Navigation: Bug algorithms

Introduction

Overview

Definition

Navigation is the process of moving a robot through its environment to a **goal state** while avoiding obstacles.

We'll see several types of algorithms for this problem.

Key Question 1

How do each of these algorithms work?

Key Question 2

What are the differences in when each of these algorithms can be applied?

Introduction

Definition

Bug algorithms are a general class of navigation algorithms that do not require the robot to know the obstacles ahead of time.

Defining characteristic: Only **local** information about the obstacles is available.

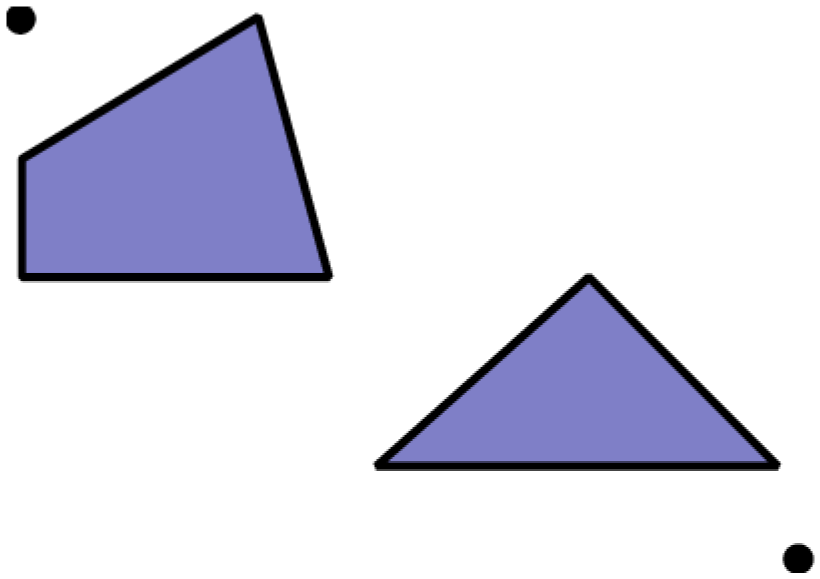
- No long-range sensing.
- No detailed representations of the obstacle shapes.

Motivating problem: Finding the Eiffel Tower

Sensor model

Bug algorithms assume that the robot has some combination of sensors that allow it to perform two basic actions:

1. Sense the direction and distance to the goal.
2. Follow nearby obstacle boundaries and measure the distance traveled.



An optimistic but incorrect algorithm

Here's a first crack at solving the problem:

- Move toward the goal until reaching an obstacle.
- Follow obstacle boundary until it's possible to move directly toward the goal.
- Repeat.

When does this fail?

What about an algorithm that can **guarantee** reaching the goal?