CS440/ECE448 Lecture 38: Configuration Space

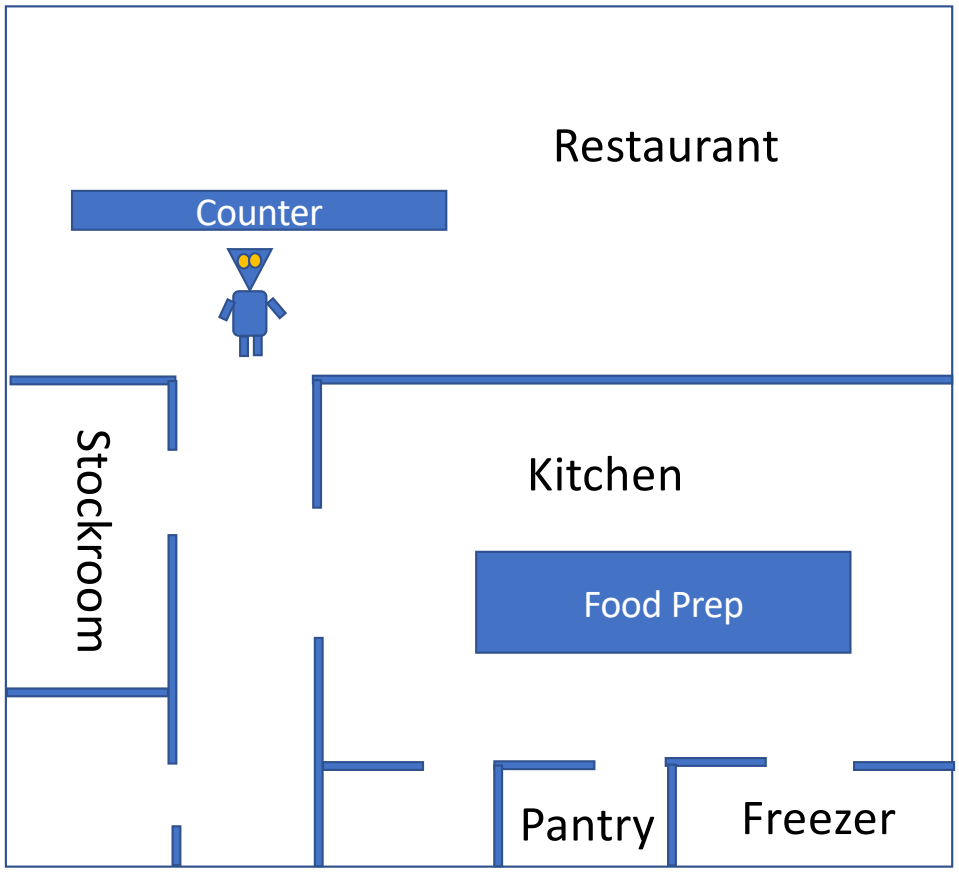
Mark Hasegawa-Johnson, 4/2022

Outline

- Planning = Search
- Poor Robot!
- Configuration Space
- The Robot Arm Problem
- Geometry of the Robot Arm Problem
- Searching for a Solution in Configuration Space

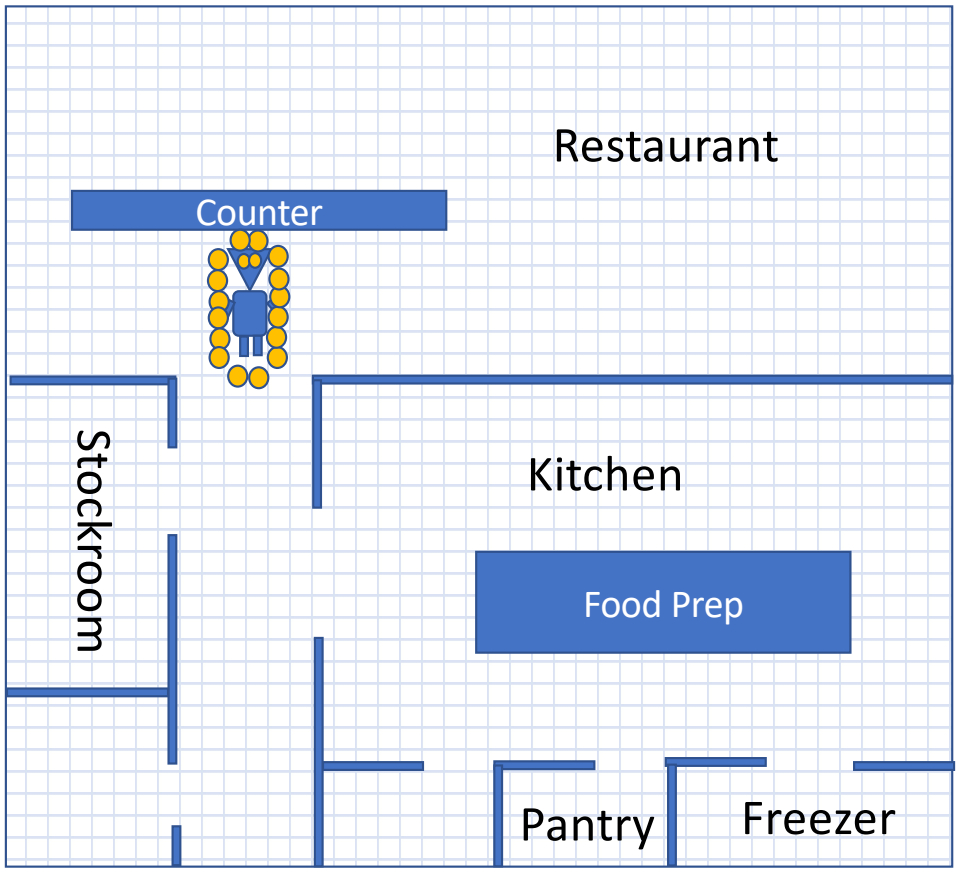
Planning = search

- The problem: robot needs to get to the stockroom



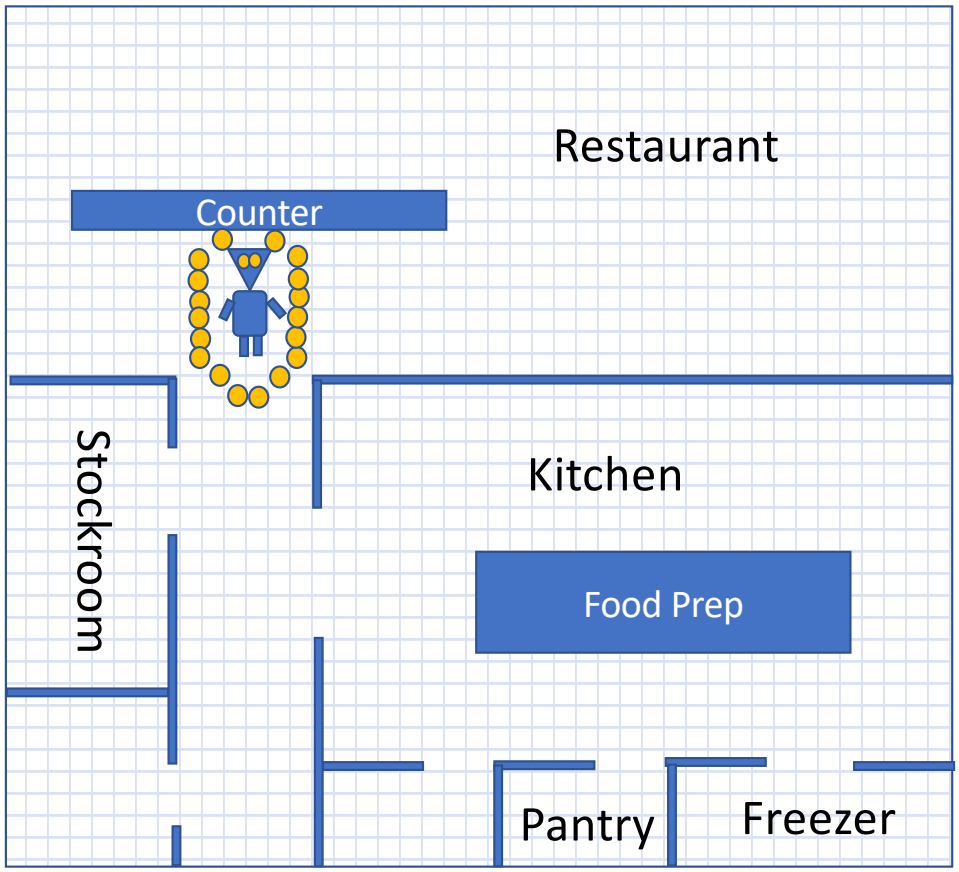
Planning = search

- Step #1: Robot consults his internal map and uses BFS or A* to find the best path.



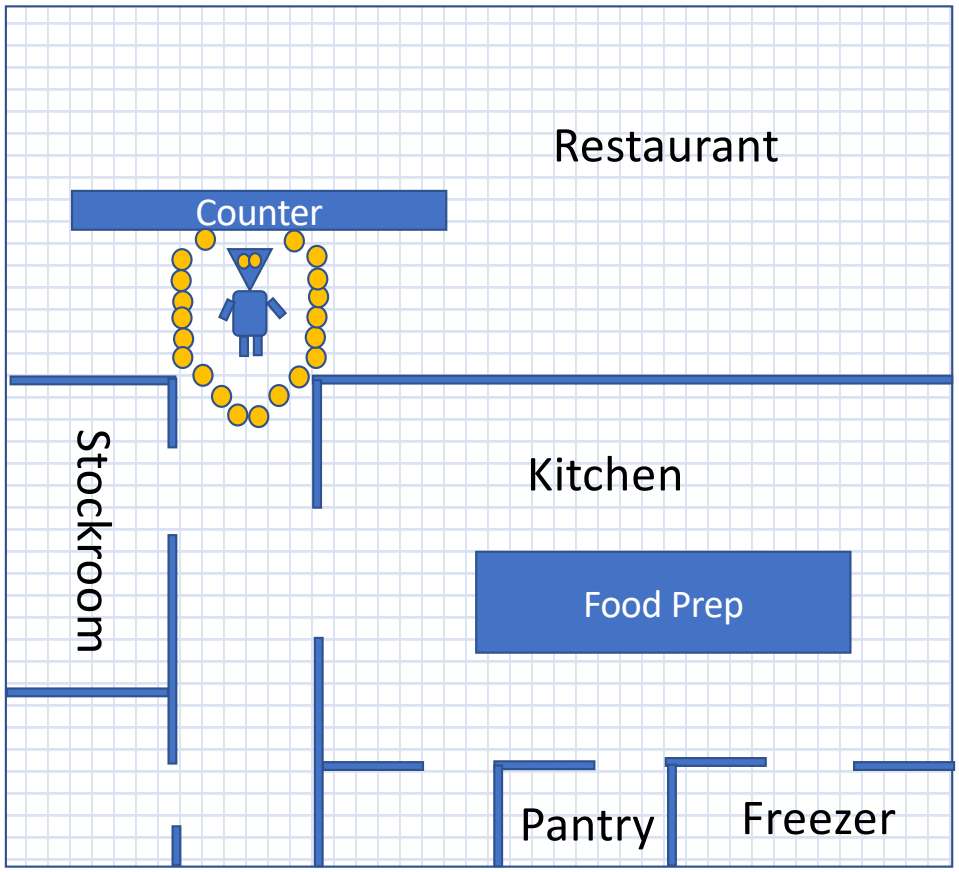
Planning = search

- Step #1: Robot consults his internal map and uses BFS or A* to find the best path.



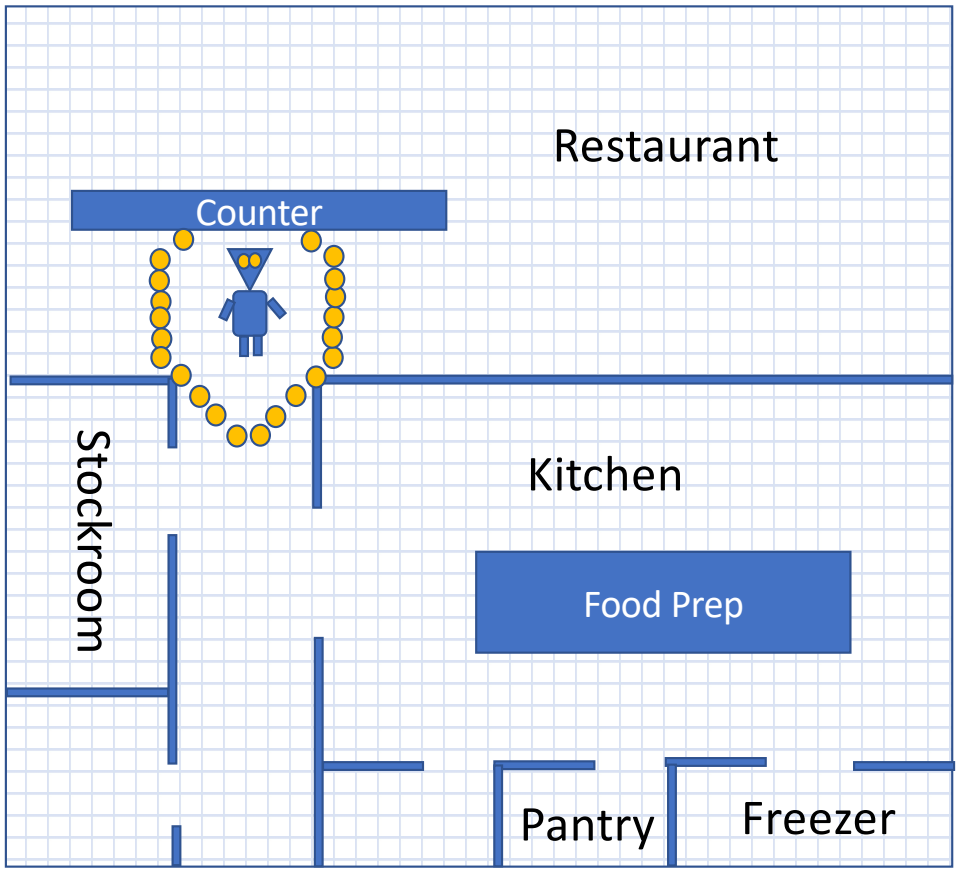
Planning = search

- Step #1: Robot consults his internal map and uses BFS or A* to find the best path.



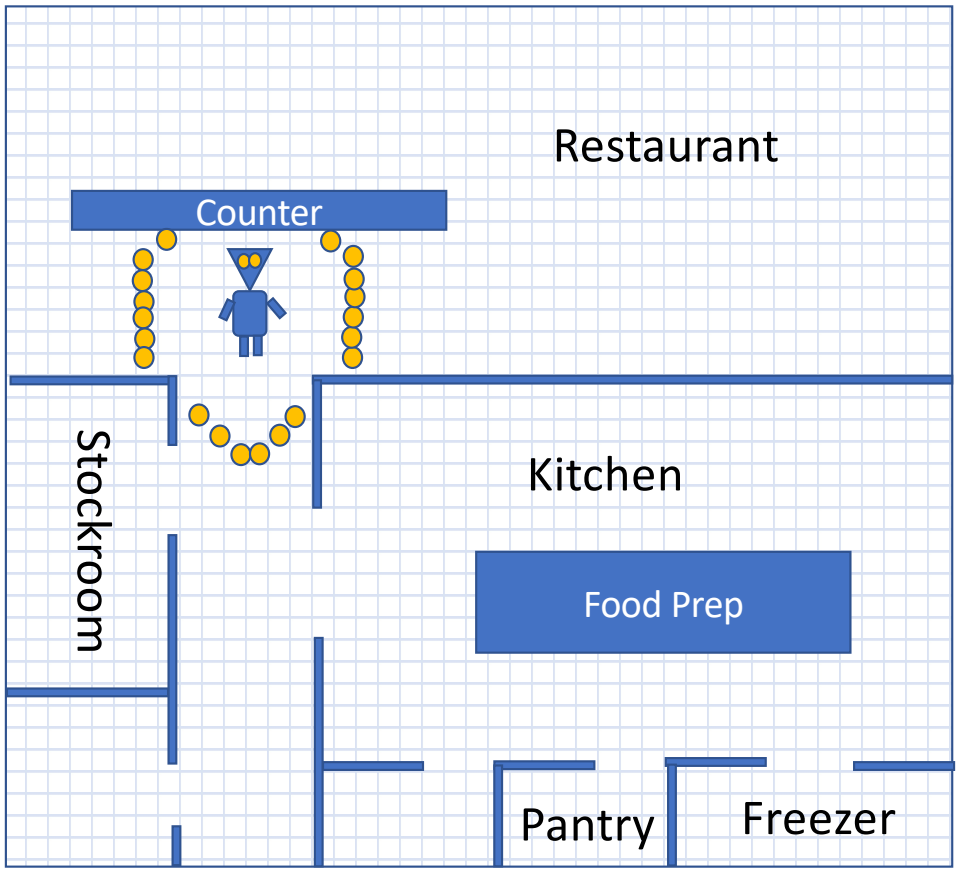
Planning = search

- Step #1: Robot consults his internal map and uses BFS or A* to find the best path.



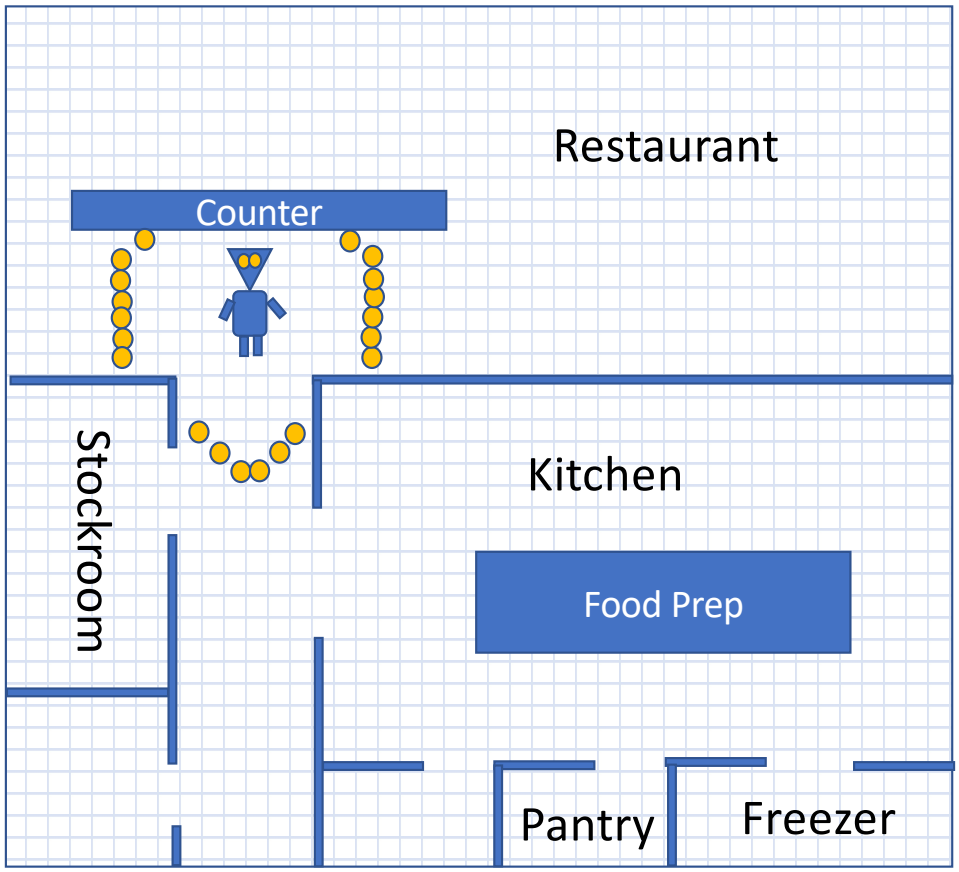
Planning = search

- Step #1: Robot consults his internal map and uses BFS or A* to find the best path.



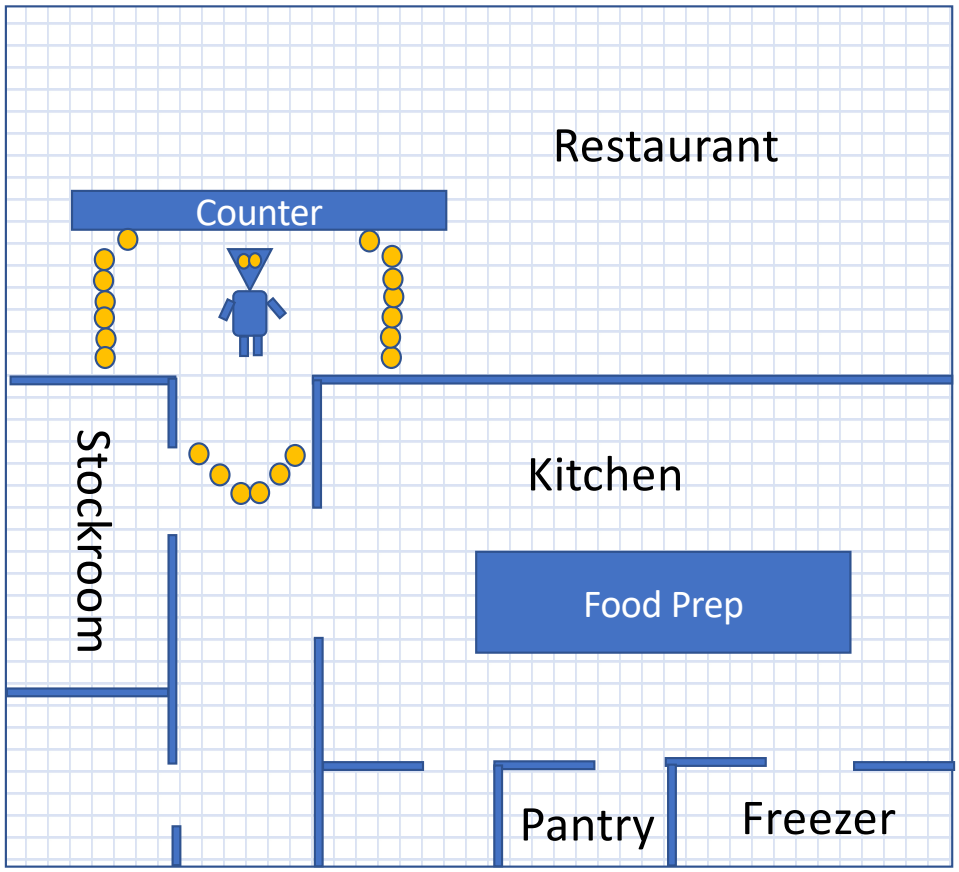
Planning = search

- Step #1: Robot consults his internal map and uses BFS or A* to find the best path.



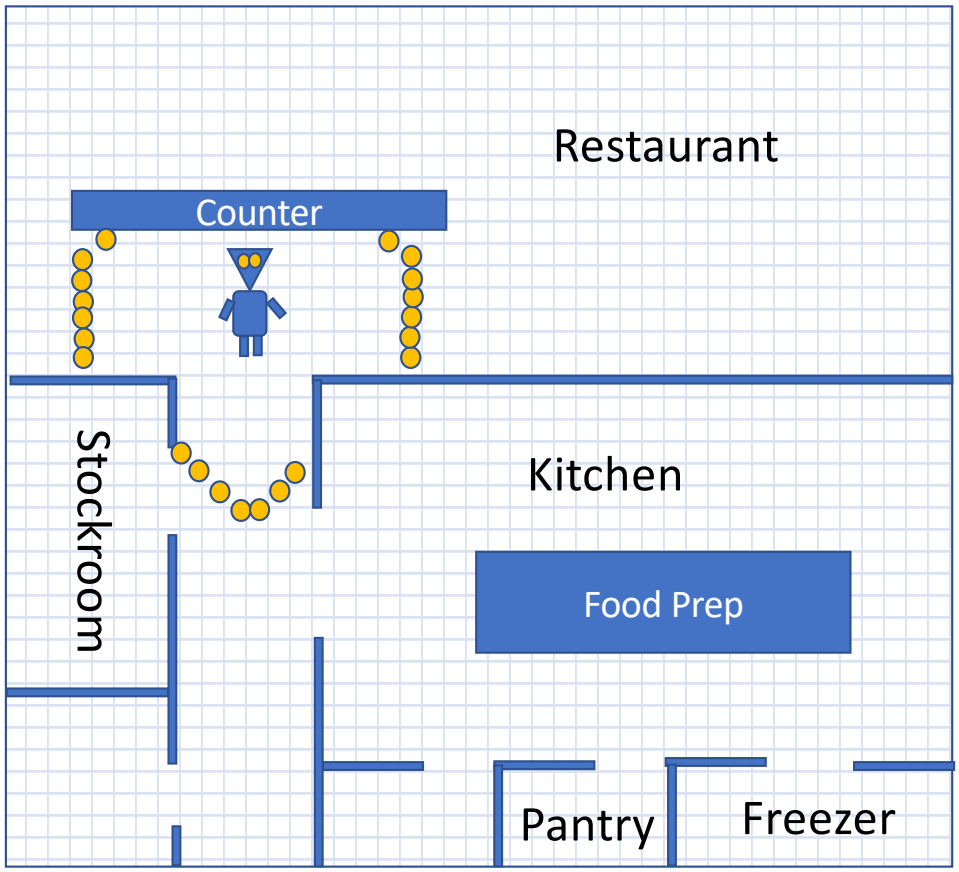
Planning = search

- Step #1: Robot consults his internal map and uses BFS or A* to find the best path.



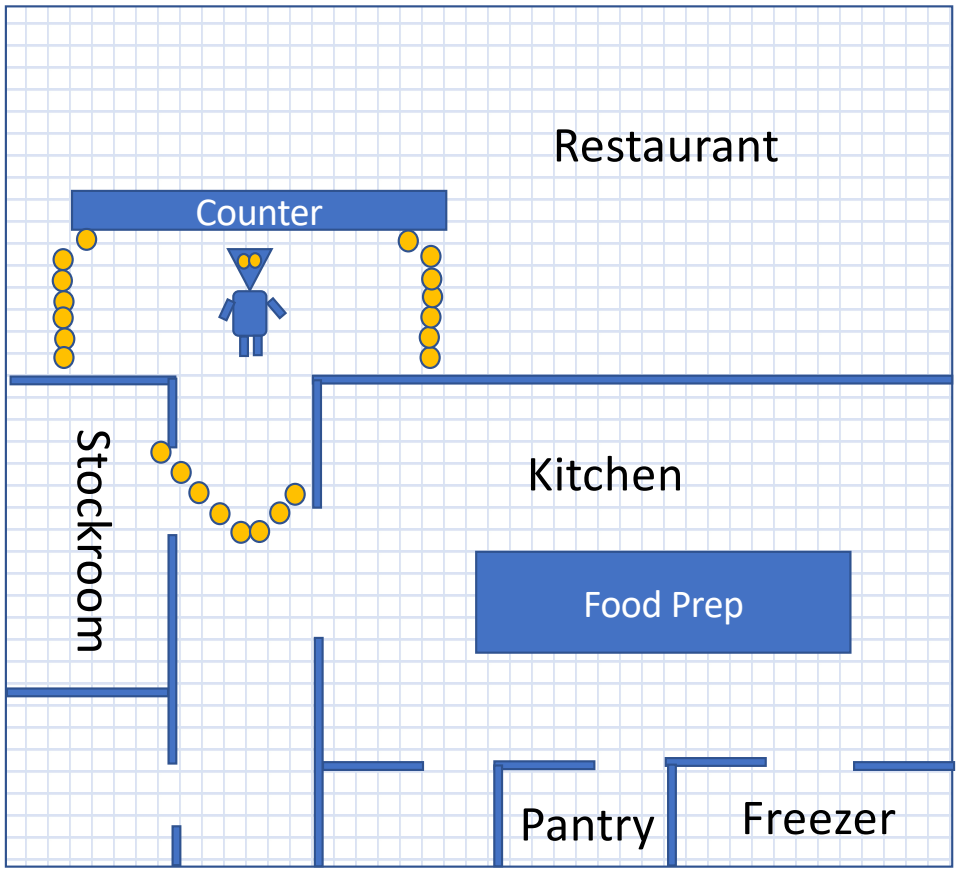
Planning = search

- Step #1: Robot consults his internal map and uses BFS or A* to find the best path.



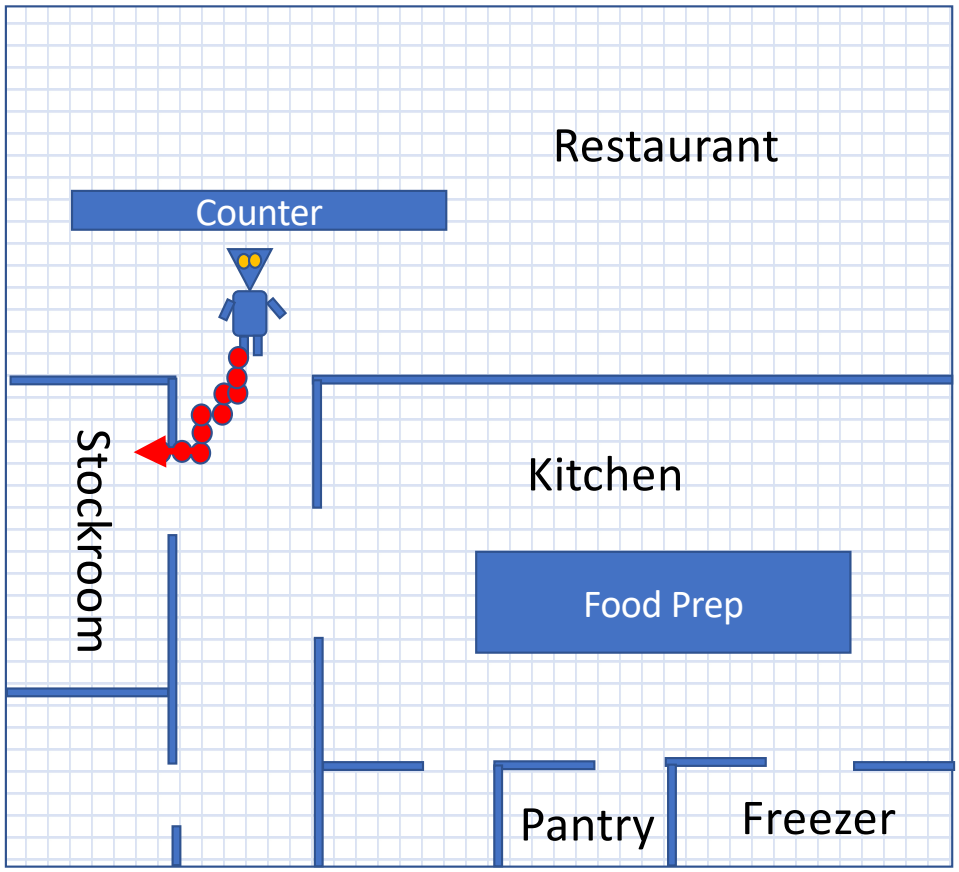
Planning = search

- Step #1: Robot consults his internal map and uses BFS or A* to find the best path.



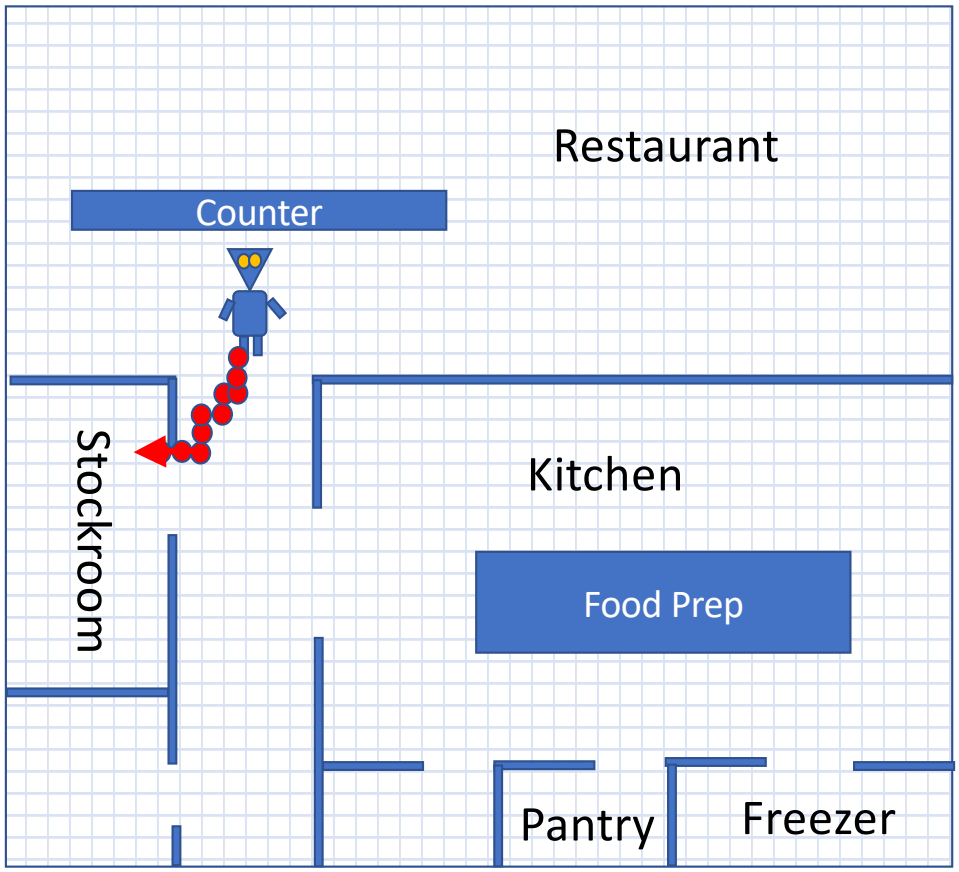
Planning = search

- Step #1: Robot consults his internal map and uses BFS or A* to find the best path.



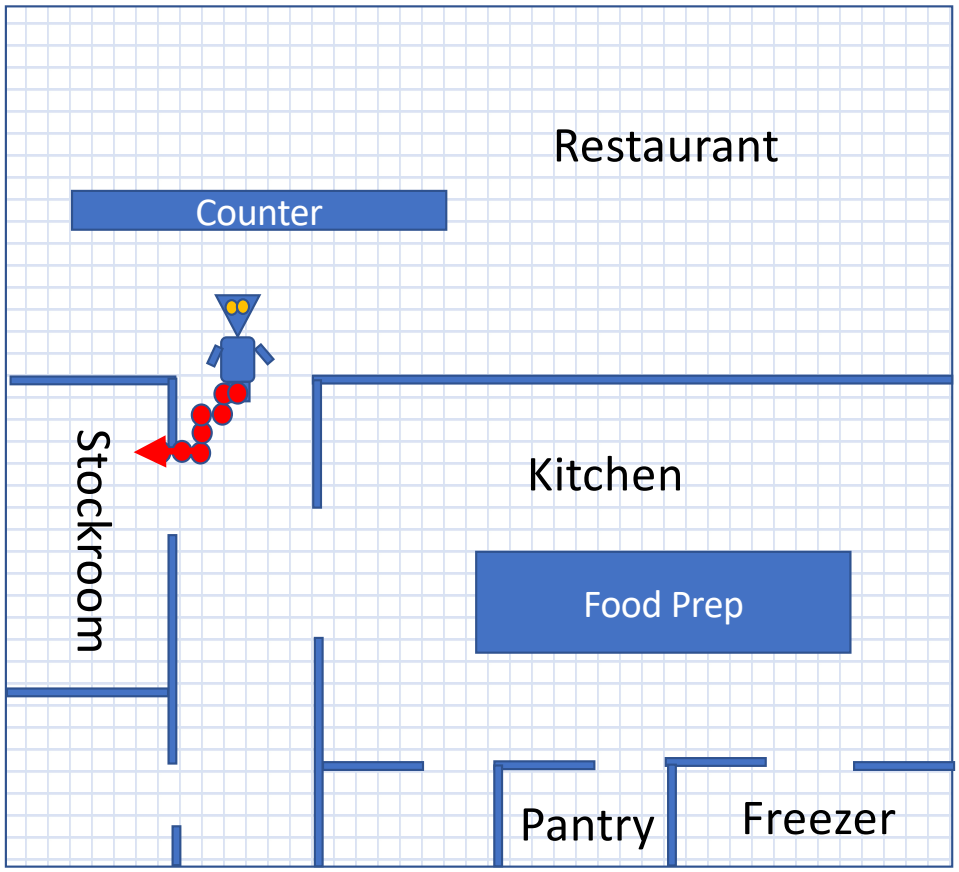
Planning = search

- Step #2: Robot attempts to follow that path.



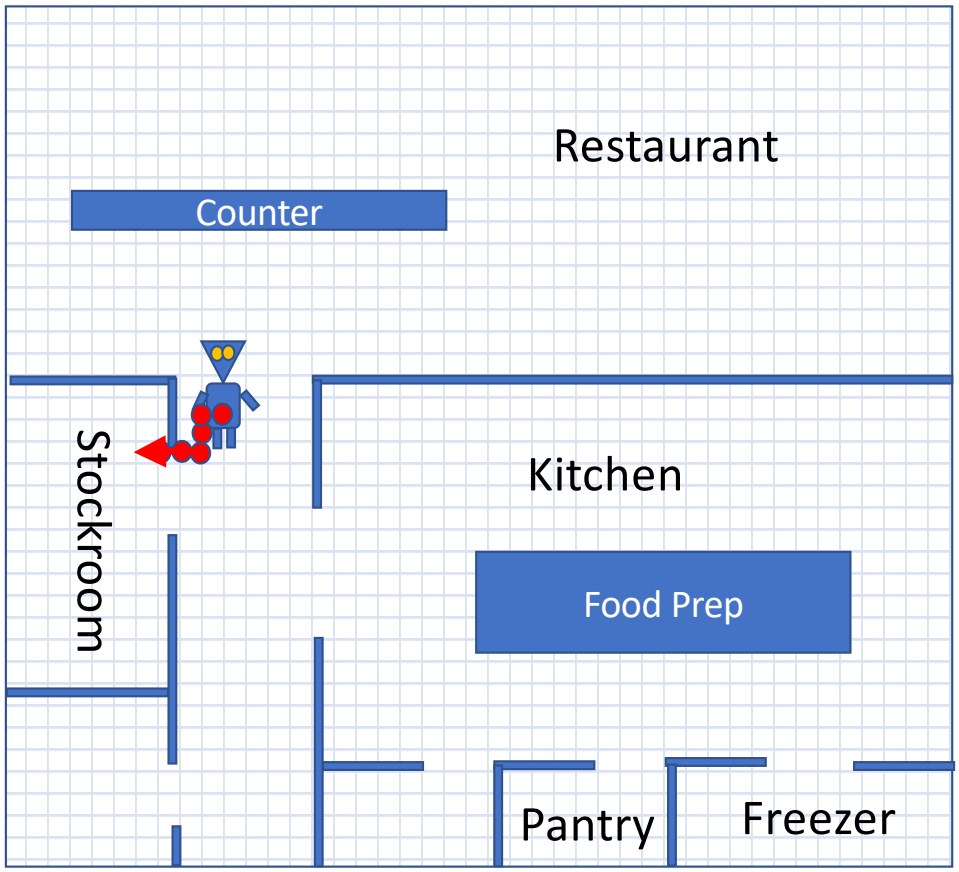
Planning = search

- Step #2: Robot attempts to follow that path.



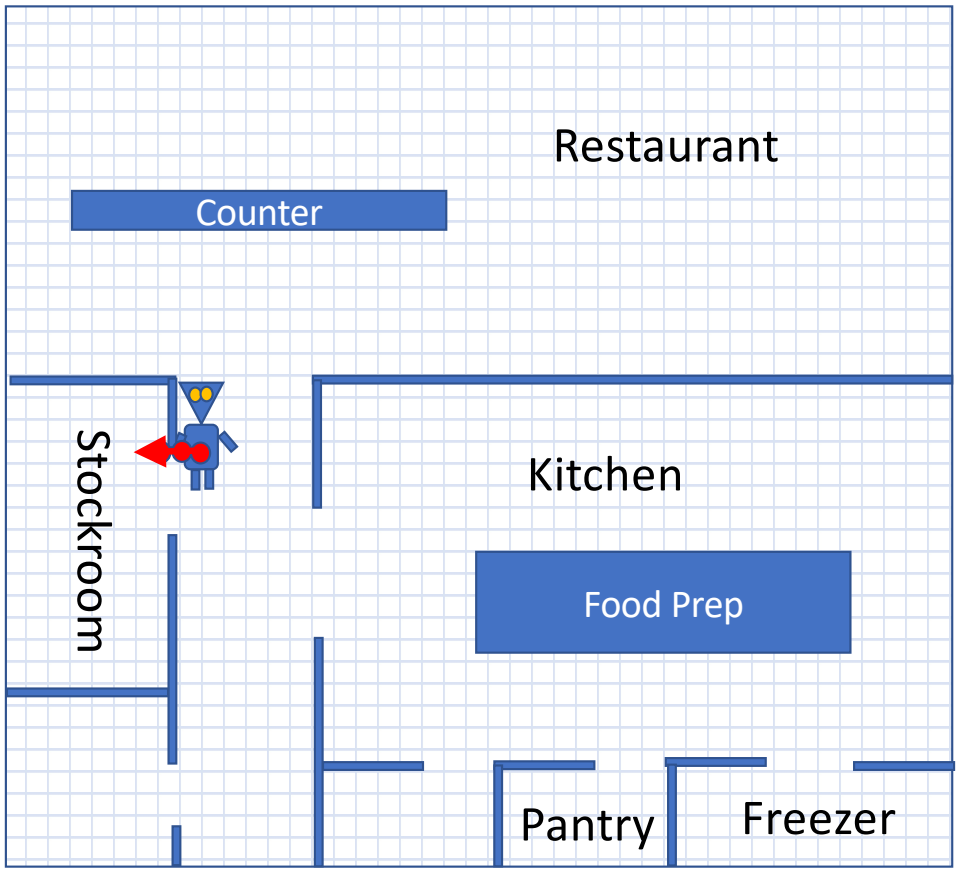
Planning = search

- Step #2: Robot attempts to follow that path.



Planning = search

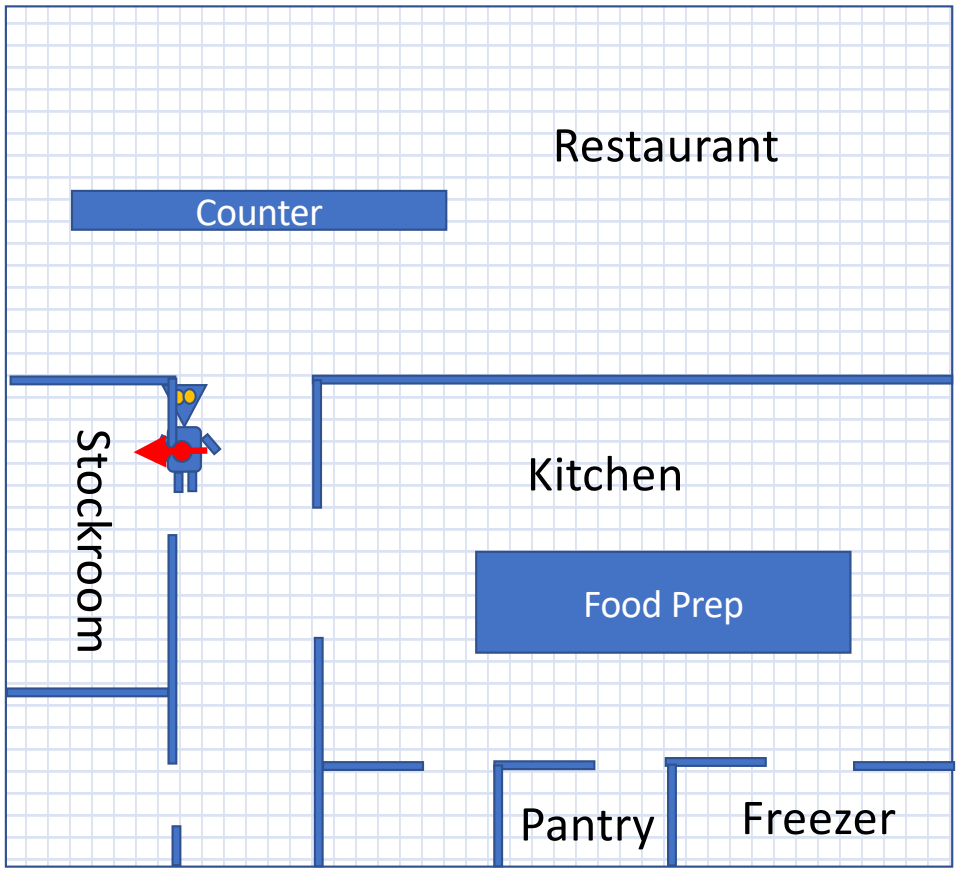
- Step #2: Robot attempts to follow that path.



Planning = search

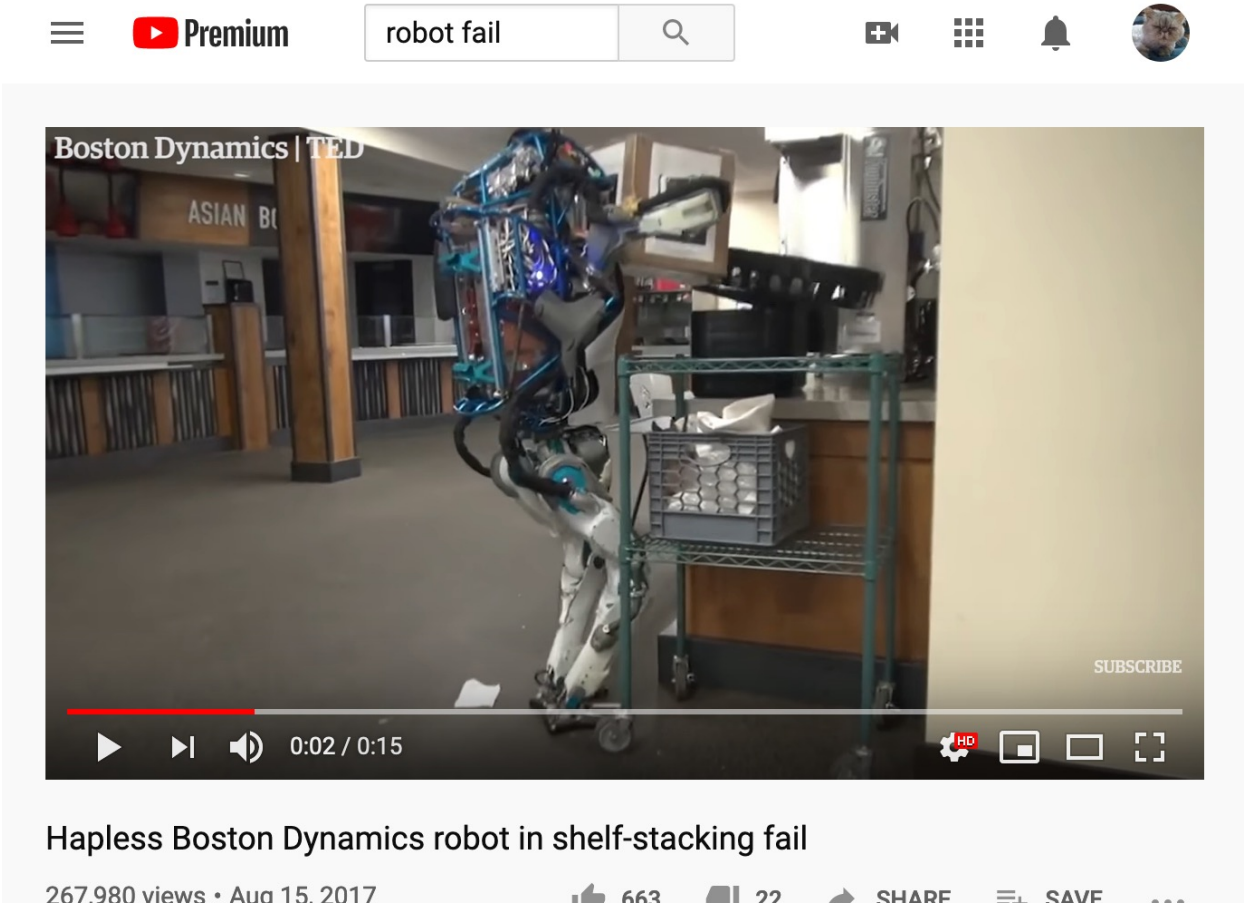
- Step #3: Robot bumps into the doorframe.

Poor robot.



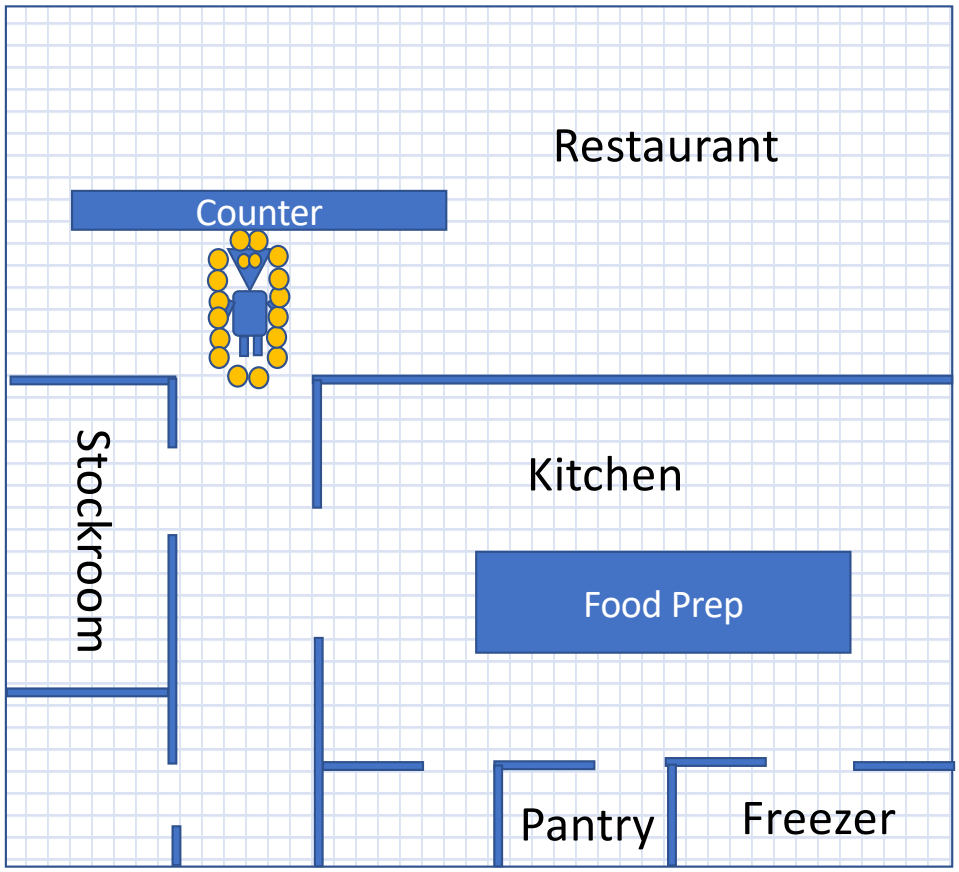
Poor robot

https://www.youtube.com/watch?v=JzlsvFN_5HI



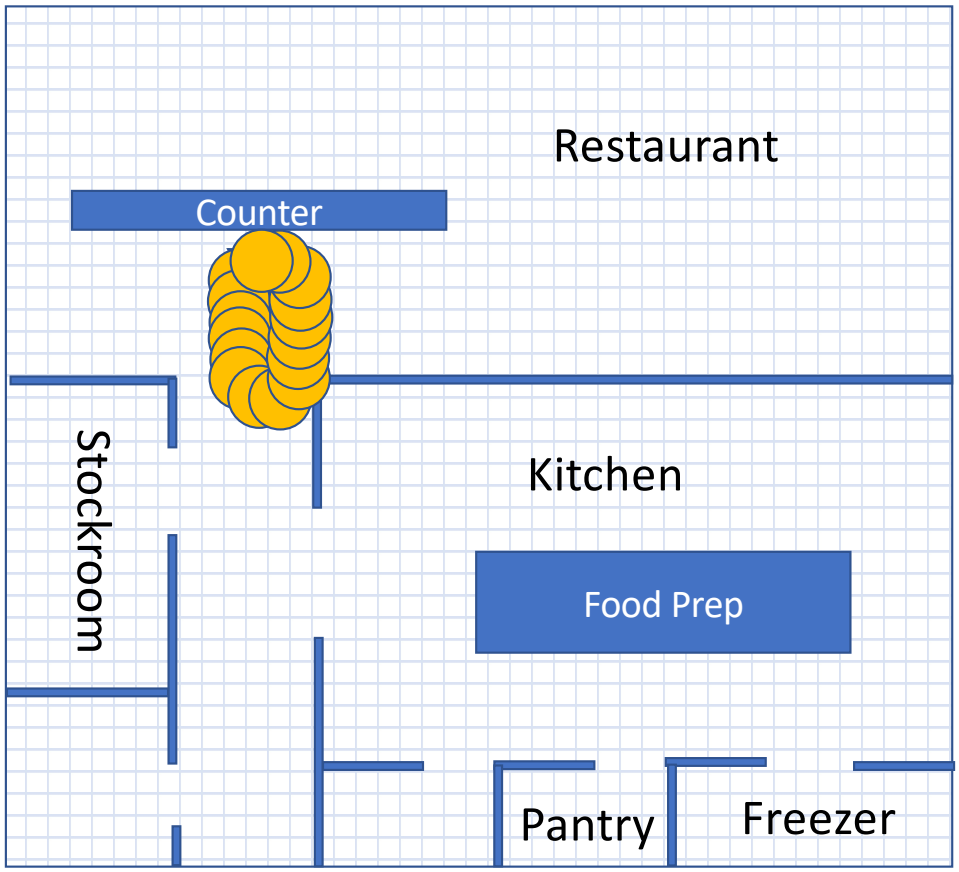
How can we help the robot?

- Let’s give it more information.
- Let’s tell it how wide it is.



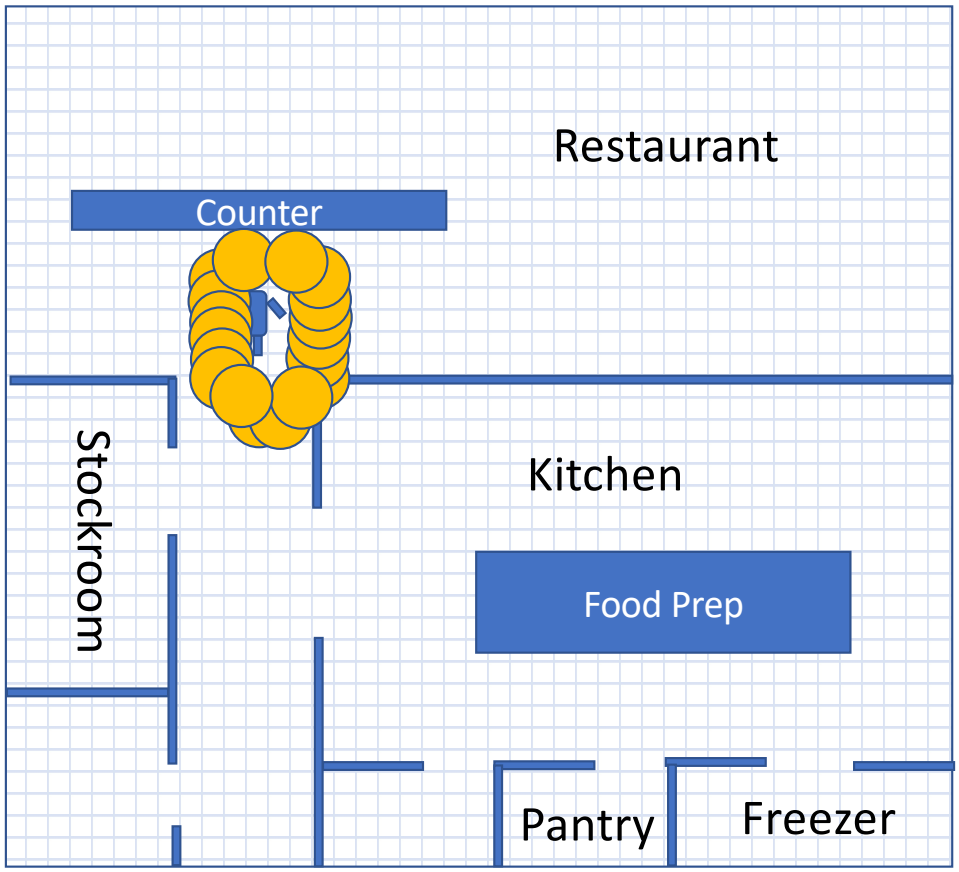
How can we help the robot?

- Option #1: every node in the search tree carries information about the size of the robot.



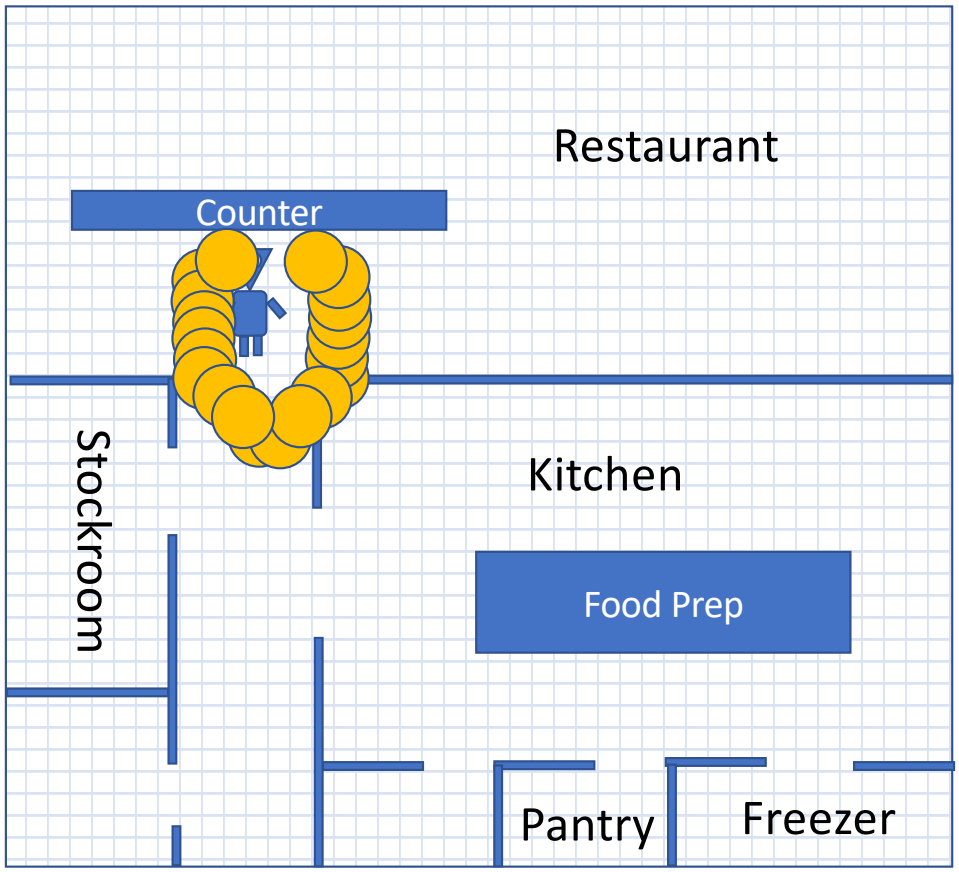
How can we help the robot?

- Option #1: every node in the search tree carries information about the size of the robot.



How can we help the robot?

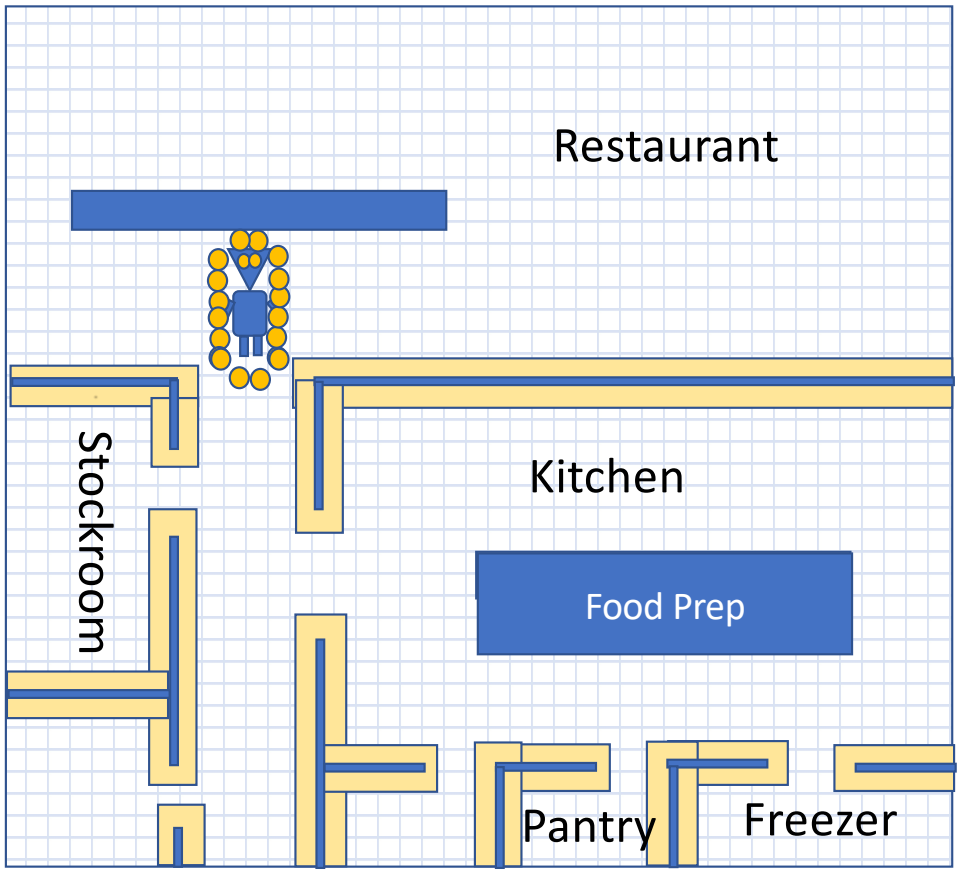
- Option #1: every node in the search tree carries information about the size of the robot.



OK, that’s a little unwieldy…

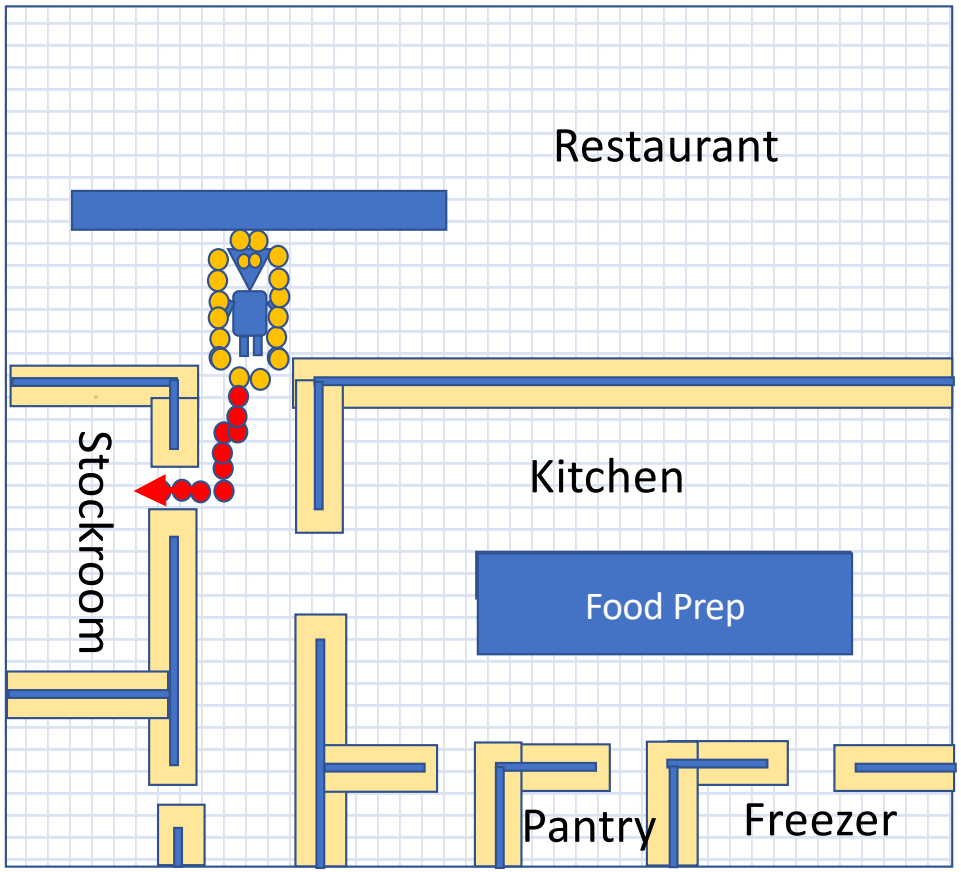
How can we help the robot?

- Option #2: the map tells the robot how wide it is.



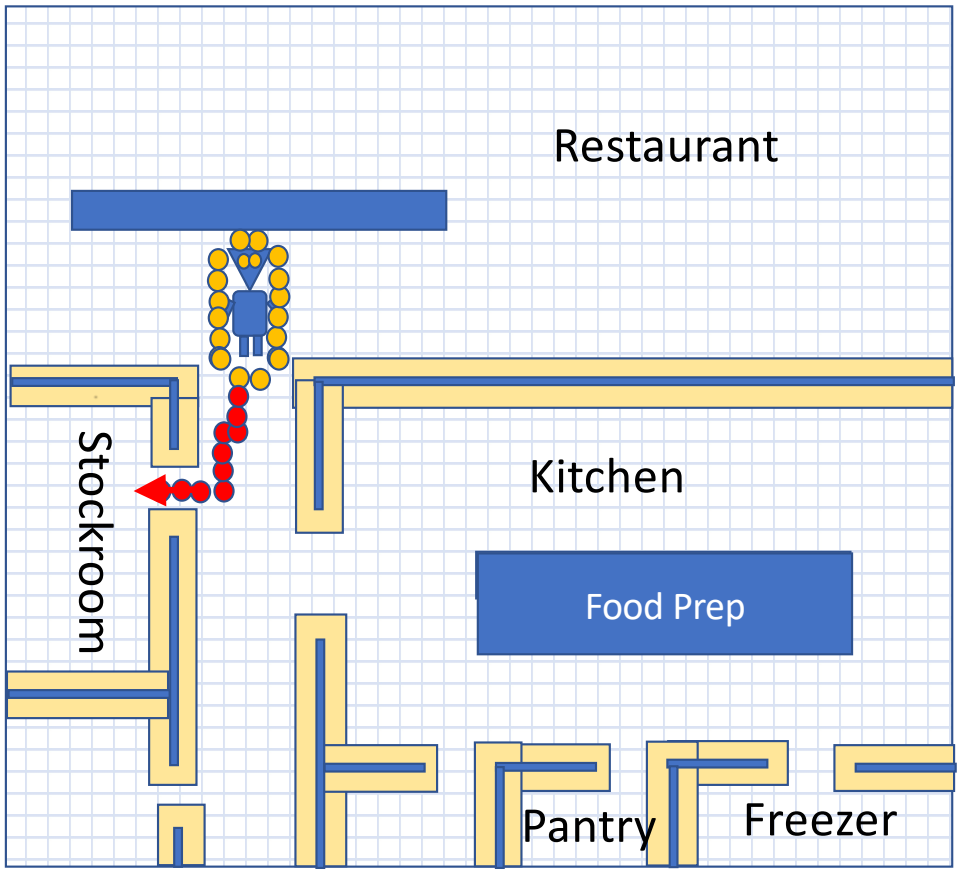
How can we help the robot?

- Option #2: the map tells the robot how wide it is.
- Now, any optimal path that the robot finds is a path that it can actually use.



Configuration Space

- This new search space is called a **configuration space**.
- It specifies which configurations are possible.



Configuration Space

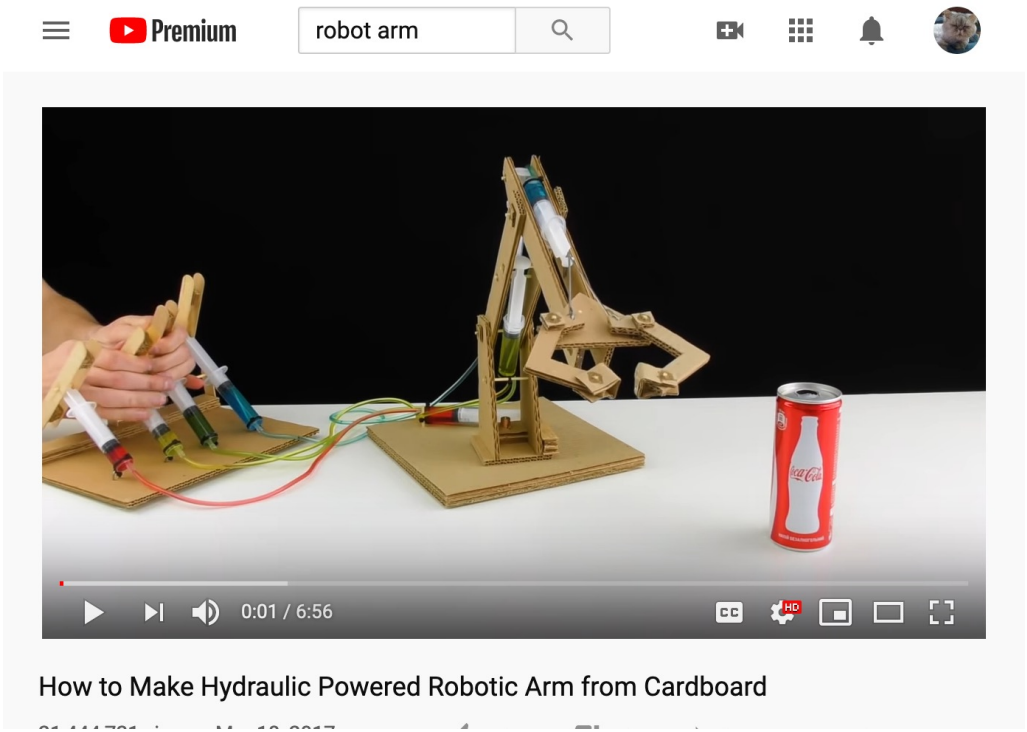
In classical mechanics,

- the parameters that define the configuration of a system are called **generalized coordinates**, and
- the vector space defined by these coordinates is called the **configuration space**.

https://en.wikipedia.org/wiki/Configuration_space_(physics)

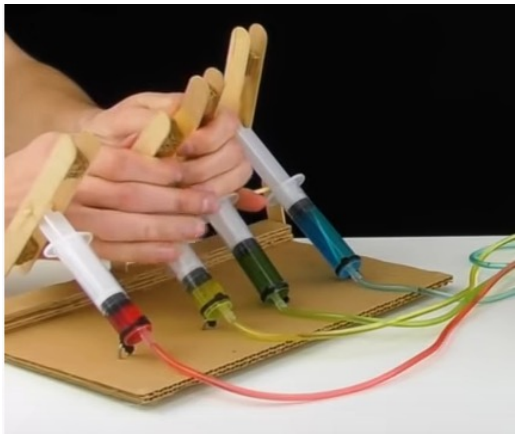
Configuration Space Example: Robot Arm

https://www.youtube.com/watch?v=P2r9U4wkjcc

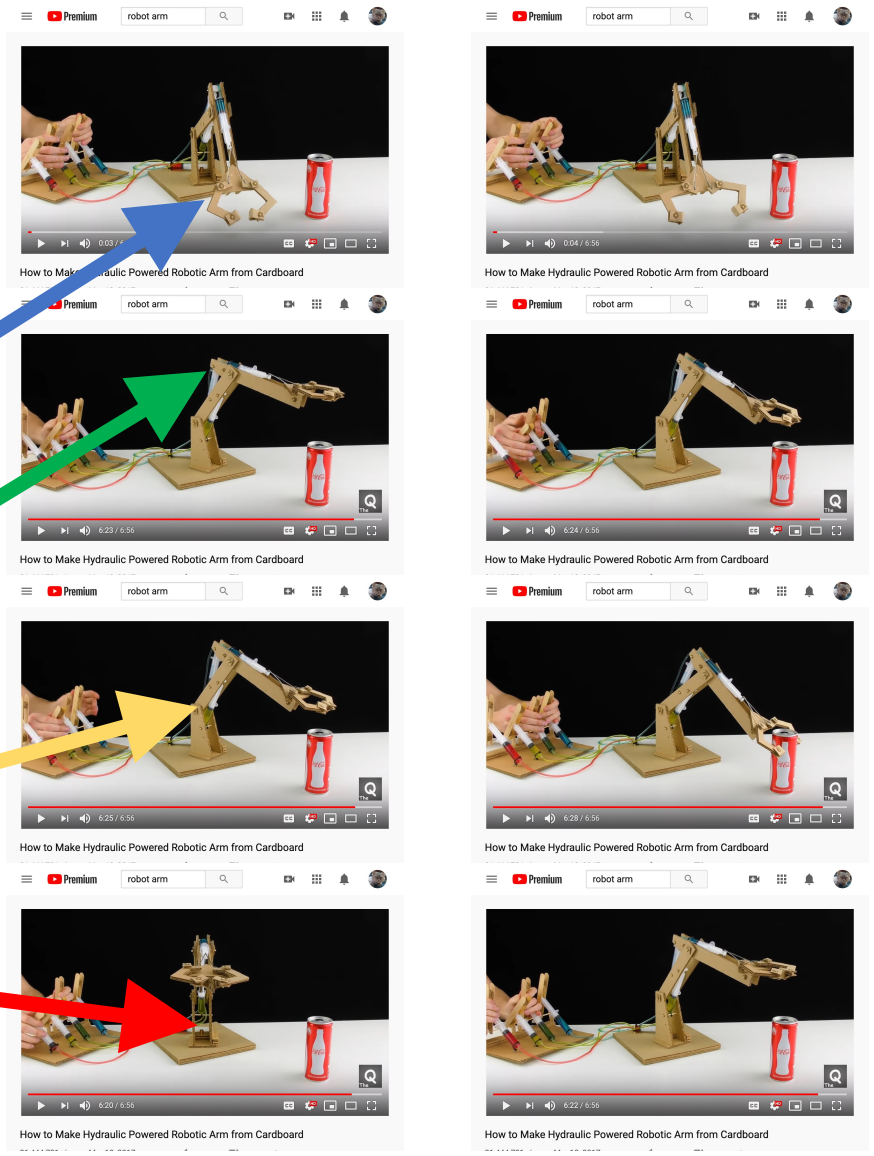


Configuration Space Example

Configuration space:
4 coordinates

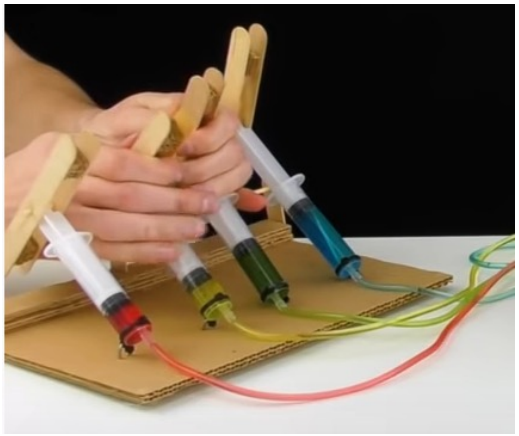


1. Grip
2. Elbow
3. Shoulder
4. Rotation

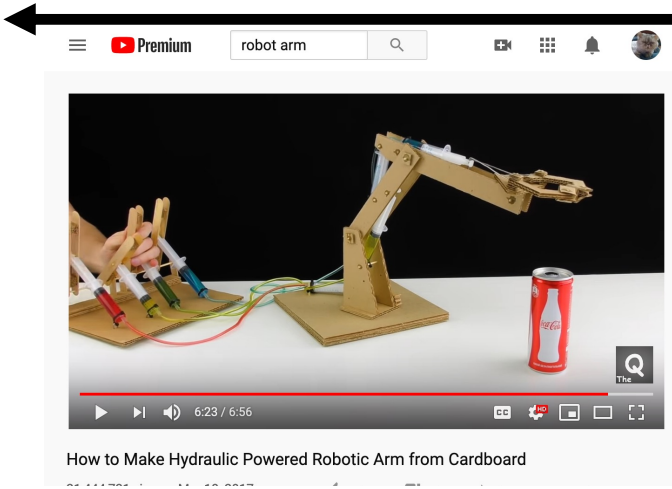


The MP2 Configuration Space:

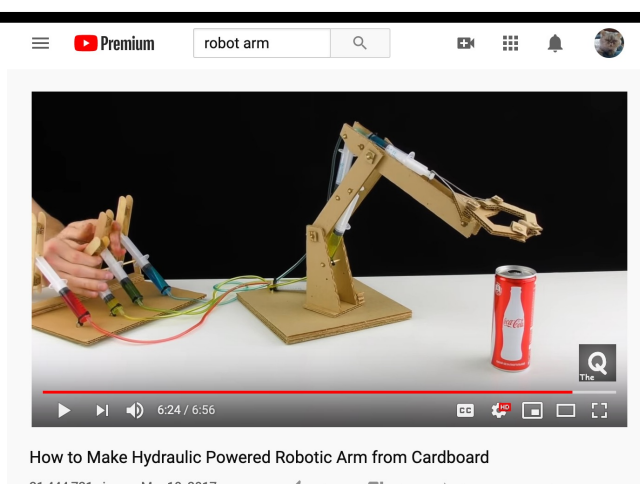
Just 2 coordinates



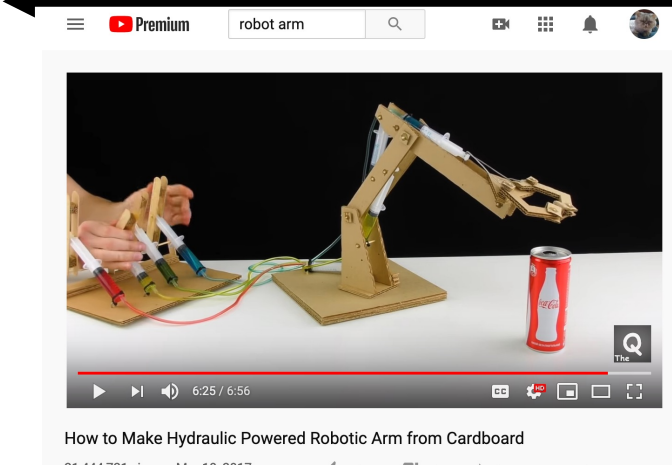
Elbow ≈ 120°



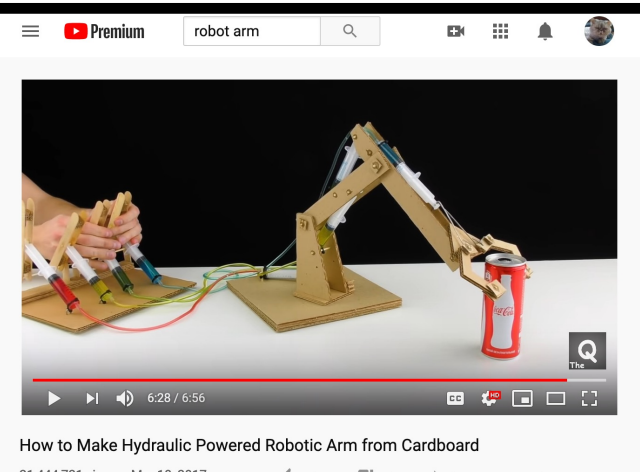
Elbow ≈ 90°



Shoulder ≈ 60°



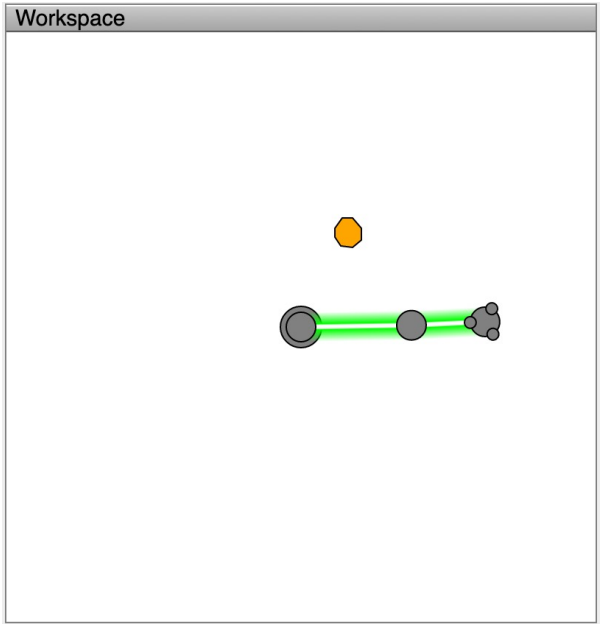
Shoulder ≈ 45°



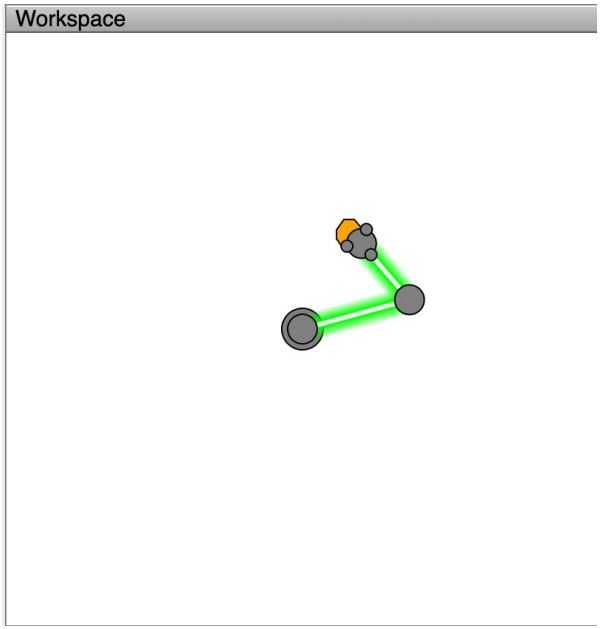
The Robot Arm Reaching Problem

Jeff Ichnowski, University of North Carolina, https://www.cs.unc.edu/~jeffi/c-space/robot.xhtml

- Given a robot arm in START,
- how should I adjust ELBOW and SHOULDER to most quickly reach GOAL?



START



GOAL

The Robot Arm Reaching Problem

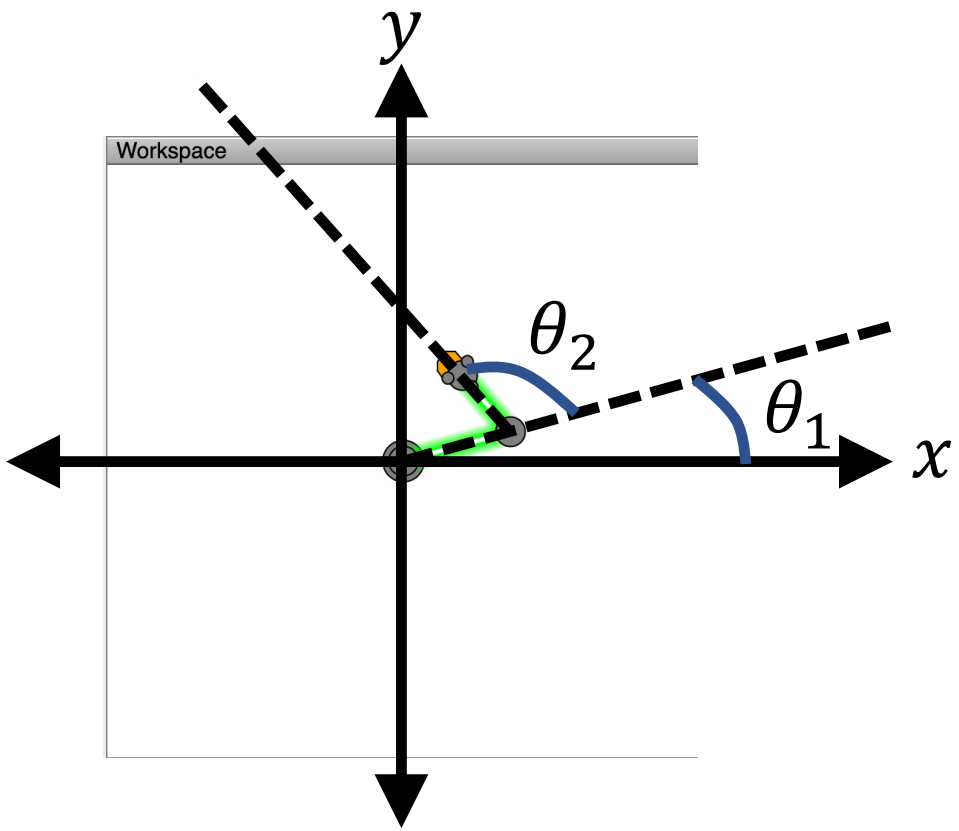
Jeff Ichnowski, University of North Carolina, https://www.cs.unc.edu/~jeffi/c-space/robot.xhtml

Define some variables:

- θ_1 = shoulder angle
- L_1 = length of upper arm
- θ_2 = elbow angle
- L_2 = length of lower arm

Then

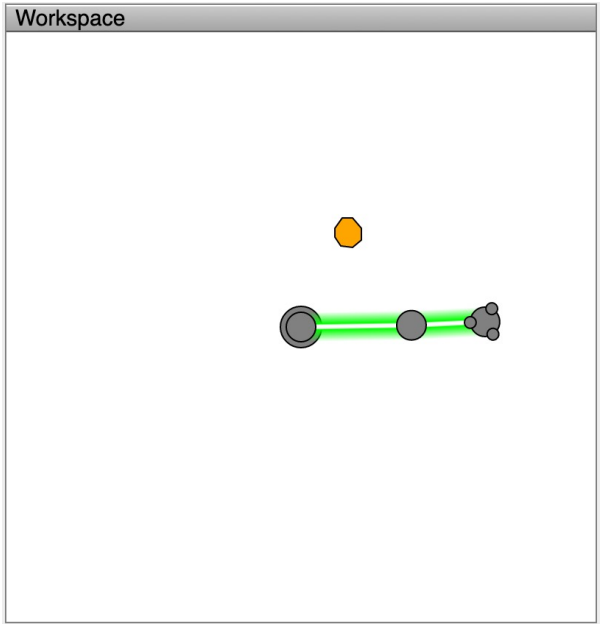
$$x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2)$$
$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2)$$



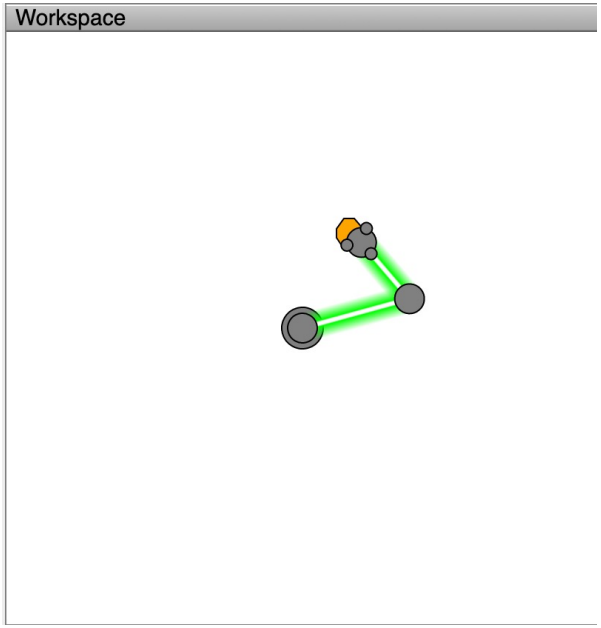
The Robot Arm Reaching Problem

Jeff Ichnowski, University of North Carolina, https://www.cs.unc.edu/~jeffi/c-space/robot.xhtml

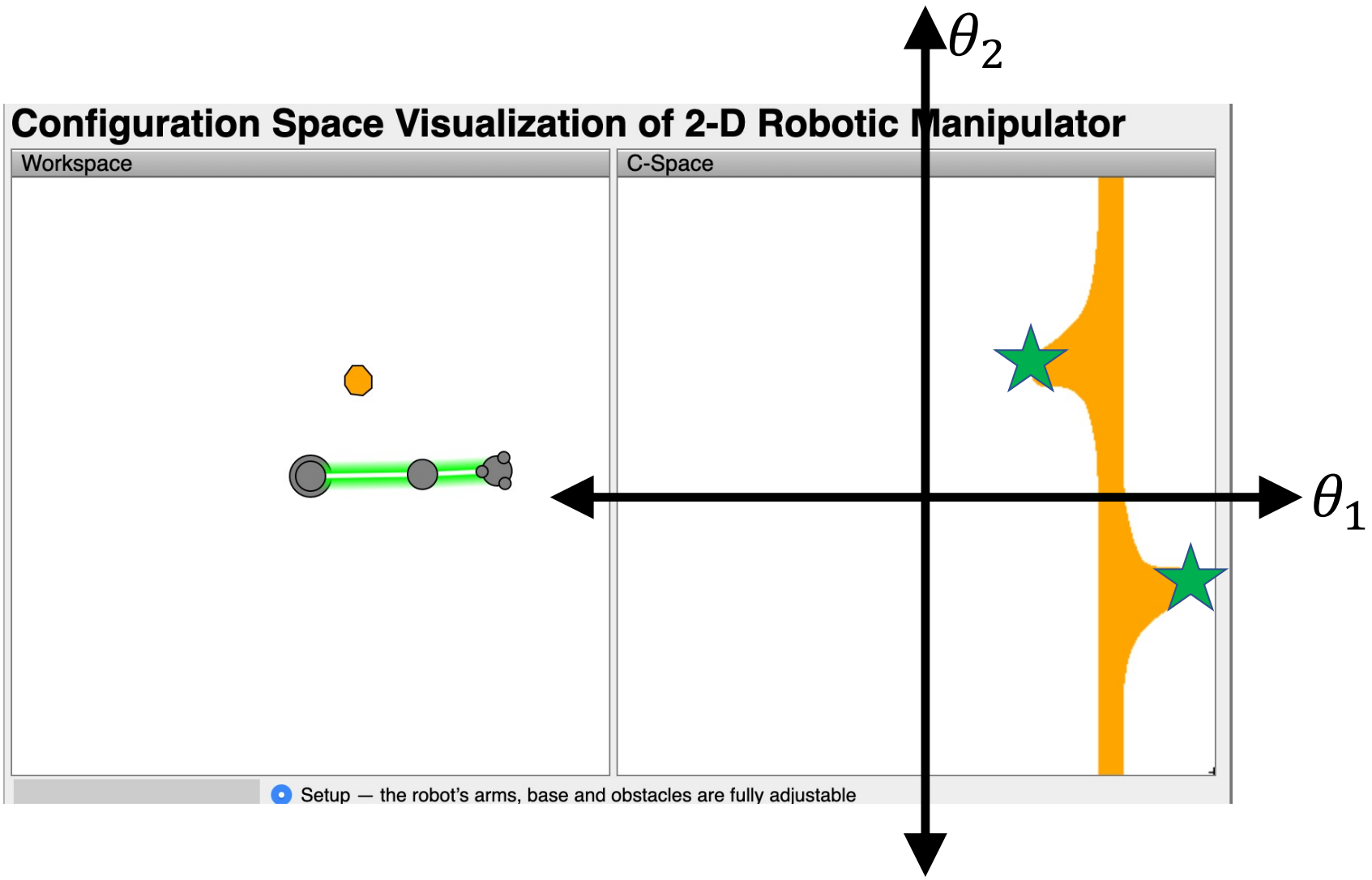
- Given a robot arm in STARTING VALUES OF (θ_1, θ_2),
- how should I adjust (θ_1, θ_2) to most quickly reach GOAL VALUES OF (x, y)?

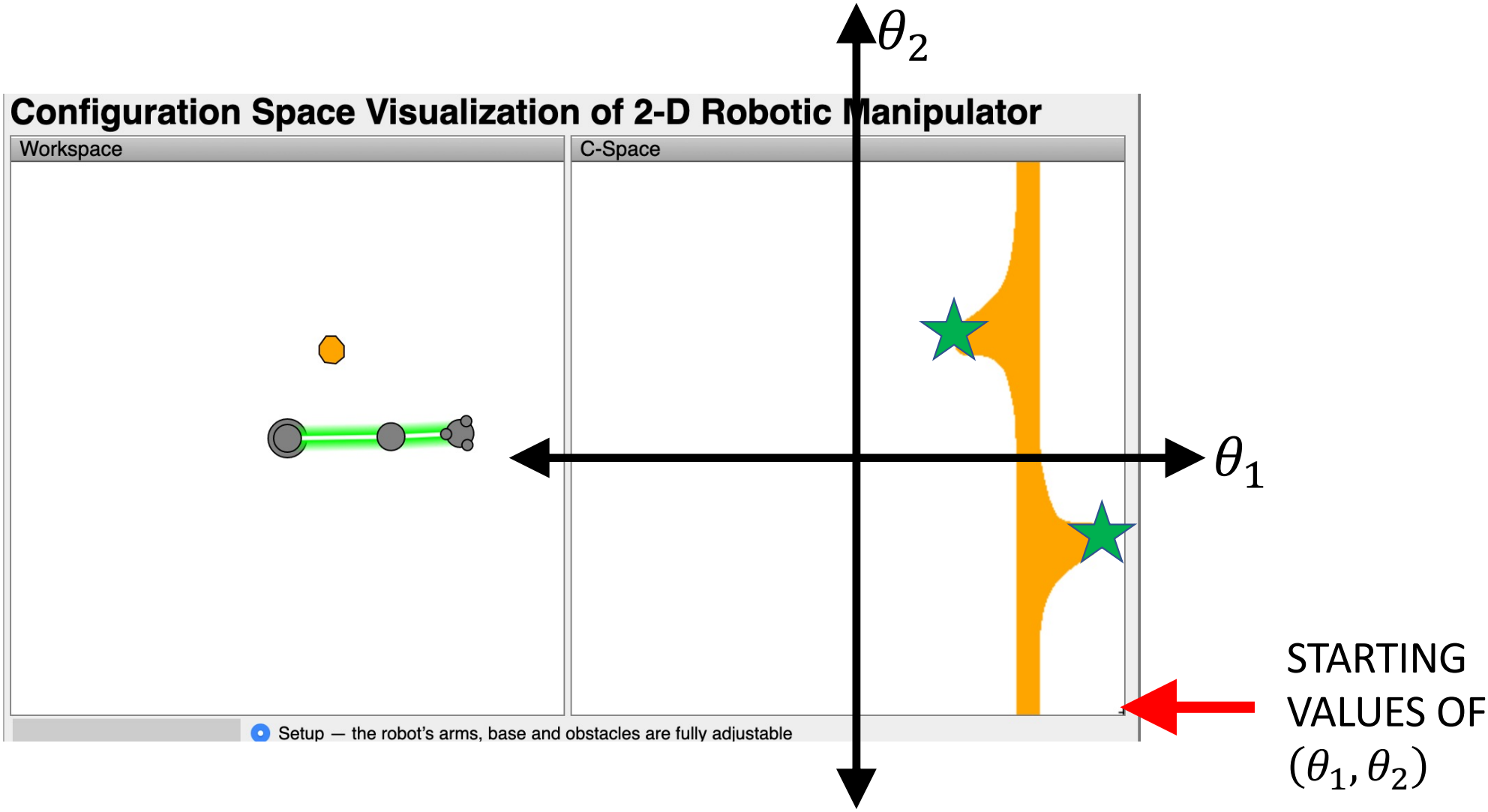


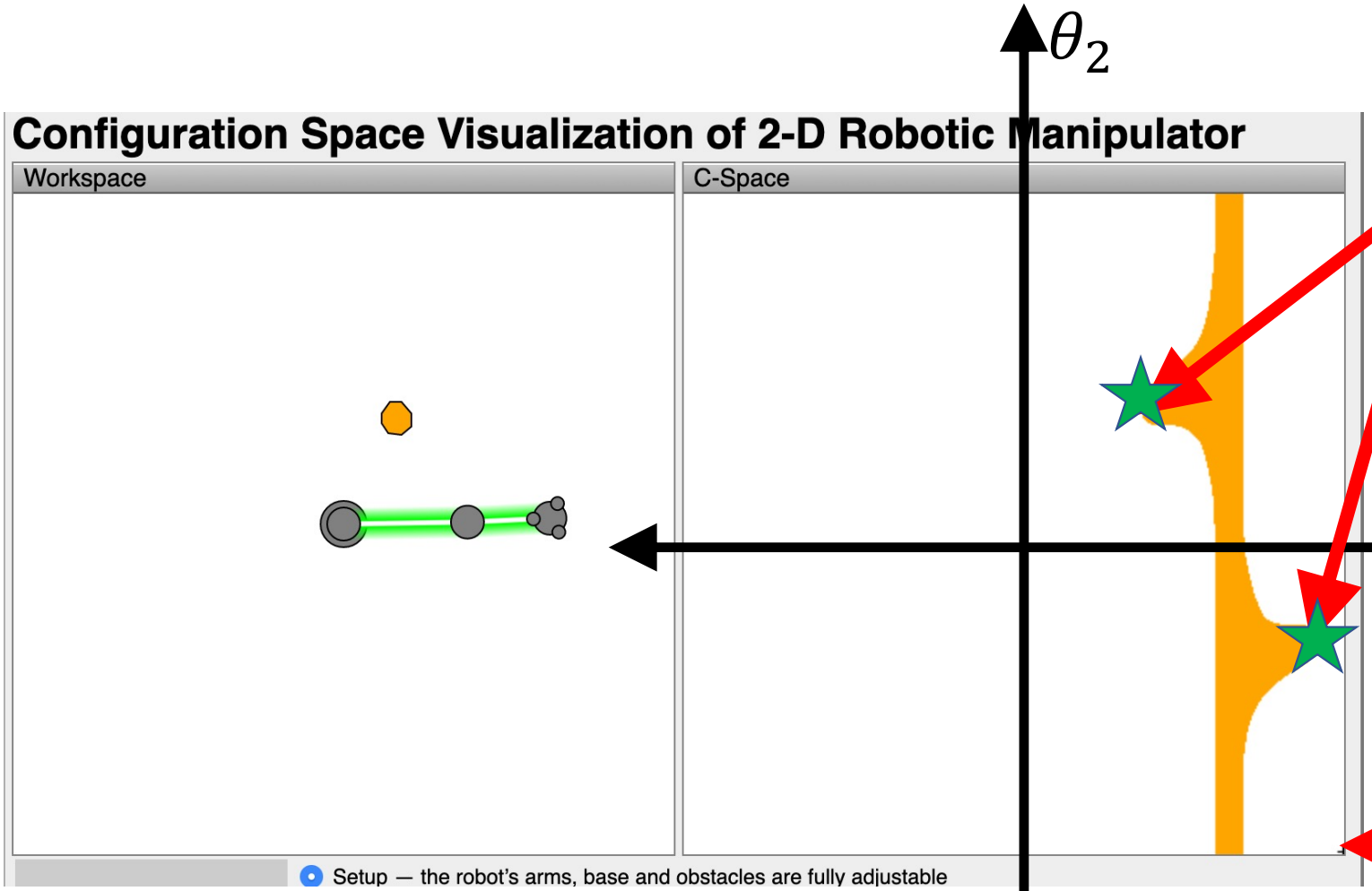
START



GOAL


Configuration Space Visualization of 2-D Robotic Manipulator
Workspace
C-Space
Setup — the robot's arms, base and obstacles are fully adjustable
θ_2
θ_1


Configuration Space Visualization of 2-D Robotic Manipulator
Workspace
C-Space
Setup — the robot's arms, base and obstacles are fully adjustable
θ_2
θ_1
STARTING VALUES OF (θ_1, θ_2)

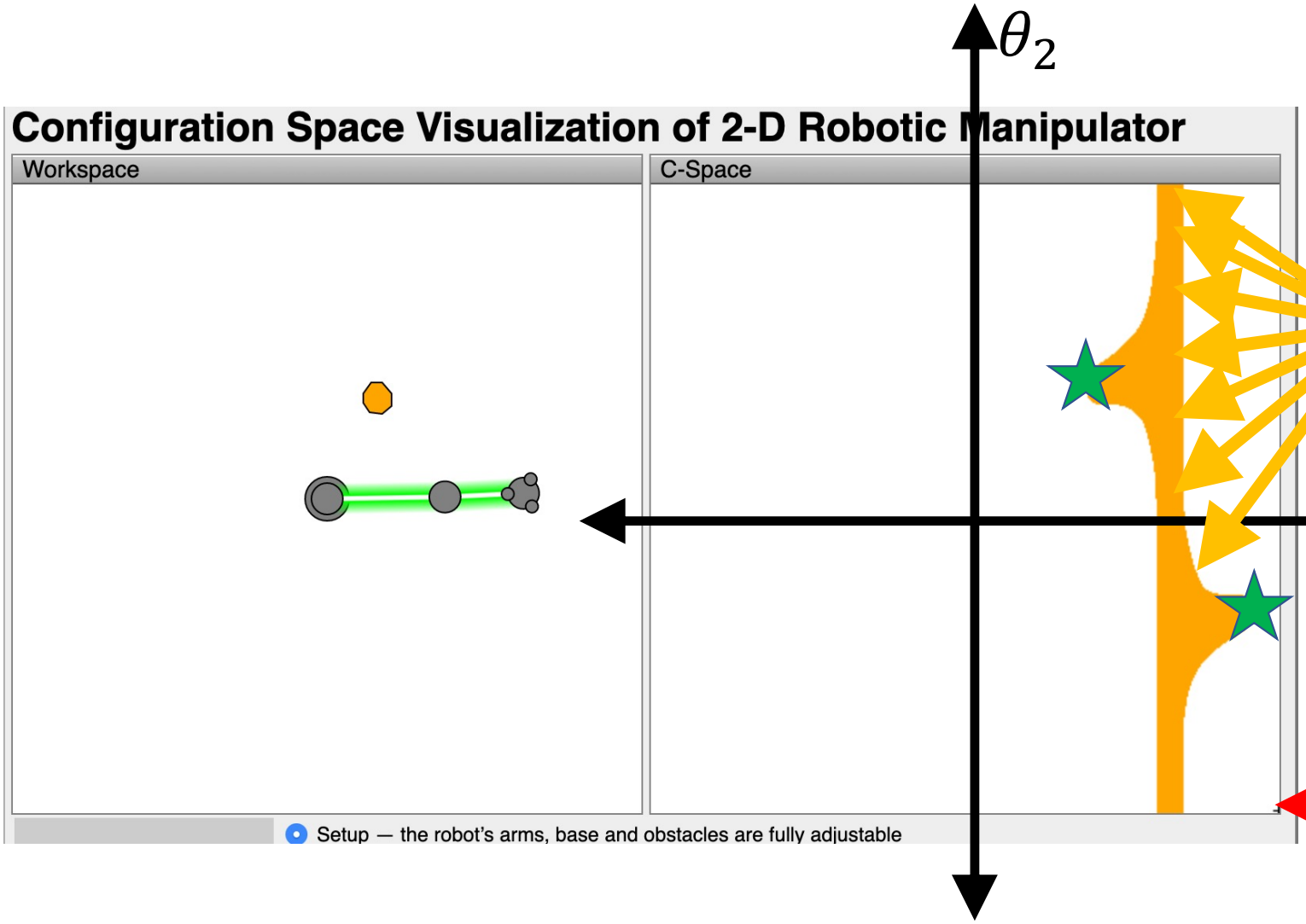


θ_2

θ_1

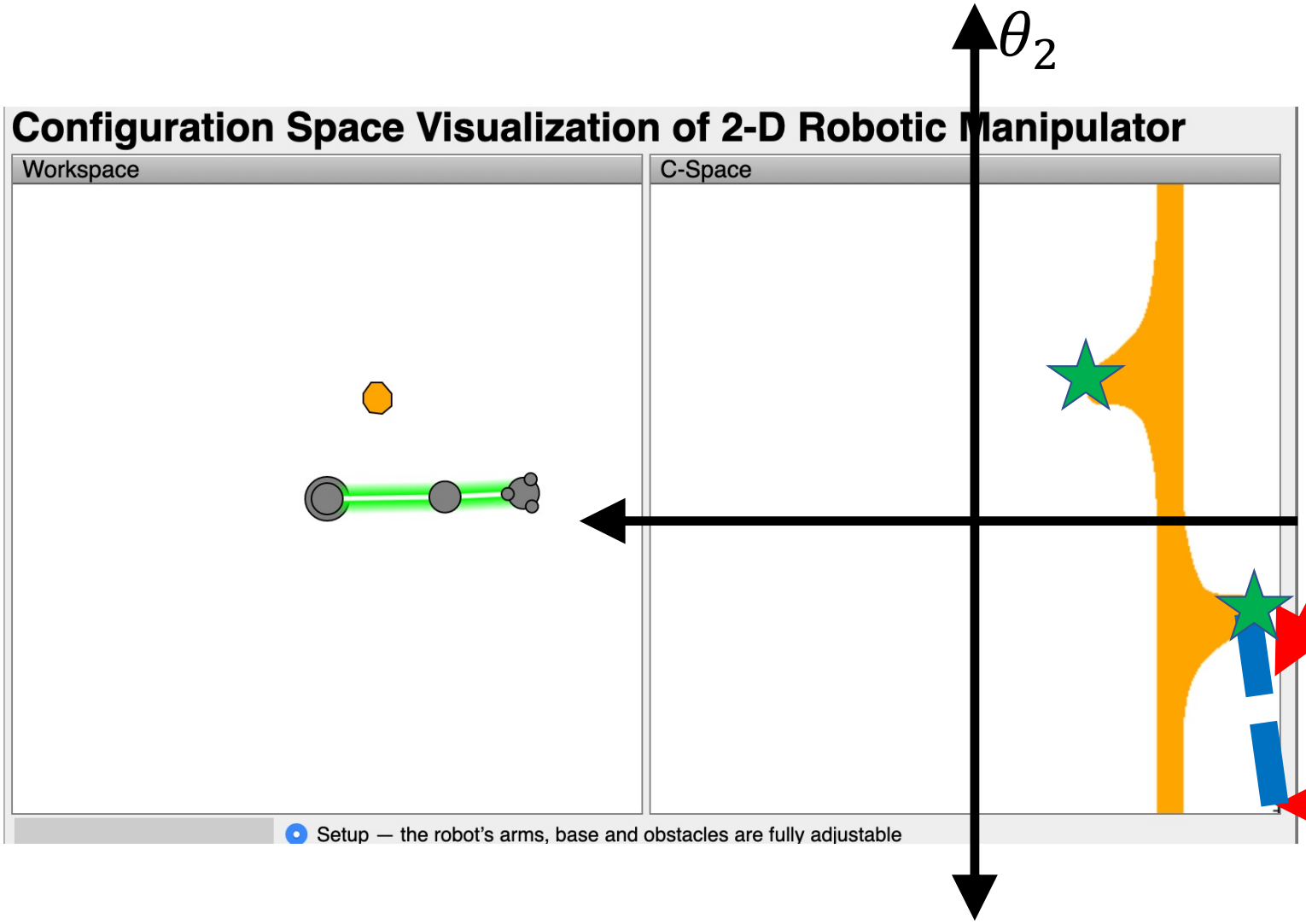
THESE VALUES OF (θ_1, θ_2) CORRESPOND TO REACHING THE GOAL

STARTING VALUES OF (θ_1, θ_2)



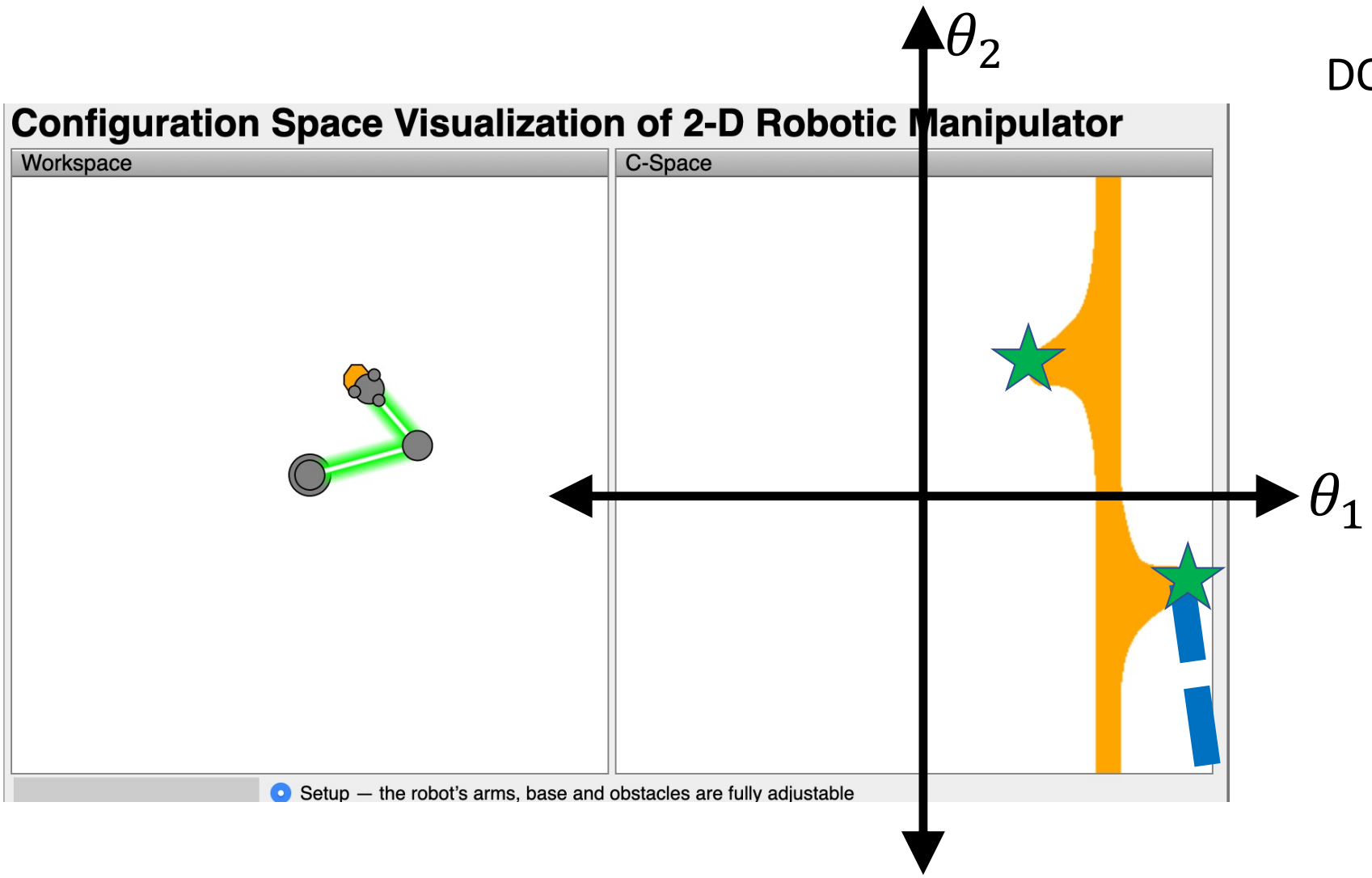
Bumping anywhere else on the orange bar means that the upper or lower arm collides, not the hand. Call these regions OBSTACLES.

STARTING VALUES OF (θ_1, θ_2)



This is the shortest path from START to GOAL avoiding all OBSTACLES

STARTING VALUES OF (θ_1, θ_2)


θ_2
Configuration Space Visualization of 2-D Robotic Manipulator
Workspace
C-Space
θ_1
Setup — the robot's arms, base and obstacles are fully adjustable

DONE!

How to solve the Robot Arm problem

1. Create a configuration space (a space whose coordinates are the set of all configuration parameters for the robot. Two dimensions, for the MP; in the real world, 3d or 4d is more common).
2. Label the START
3. Label the GOAL (there might be more than one set of configuration parameters that is an acceptable way to reach the GOAL).
4. Label the OBSTACLES (convert them from (x, y) to (θ_1, θ_2)).
5. Use BFS or A* to find the shortest path from START to GOAL, avoiding all OBSTACLES.

BTW, this person is really a maestro. Watch it again, if you want to.

https://www.youtube.com/watch?v=P2r9U4wkjcc