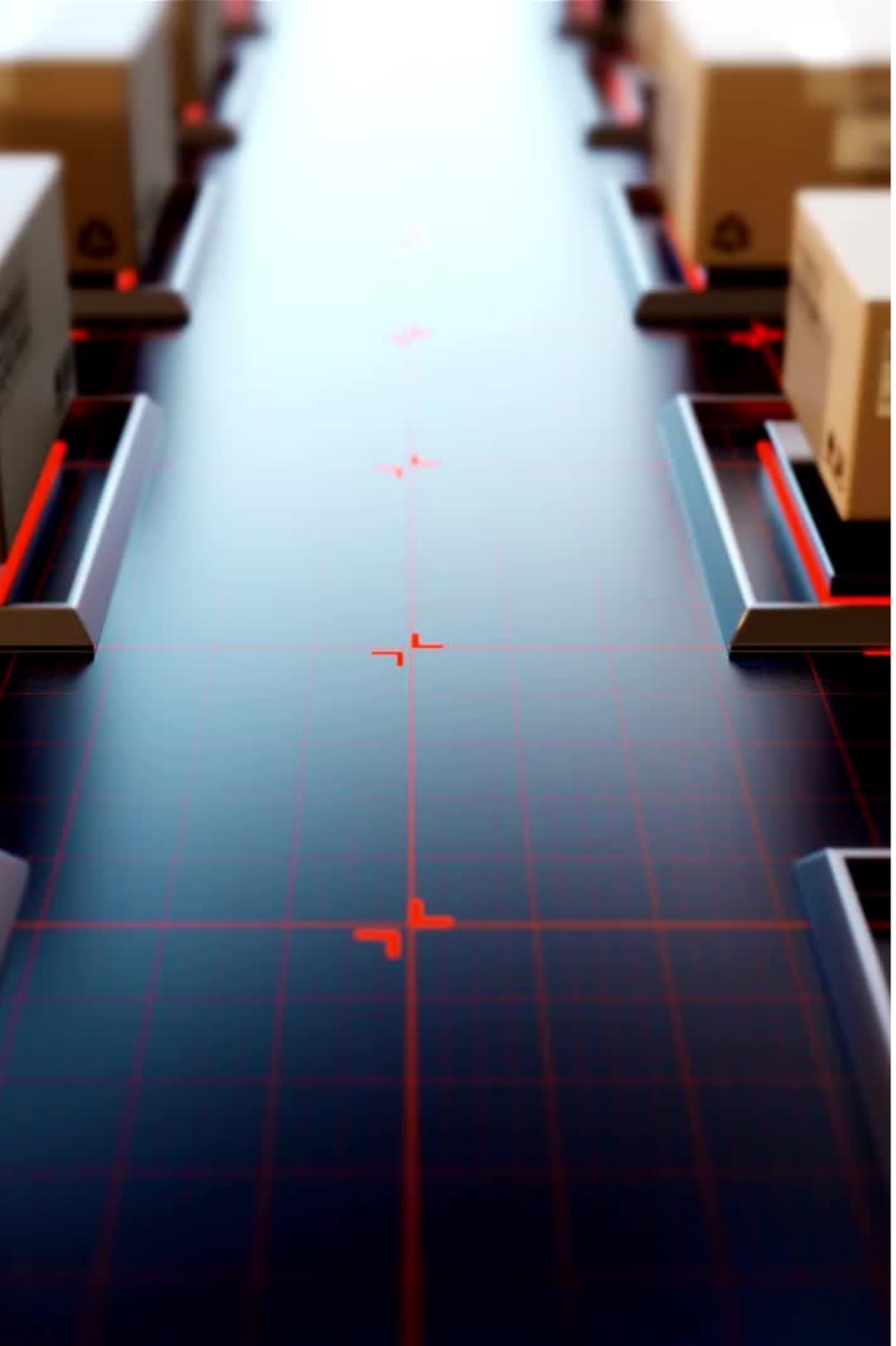
- If the path is initially empty, the script increases the number of nodes in the PRM, updates the PRM, tries to find a path again, and repeats until a path is found.

RESULTS



Workspace

Name	Value
endLocation	[8.0607,8.7266]
inflateRadius	0.3000
myMap	*1x1 binaryOccup...*
path	*11x2 double*
prm	*1x1 mobileRobot...*
startLocation	[1.5421,8.4696]
walls	*50x50 double*



CONCLUSION

- Efficient path planning is a cornerstone in the field of mobile robotics, contributing to advancements in various fields, including autonomous vehicles, logistics, and surveillance. Our project addresses the critical need for reliable and dynamic navigation systems.

- As we delve into the details of our simulation environment and the implementation of these methodologies, we can explore the intricacies of path planning for mobile robots.

THANK YOU