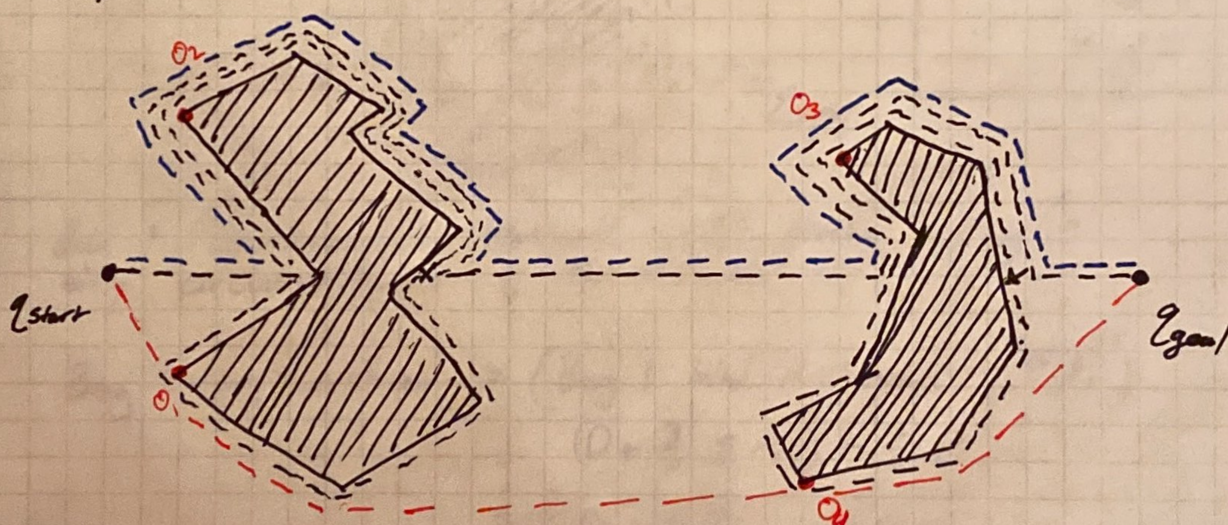
ASEN 5519 - ALGORITHMIC MOTION PLANNING

HOMEWORK #1

JOE MICELI

EXERCISE #1

Draw trajectories for Bug 1, Bug 2, tangent bug (∞ radius) for point robots in the space below



KEY:

--- Bug 1 path

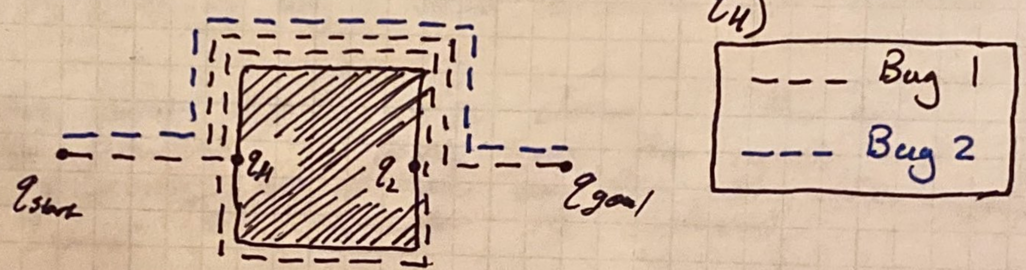
--- Bug 2 path

--- Tangent bug path

EXERCISE #2

Construct an example for which the upper bound of the traveled path for bug 1 is reached. How does Bug 2 perform?

Bug 1 upper bound = $D + \frac{3}{2} \Sigma P_i$ (q_L is opposite side of obstacle from q_H)



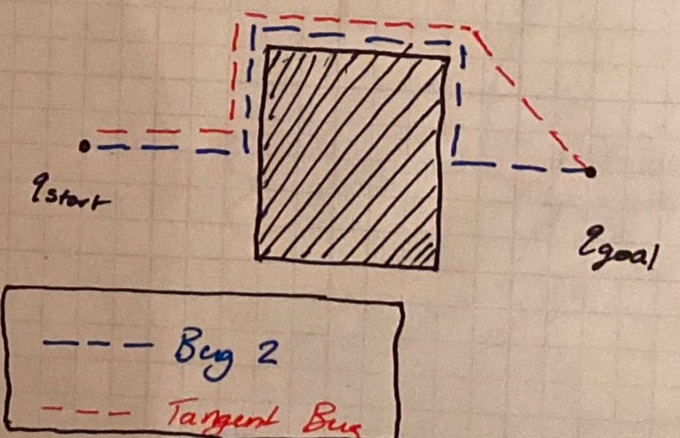
Bug 1 circumnavigates obstacle 1.5 times, Bug 2 only circumnavigates $\frac{1}{2}$ the obstacle

$$\text{Bug 2 total distance} = (\text{Bug 1 total distance} - \Sigma P_i)$$
$$= (D + \tfrac{3}{2} \Sigma P_i) - \Sigma P_i$$
$$= D + \tfrac{1}{2} \Sigma P_i$$

Exercise #3

Difference between Bug 2 and tangent bug w/ $\emptyset$ range detector?

The main difference is that bug 2's ~~[illegible]~~ ~~[illegible]~~ behaviors are constrained by the m-line (line from start to goal). Even w/ 0 range detector, tangent bug can leave obstacles earlier than bug 2.

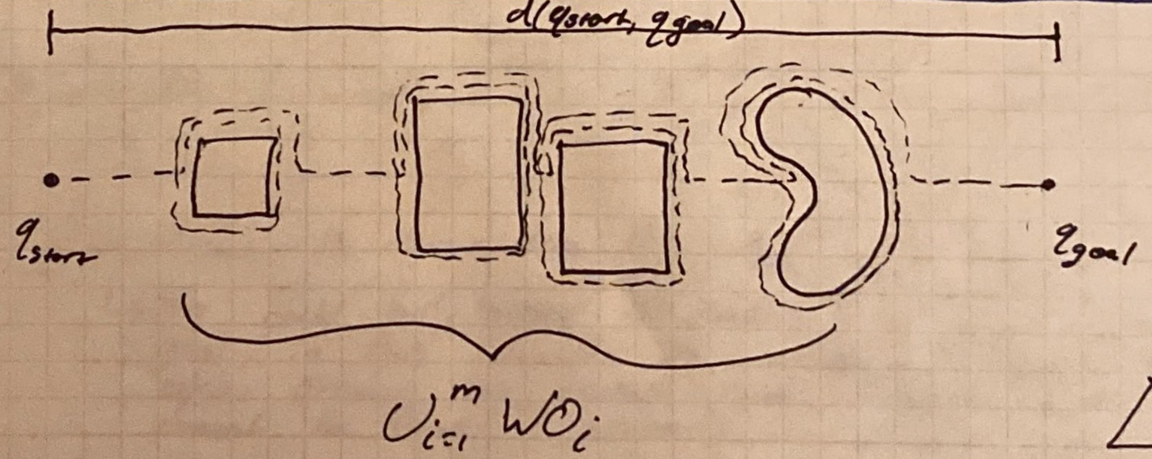


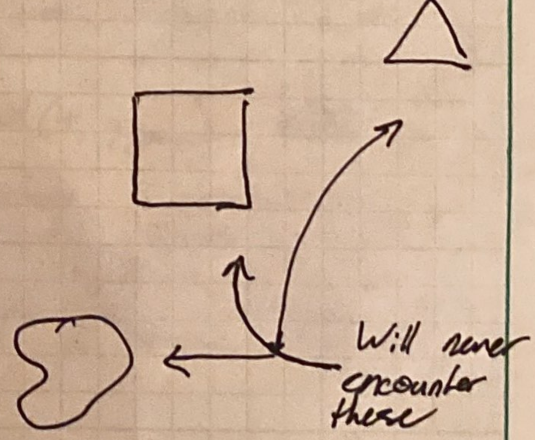
Exercise 4

Robot w/ goal of reaching q_{goal} in workspace W w/ obstacles $WO = \bigcup_{i=1}^{n} WO_i$ where WO_i for all $i \in \{1, 2, \ldots, m\}$ $(m < n)$ is within radius $d(q_{start}, q_{goal})$ from q_{start} and rest of obstacles are outside this radius. What is max # of obstacles Bug 1 could encounter

Answer:
Max # of obstacles Bug 1 could encounter: m obstacles

Reason:
Each leave point of Bug 1 is closer to the goal then the previous hit point. Thus, the robot will never encounter an obstacle outside of $d(q_{start}, q_{goal})$ from q_{start}. To encounter m obstacles, all m obstacles must lie on the m-line between q_{start} and q_{goal}.





EXERCISE #5

Prove if tangent bug is complete or not.

Complete ≡ Algorithm finds a path to goal (if it exists) or terminates in failure if it does not in finite time

Tangent Bug Is complete.

» Assume it never terminates:

- Each leave point is closer than it's corresponding hit point
  - ↳ Unless obstacle has ∞ perimeter
- Each hit point must be closer than the previous leave point.
- So there must be a finite # of hit-leave pairs
- Robot will exhaust them, hit the goal, terminate

» Terminates Incorrectly

- 2 cases:

(1) d_{reach} is never less than $d_{followed}$

↳ This could only happen if the obstacle encased the goal, in this situation, robot would completes a cycle around the obstacle and identify a path cannot be found (no error)

(2) $d(x, n) + d(n, q_{goal})$ and $d(x, q_{goal})$

BOTH stop changing, which would cause the robot to be unable to move toward goal or begin boundary following. This is not possible though on a finite workspace w/ finite obstacles and a known direction to goal

EXERCISE #6

Write an algorithm for a robot with a 270° FOV rangefinder.

① While True do

② Rotate 90° clockwise then 90° CCW

③ Repeat

④ Continuously move toward point $n \in \{T, O_i\}$ which minimizes $d(x, n) + d(n, q_{goal})$

⑤ Until

- The goal is encountered

OR

- The direction that minimizes $d(x, n) + d(n, q_{goal})$ begins to increase $d(x, q_{goal})$

⑥ Rotate 90° CW then 90° CCW

⑦ Choose boundary ~~[illegible]~~ following direction wich contion which continues the most recent motion-to-goal direction

⑧ Repeat

⑨ Continuously update d_{reach} $d_{followed}$ and $\{O_i\}$

⑩ Continuously move toward $n \in \{O_i\}$ in the chosen boundary direction

⑪ Until

- Goal is reached
- Robot completes a cycle around obstacle in which case goal cannot be achieved
- $d_{reach} < d_{followed}$

⑫ End While