מבוא לרובוטיקה

עבודת בית 5

236972

עדן סרחאן

324849256

תאלא סאבא

324030220

Part 1:

Q1:

Bug 0 pseudo code:

```
# Main loop to move the robot to the goal
 while not reached_goal:
# Check if the goal has been reached
 if no_obstacle_between(current_position, goal):
 move_towards_goal(current_position, goal)
else:
# Obstacle detected, find the closest point on the obstacle boundary
 closest_obstacle_point = find_closest_obstacle(current_position, goal)
 # Move along the obstacle boundary until past the obstacle
move_along_obstacle_boundary(current_position, closest_obstacle_point)
sleep(0.5)
```

no_obstacle_between(current_position, goal)

- **What it does:** Checks if there is a clear path from the robot's current position to the goal.
- **Returns:** True if no obstacles exist; False otherwise.

move_towards_goal(current_position, goal)

- **What it does:** Moves the robot directly toward the goal along a straight line(m-line)
- , as long as no obstacle is in the way.
- **Updates:** The robot's current_position.

find_closest_obstacle(current_position, goal)

- **What it does:** Identifies the point on the obstacle boundary that is nearest to the goal.
- **Returns:** The closest point on the obstacle boundary.

move_along_obstacle_boundary(current_position, closest_obstacle_point)

- **What it does:** Guides the robot to follow the edge of the obstacle(on the left) until it reaches the point closest to the goal.
- **Updates:** The robot's current_position as it moves along the boundary.

Q2:

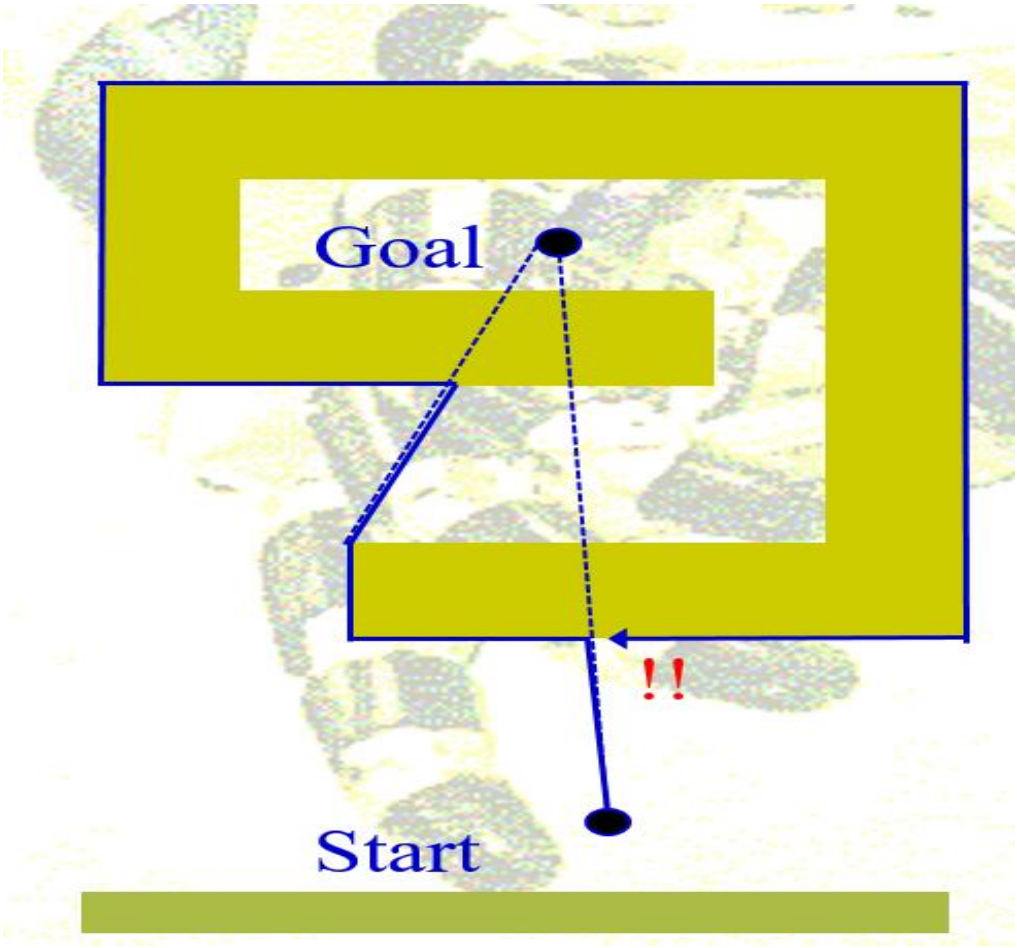
Main advantages:

- Efficient in simple environment: In an environment where there are simple obstacles , bug0 finds the solution relatively easy
- Minimal memory usage : bug 0 does not require much memory , when in fact it needs the current position and object detection
- Simple: bug0 is relatively simple to implement and understand

Main disadvantages:

- No path optimization : bug0 does not optimize the path it finds
- Not immune to local minimum traps: bug0 can get as we explained in part 3 where the drone gets around the obstacle and at the end it goes back to its beginning point , dictating that there isn't a solution
- Inefficient in complex environment : bug0 takes a long time to complete especially in complex environments

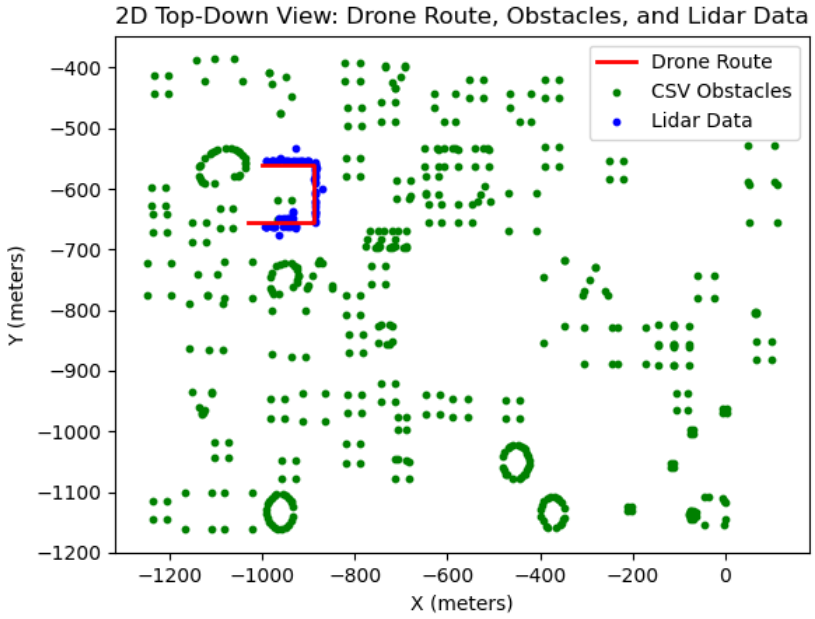
Q3:



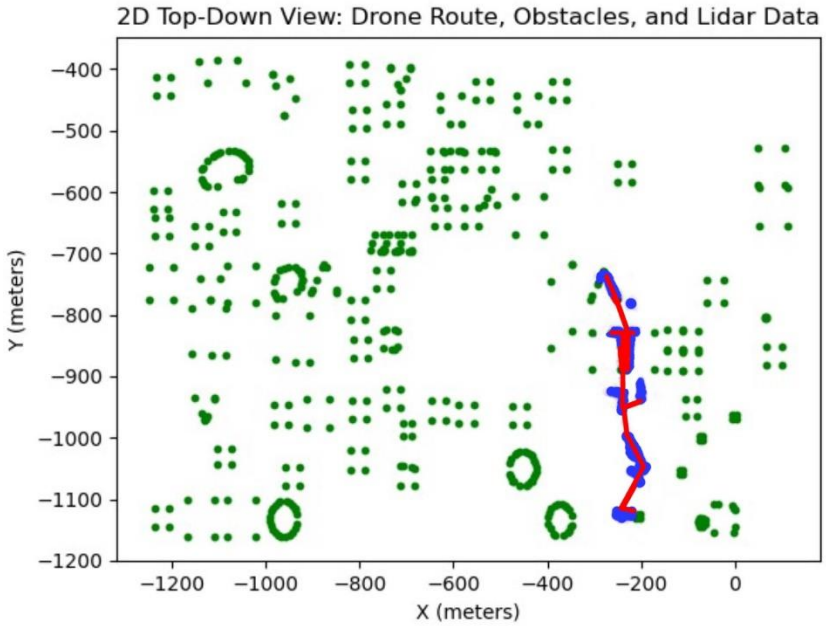
bug0 doesn’t search the environment thoroughly enough to detect when it gets into a loop , for example in the picture here from the lectures we can see when the drone bumps into the object the second time , it goes left along all around the object until it comes back to the same point , dictating that there is no solution when in fact there is

Part 2:

Q1


2D Top-Down View: Drone Route, Obstacles, and Lidar Data
Drone Route
CSV Obstacles
Lidar Data
Y (meters)
X (meters)
−400
−500
−600
−700
−800
−900
−1000
−1100
−1200
−1200
−1000
−800
−600
−400
−200
0

Q2:


2D Top-Down View: Drone Route, Obstacles, and Lidar Data
Y (meters)
X (meters)
−400
−500
−600
−700
−800
−900
−1000
−1100
−1200
−1200
−1000
−800
−600
−400
−200
0