2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

# F1/10 Autonomous Racing

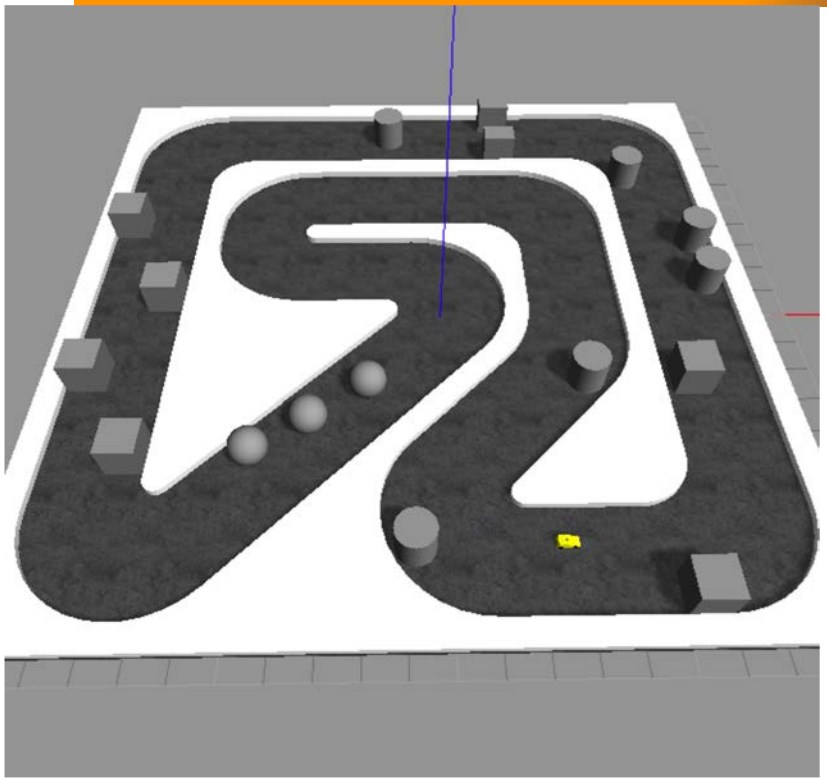
## Follow-the-gap

**Madhur Behl**
Computer Science | Engineering Systems and Environment | UVA Link Lab

madhur.behl@virginia.edu

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

# Wall following limitation – obstacles



- Build a map, and plan an obstacle free path to follow.

- Locally avoid obstacles (reactive approach) without any knowledge of the map.

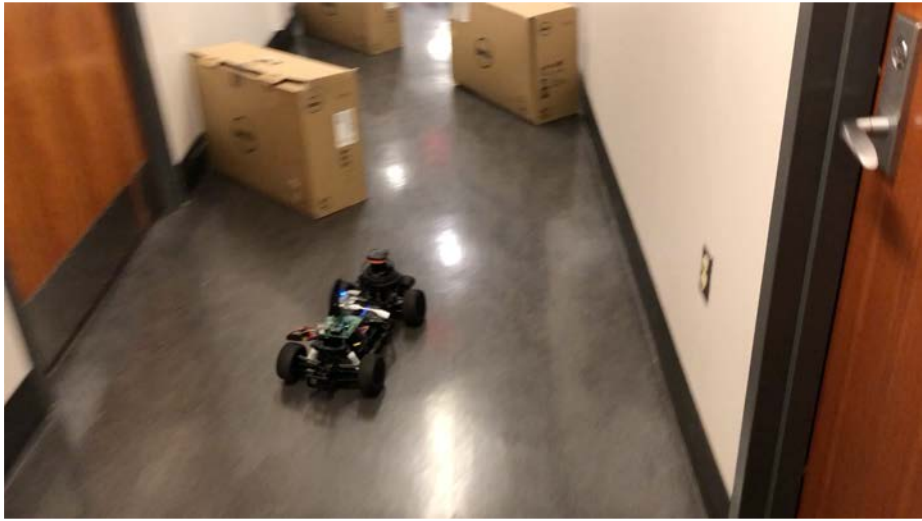
# Obstacle Avoidance: Follow the gap

**Reactive Navigation:**

Use immediate sensor input to decide driving command

**Planning for obstacle avoidance:**

Use LIDAR for both static and dynamic obstacle avoidance



# Race: Reactive Methods

**Race Format:**
Time-trial; single car on track

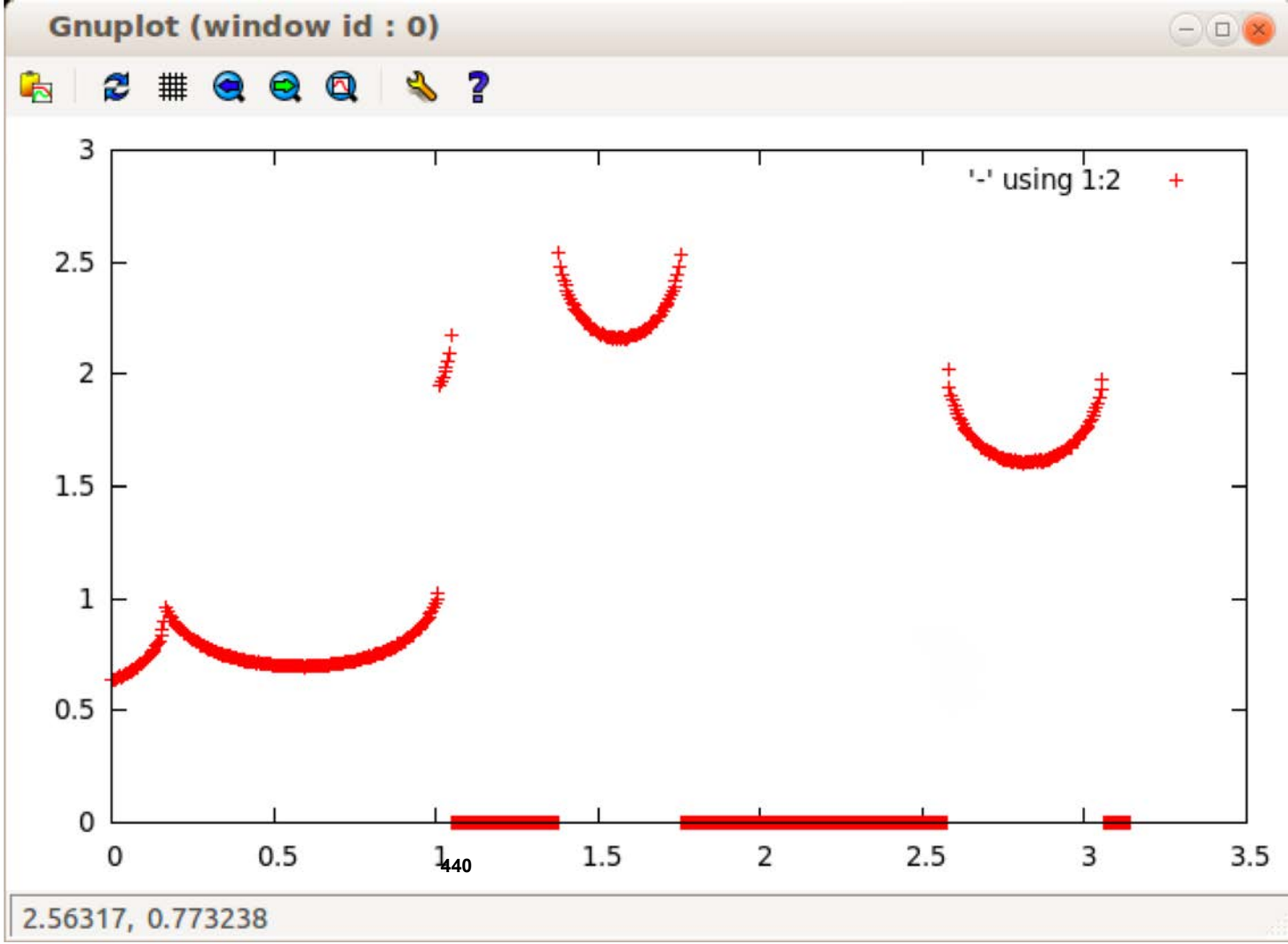
**Penalties:** Crashing

**Baseline:** Complete 5 laps without crashing

**Example Video:** CPS Week 2018

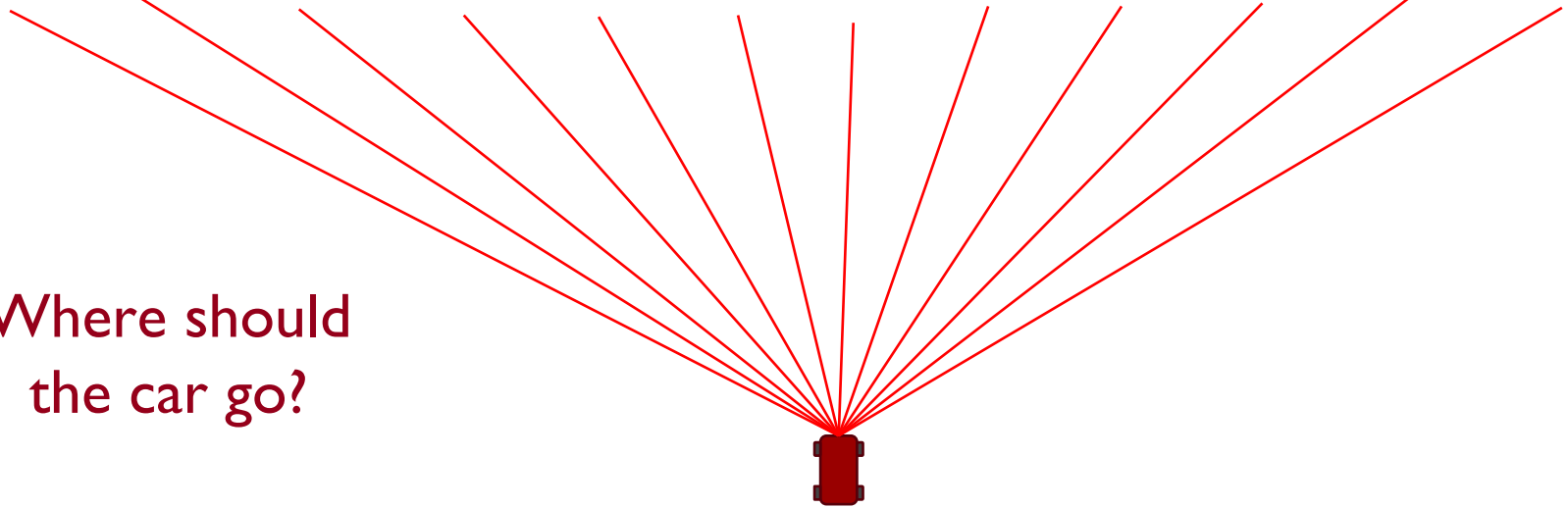
# How do we get to that level of performance without a map?

Where should the car go ?



# Follow the Gap

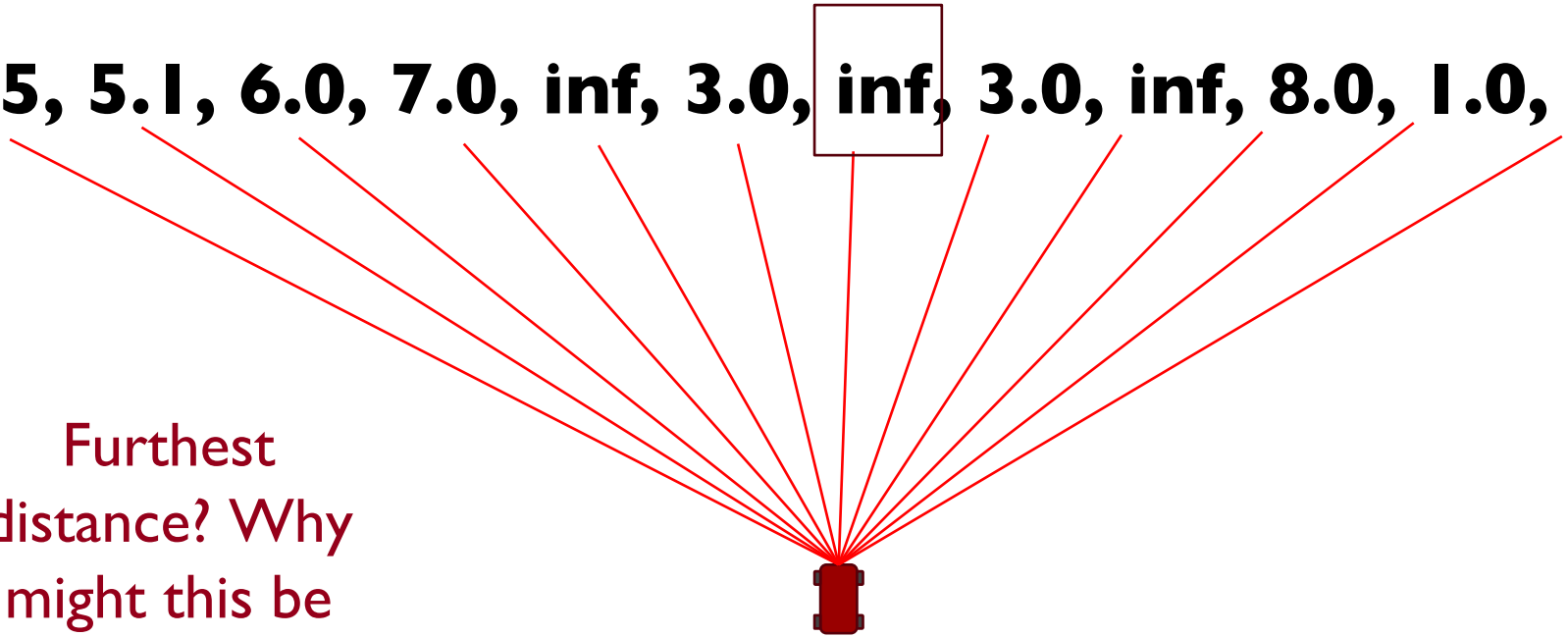
**[0.5, 5.1, 6.0, 7.0, inf, 3.0, inf, 3.0, inf, 8.0, 1.0, 3.0]**



Where should the car go?

# Follow the Gap

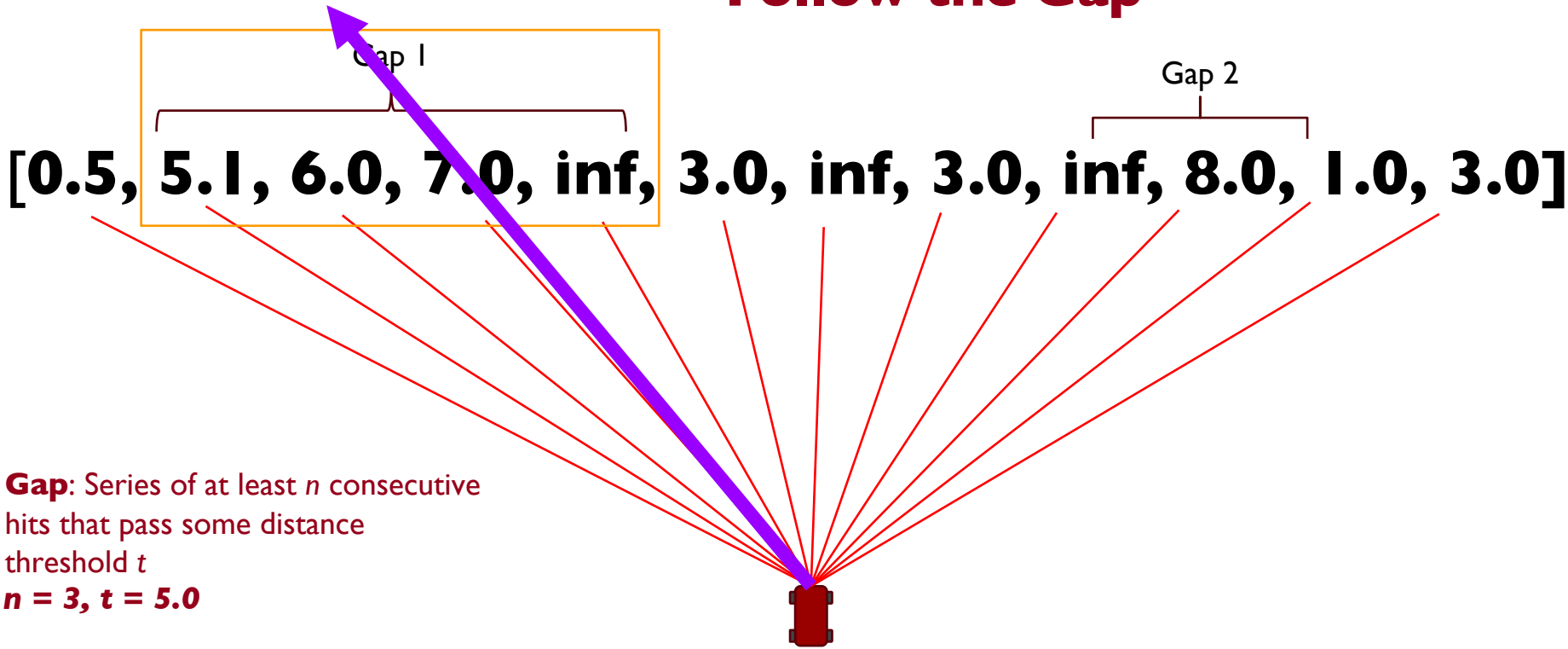
**[0.5, 5.1, 6.0, 7.0, inf, 3.0, inf, 3.0, inf, 8.0, 1.0, 3.0]**



Furthest distance? Why might this be wrong?

442

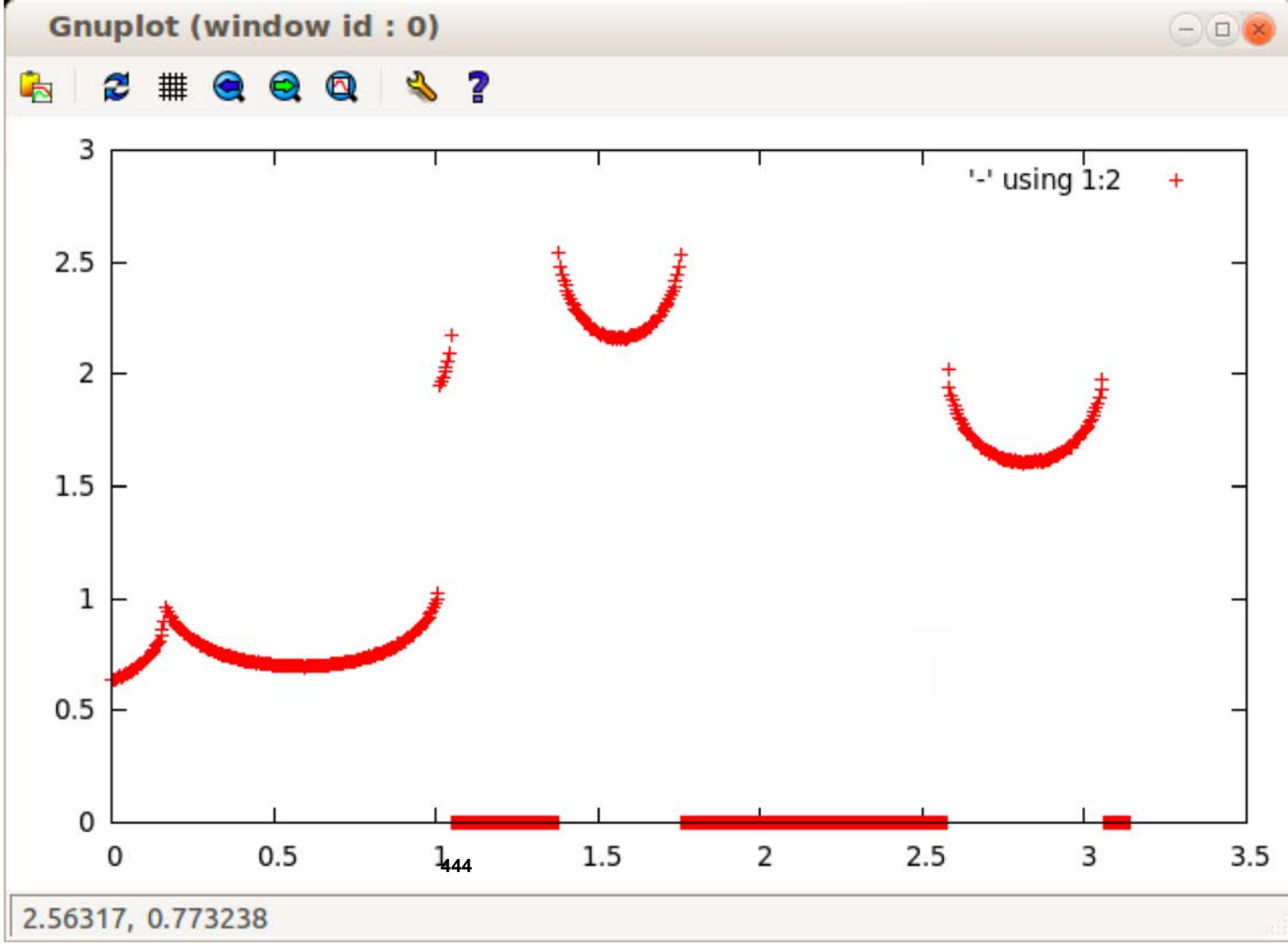
# Follow the Gap




**Gap**: Series of at least *n* consecutive hits that pass some distance threshold *t*

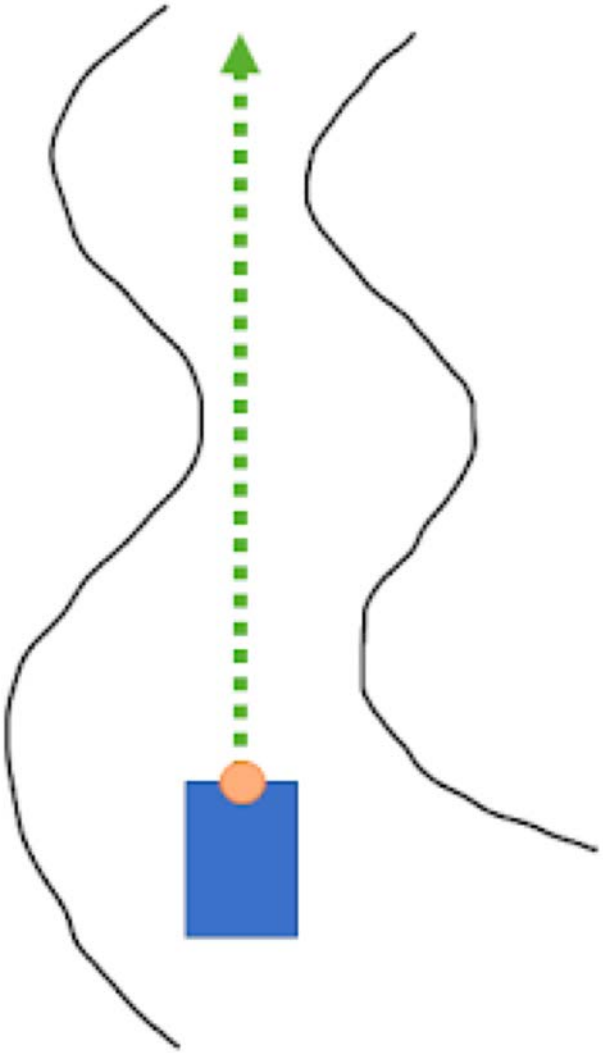
***n* = 3, *t* = 5.0**

Where should the car go ?



# Gap finding

- **Find the gaps**
- Calculate the width of each gap
- Determine the widest gap
- Optional: Determine the “deepest” gap

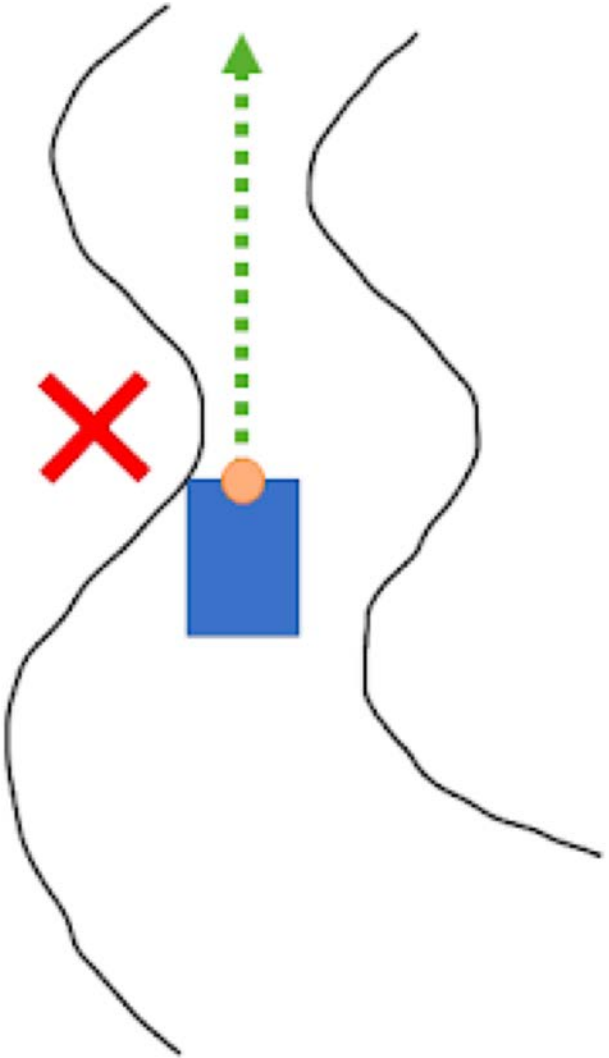


# Follow-the-gap : Intuition

Heading in the longest possible straight line is good for "cutting corners"

if the angle exceeds the car's steering angle that we just turn as sharply as possible towards the correct angle

446



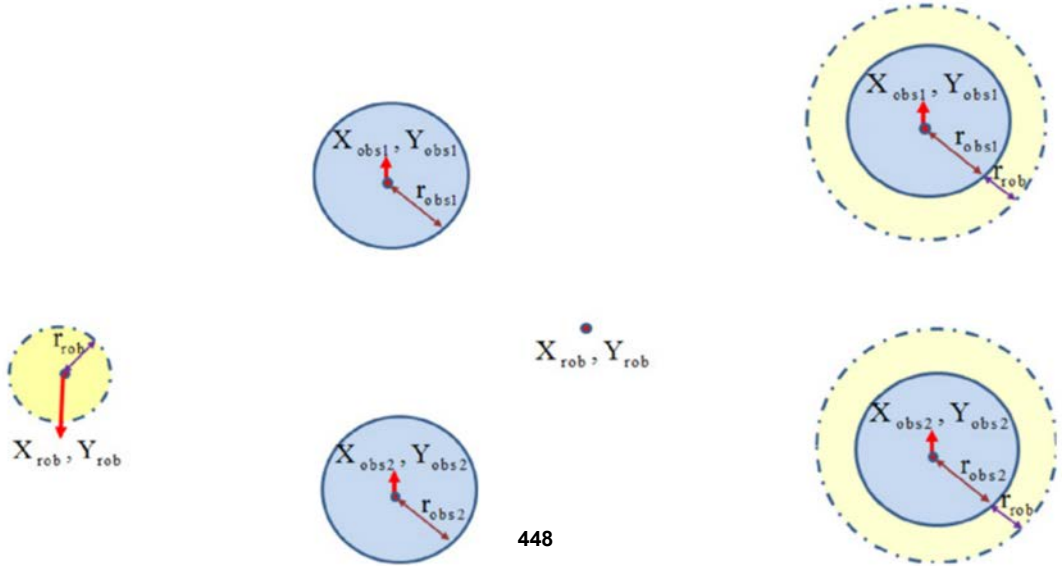
# Follow-the-gap : Intuition

Purely following a straight line fails to account for the fact that you're a car, not an infinitesimally small point...

447

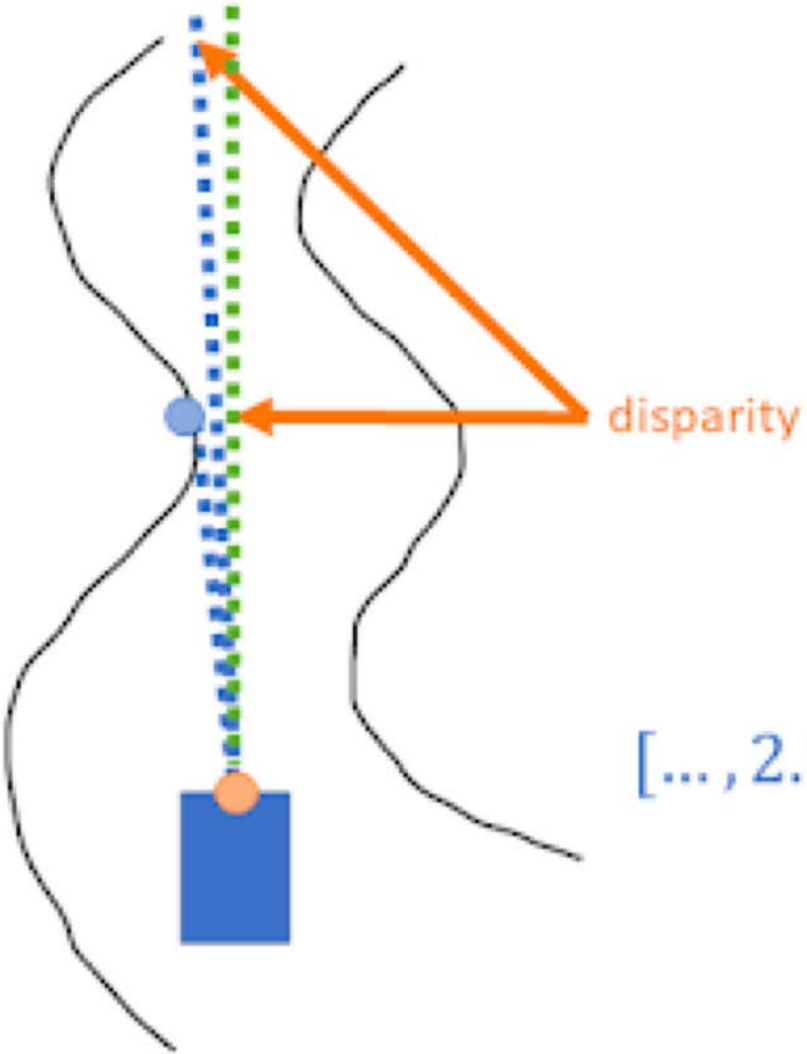
# Point Robot Approach

- Robot and Obstacles are assumed circular.
- Radius of robot is added to radius of obstacles
- The Robot is reduced to a point, while Obstacles are equally enlarged.




(a) Circular robot with circular obstacles. (b) Point robot with enlarged obstacles.

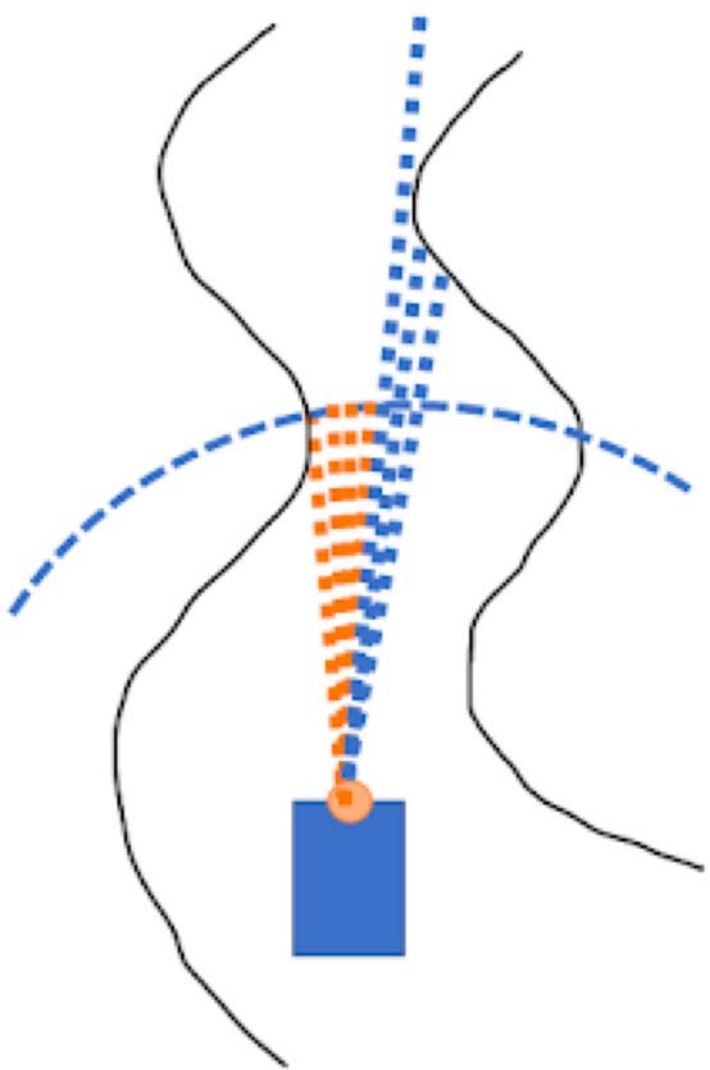
# Follow-the-gap : Intuition




We start with LIDAR readings (presented as an array of floating-point distances), and look for consecutive readings that differ by an amount over some threshold.

$$[\ldots, 2.1, 2.2, 4.8, 4.9, \ldots]$$

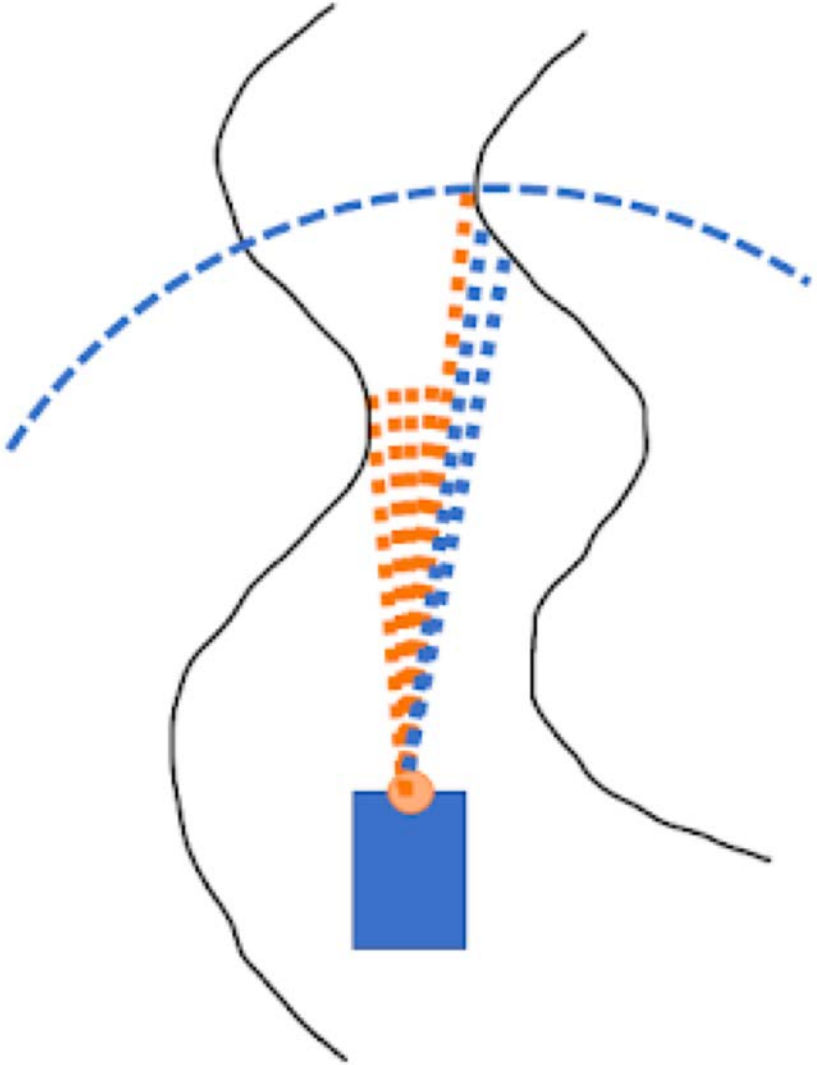
449



# Follow-the-gap : Intuition

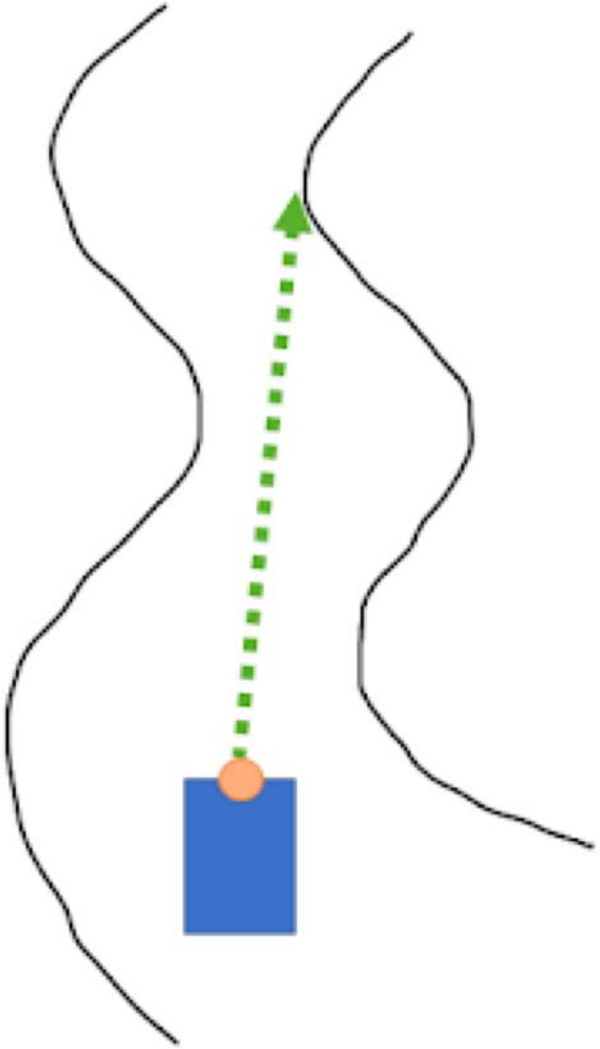
We mask over LIDAR readings (shown in orange) in order to make them appear shorter in the "filtered" array--these should approximate points that the car can actually reach (more or less)

450



# Follow-the-gap : Intuition

Repeat this process for every disparity, but never overwrite closer "distances" with farther ones

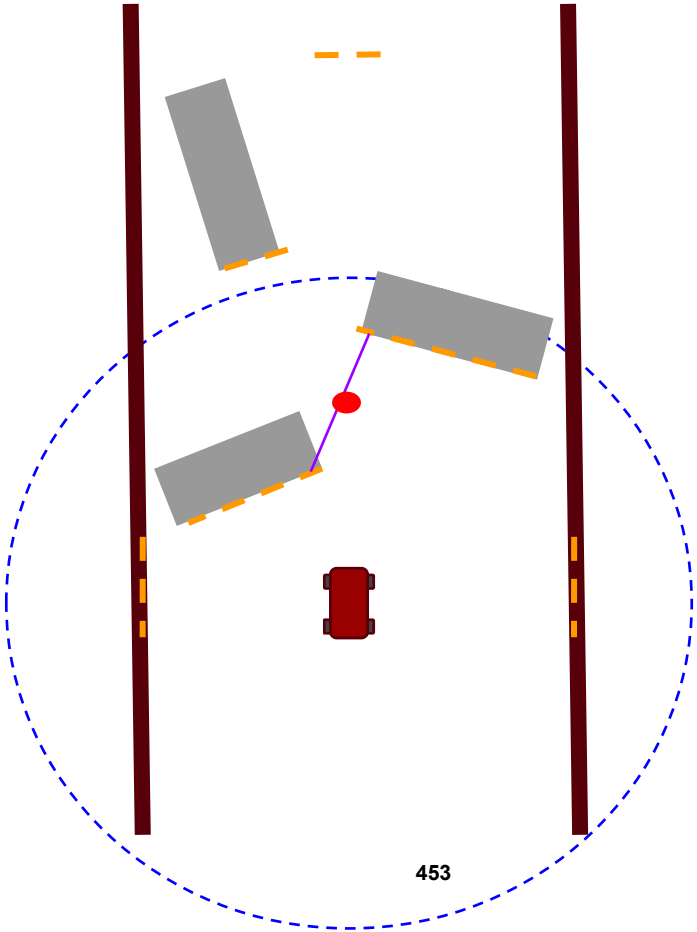


# Follow-the-gap : Intuition

In the "filtered" array of distances, the longest path should be something that's actually reachable (or at least close to reachable)

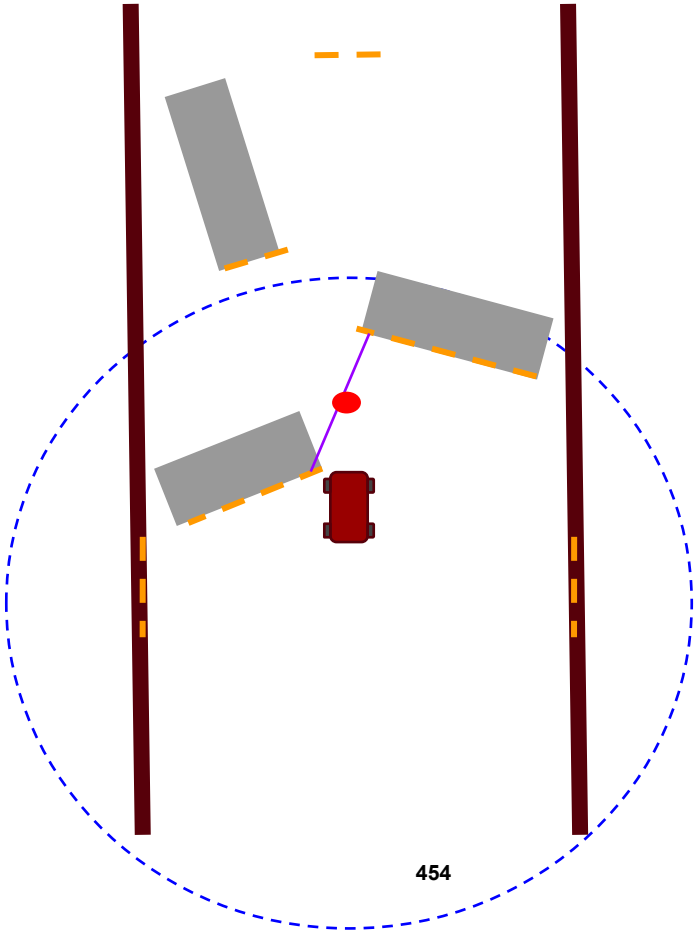
452

Why Naive "Follow the Gap" doesn't work



Why “Find the Largest Gap” doesn’t work

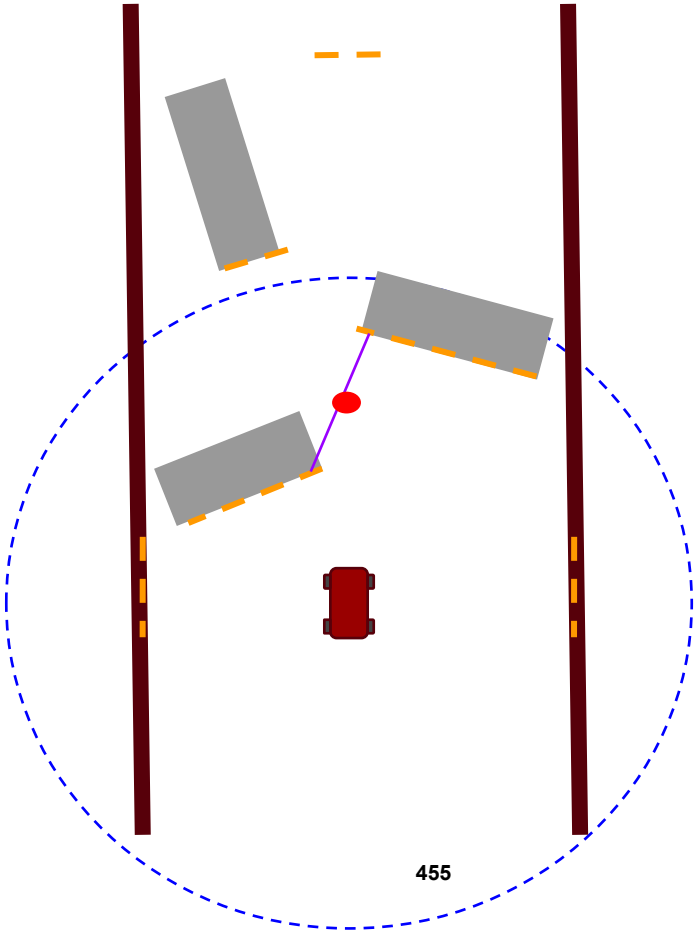
Threshold



The Idea:

"Seek out the largest gap"

Why "Find the Largest Gap" doesn't work

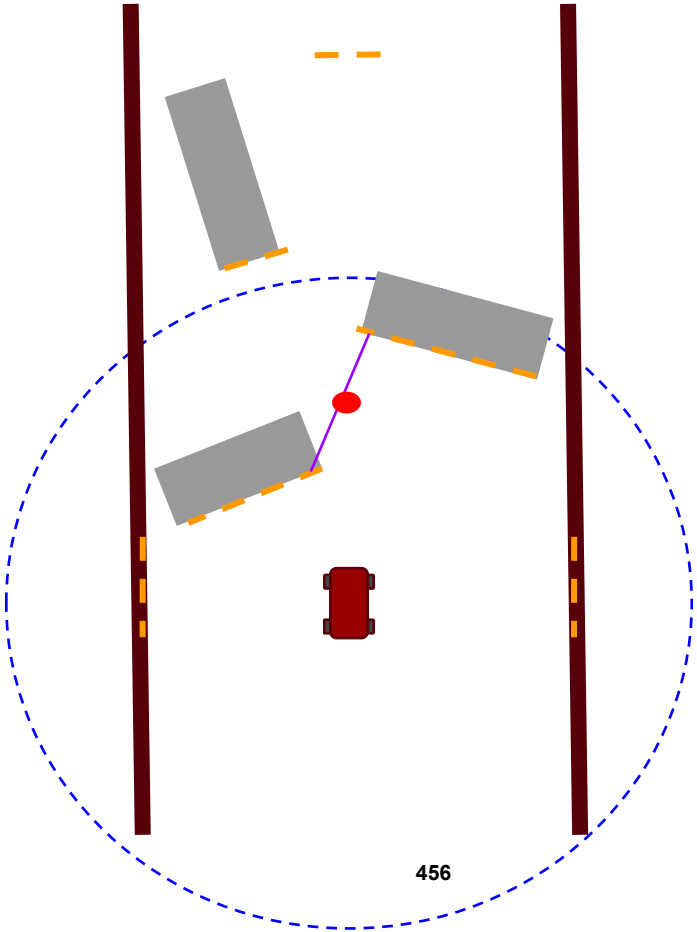


The Idea:

"Seek out the largest gap"

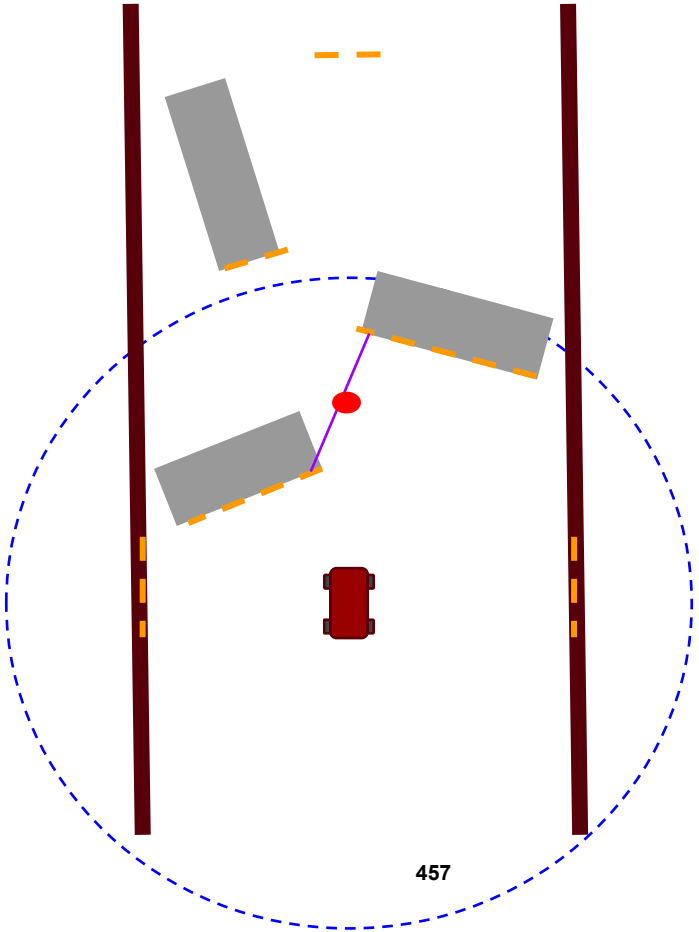
Works fine for holonomic robots (eg. turtlebots)

Why "Find the Largest Gap" doesn't work



Why "Find the Largest Gap" doesn't work

Threshold



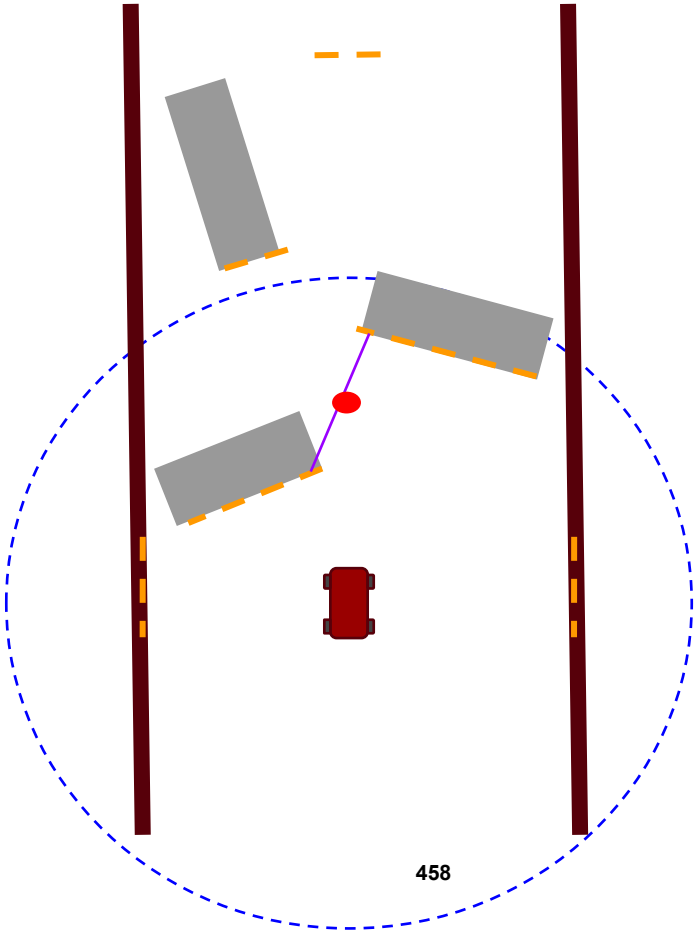

The Idea:

"Seek out the largest gap"

Works fine for holonomic robots (eg. turtlebots)

Works fine for non-holonomic robots in environments with sparse obstacles

Why "Find the Largest Gap" doesn't work




The Idea:

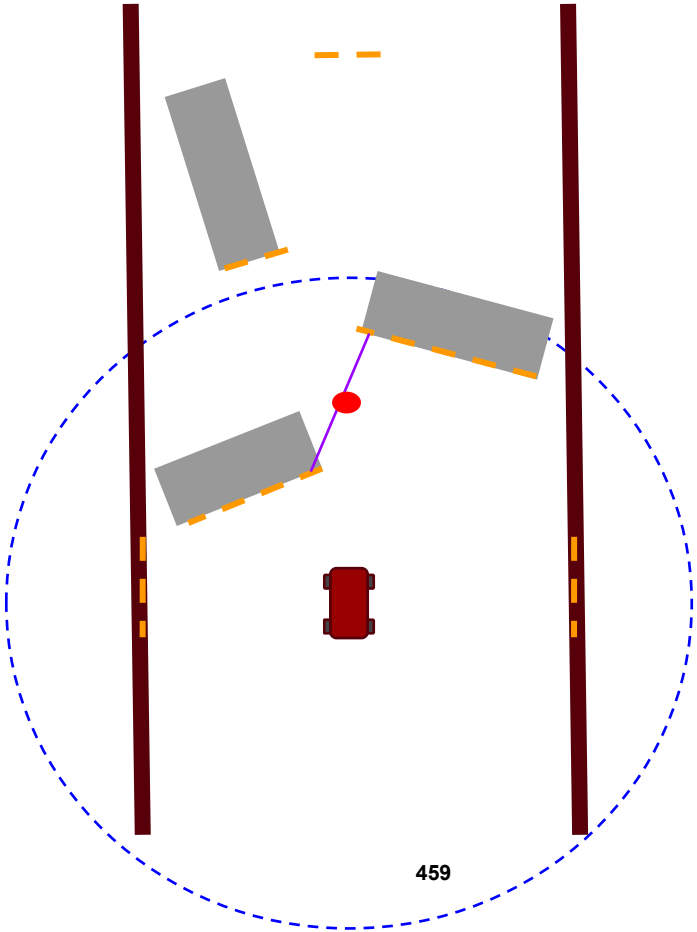
"Seek out the largest gap"

Works fine for holonomic robots (eg. turtlebots)

Works fine for non-holonomic robots in environments with sparse obstacles

Doesn't optimize for safety

Why "Find the Largest Gap" doesn't work




The Idea:

"Seek out the largest gap"

Works fine for holonomic robots (eg. turtlebots)

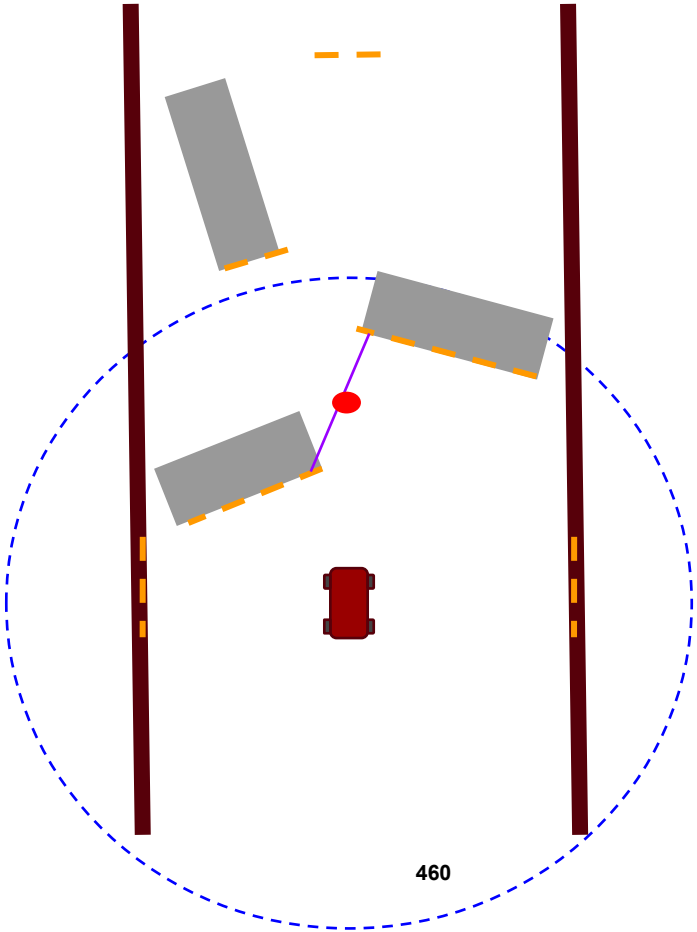
Works fine for non-holonomic robots in environments with sparse obstacles

Doesn't optimize for safety

Doesn't consider car's dimensions

Why "Find the Largest Gap" doesn't work

Threshold




The Idea:

"Seek out the largest gap"

Works fine for holonomic robots (eg. turtlebots)

Works fine for non-holonomic robots in environments with sparse obstacles

Doesn't optimize for safety

Doesn't consider car's dimensions

Hard to decide threshold t

# F1/10 Follow the Gap

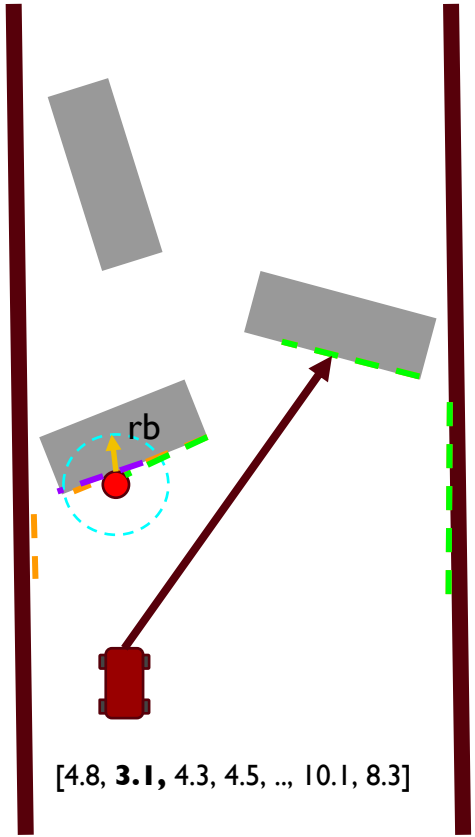
Idea

Every timestep, *cleverly* avoid the nearest obstacle

### Step 1
Find nearest LIDAR point and put a "safety bubble" around it of radius rb

### Step 2
Set all points inside bubble to distance 0. All nonzero points are considered 'free space'

### Step 3
Find maximum length sequence of consecutive non-zeros among the 'free space' points - The max-gap




### Step 4
Find the 'best' point among this maximum length sequence

Naive: Choose the furthest point in free space, and set your steering angle towards it

## Step 1

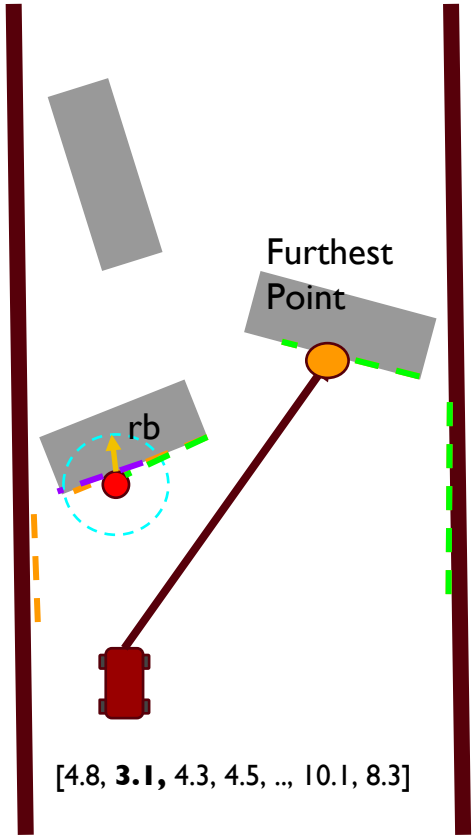
Find nearest LIDAR point and put a "safety bubble" around it of radius rb

## Step 2

Set all points inside bubble to distance 0. All nonzero points are considered 'free space'

## Step 3

Find maximum length sequence of consecutive non-zeros among the 'free space' points - The max-gap




## Step 4

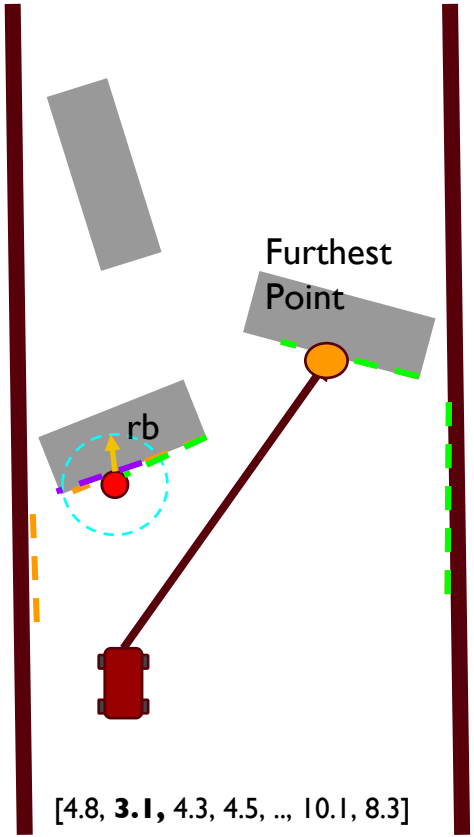
Find the 'best' point among this maximum length sequence

Naive: Choose the furthest point in free space, and set your steering angle towards it

### Step 1
Find nearest LIDAR point and put a "safety bubble" around it of radius rb

### Step 2
Set all points inside bubble to distance 0. All nonzero points are considered 'free space'

### Step 3
Find maximum length sequence of consecutive non-zeros among the 'free space' points - The max-gap




### Step 4
Find the 'best' point among this maximum length sequence

Naive: Choose the furthest point in free space, and set your steering angle towards it

Changing speed results in you losing velocity

## Step 1

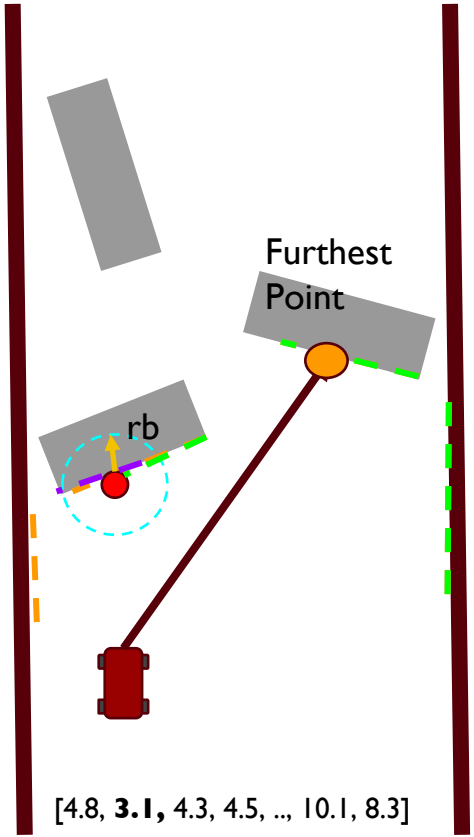
Find nearest LIDAR point and put a "safety bubble" around it of radius rb

## Step 2

Set all points inside bubble to distance 0. All nonzero points are considered 'free space'

## Step 3

Find maximum length sequence of consecutive non-zeros among the 'free space' points - The max-gap




[4.8, **3.1,** 4.3, 4.5, .., 10.1, 8.3]

[4.8, **0.0, 0.0, 0.0**, .., 10.1, 8.3]
465

## Step 4

Find the 'best' point among this maximum length sequence

Naive: Choose the furthest point in free space, and set your steering angle towards it

Changing speed results in you losing velocity

## Better Idea Intuition

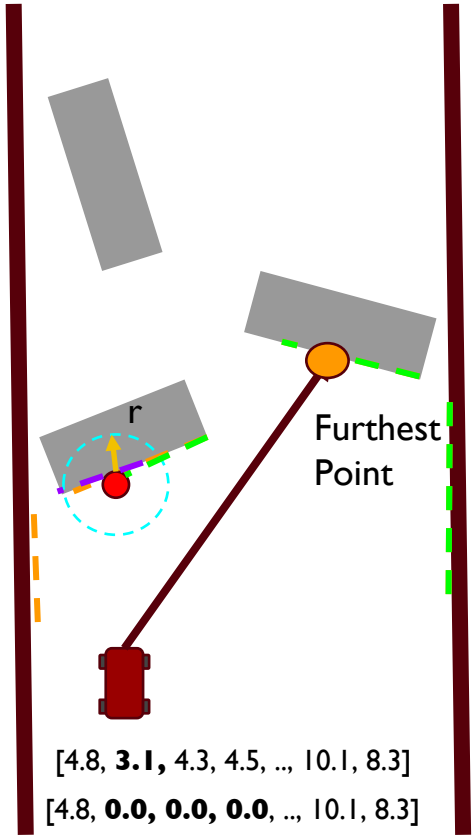
If you're 3-4m away from your closest obstacle, should you immediately make a sharp turn to avoid it?

Step 1
Find nearest LIDAR point and put a “safety bubble” around it of radius r

Step 2
Set all points inside bubble to distance 0. All nonzero points are considered ‘free space’

Step 3
Find maximum length sequence of consecutive non-zeros among the ‘free space’ points - The max-gap

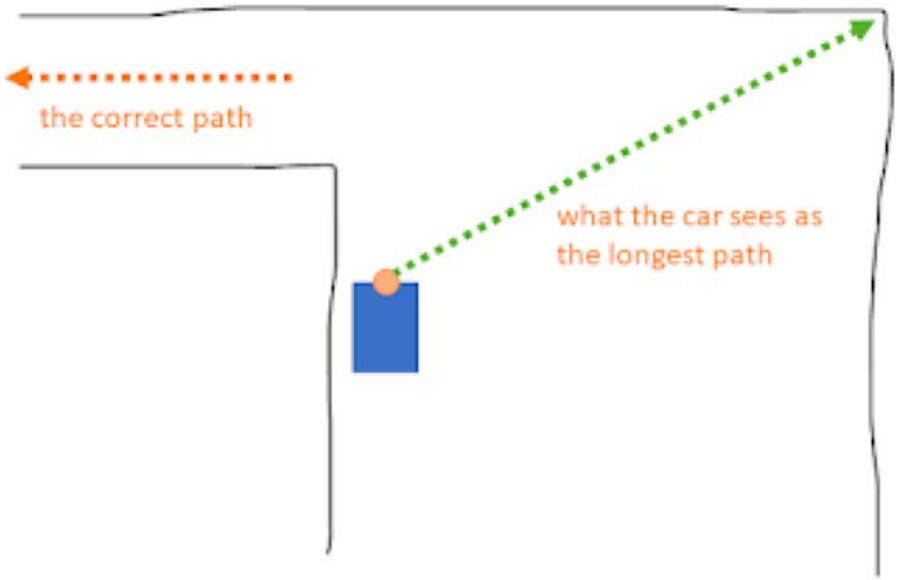
Step 4
Choose the furthest point in free space, and set your steering angle towards it




466

# Where the Algorithm Fails

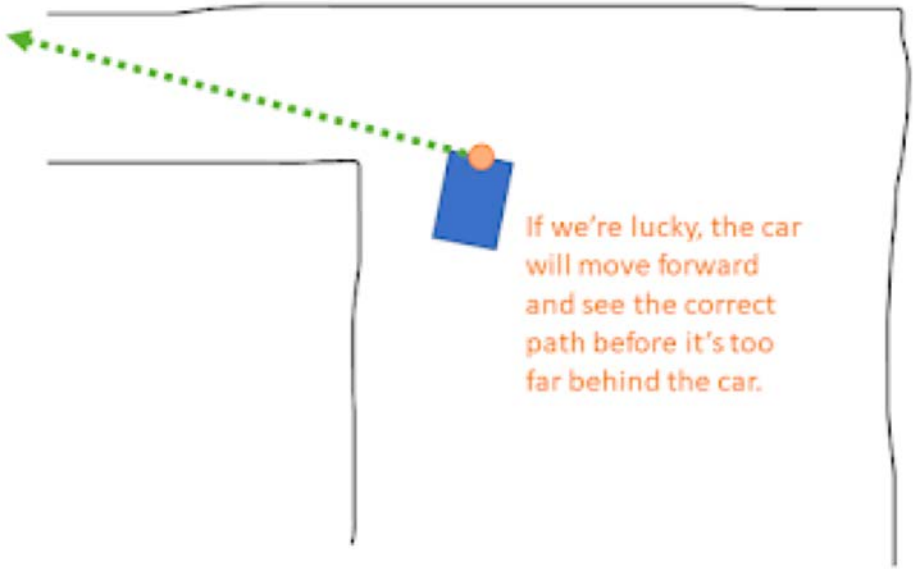
Risk doing a U-turn here



# Where the Algorithm Fails

## Risk doing a U-turn here

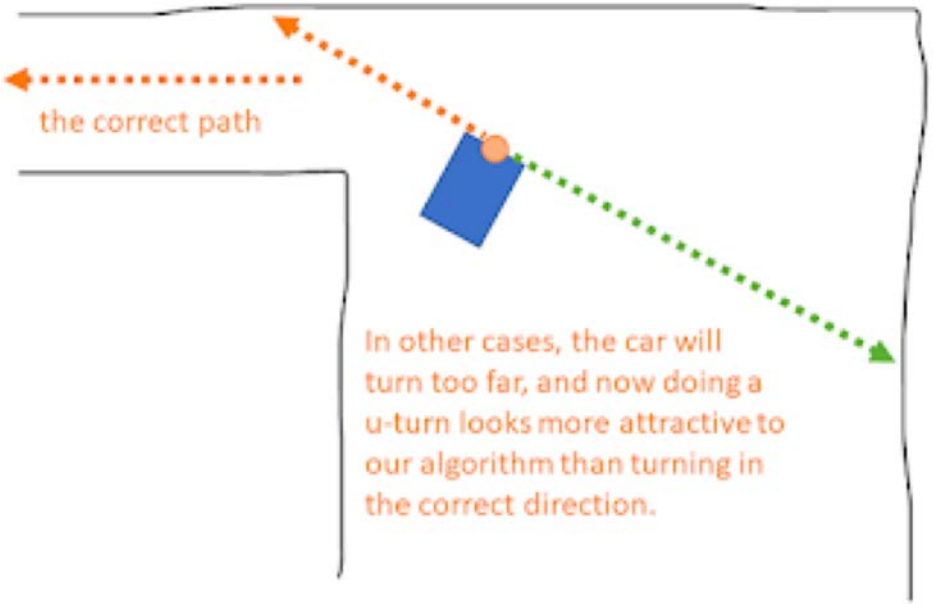
Sometimes the algorithm can narrowly avoid a u-turn in these cases (especially if the car is moving at slower speeds enabling it to react to the correct path more quickly)



# Where the Algorithm Fails

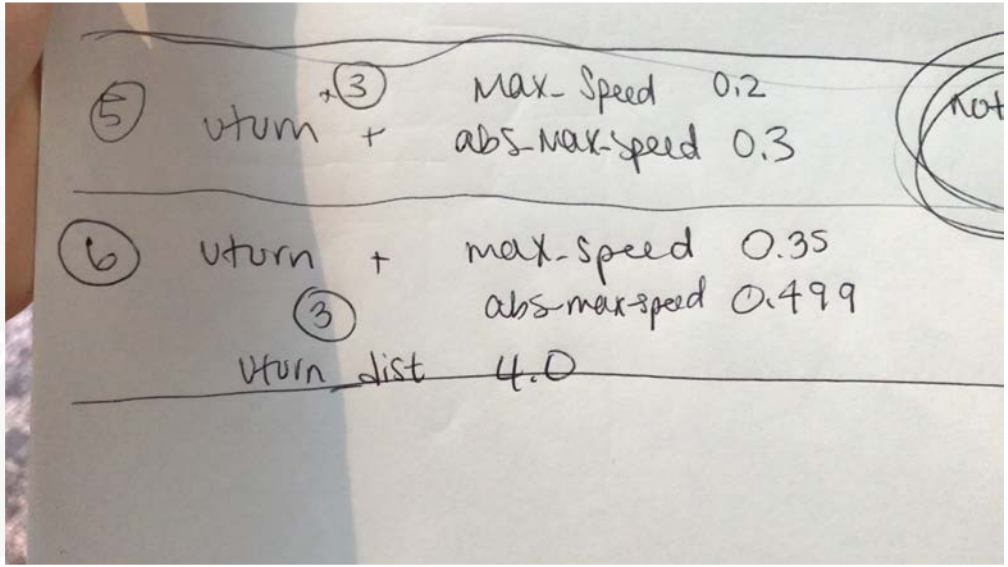
## Risk doing a U-turn here

The correct path is now behind the car, and the farthest path the car can "see" in the correct direction is shorter than the path in the wrong direction.



# Where the Algorithm Fails

## Risk doing a U-turn here



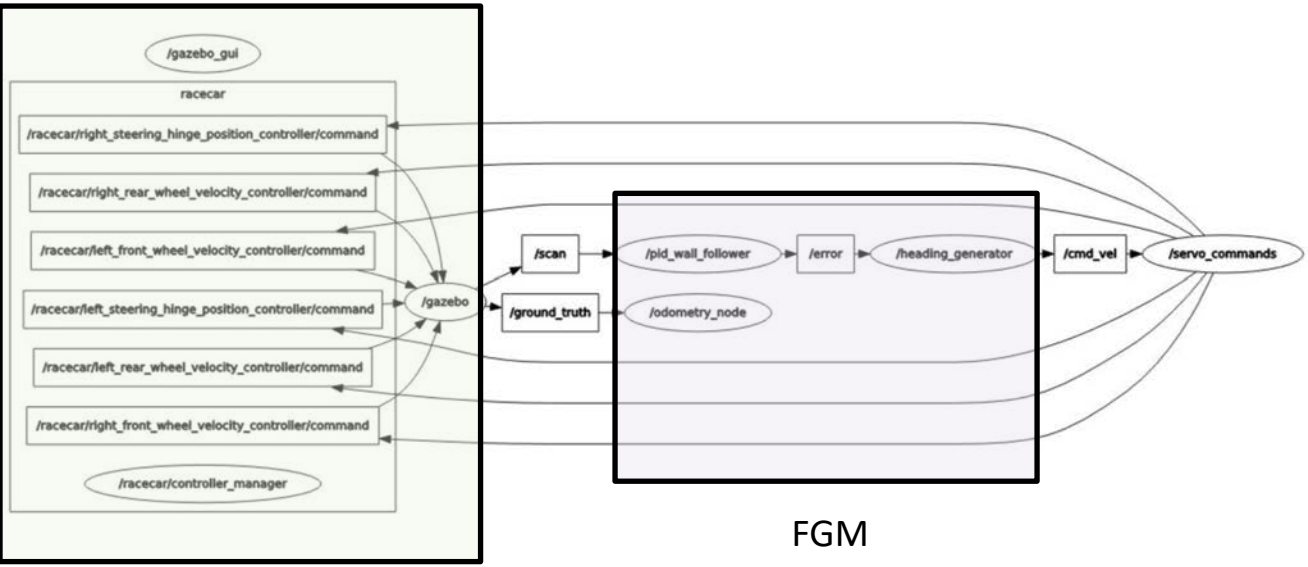
# What about longitudinal control

**Drive as fast as possible**

Pick a speed based solely on the *forward distance*

- If this distance is long enough, drive at the car's maximum speed
- If it's too short for the car to turn or avoid a collision, then stop.
- If the forward distance is anywhere in between the minimum safe distance and the safe "full-speed" distance, then simply scale the speed based on the distance (simple linear-scaling approach)

# Follow the Gap navigation and planning on the F1/10 Car




Simulator

FGM