Carnegie Mellon
ROBOTICS CLUB

Pathfinder Lab

Spring 2026

What Is Autonomous Sensing?

Autonomous sensing is when a robot collects information about its environment without human guidance, interprets it, and uses it to make decisions.

Robots must answer:

- Where am I?
- What is around me?
- Is there free space or an obstacle?
- What should I do next?

Real robots do this using:

- Cameras
- LIDAR
- Ultrasonic rangefinders

Your robot will do it using active ultrasonic sensing.

Carnegie Mellon ROBOTICS CLUB

Passive vs. Active Sensing

Passive sensing: robot observes the world as it is.

- Examples: camera, microphone, ambient light sensor.
- Pros: rich data
- Cons: complex processing, dependent on lighting/environment.

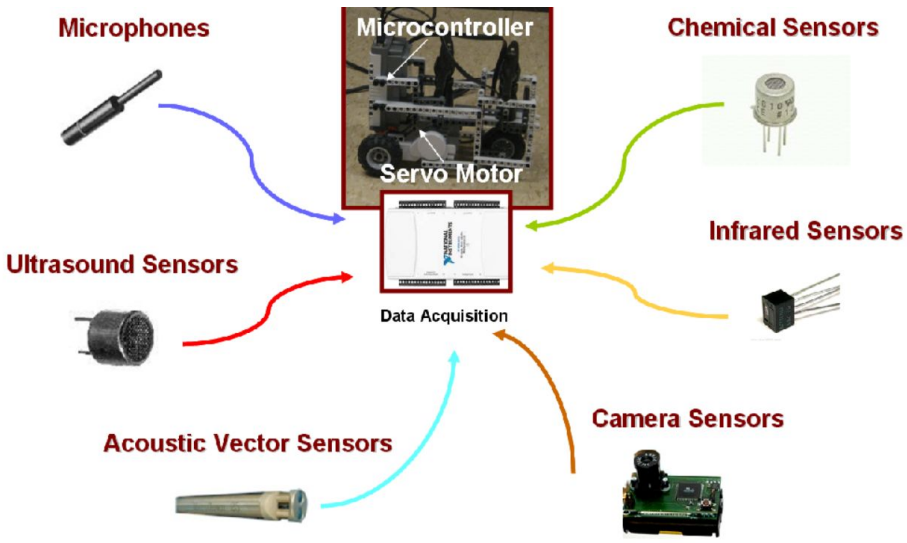
Active sensing: robot sends out energy and measures its reflection.

Examples:

- LIDAR (laser pulses)
- Radar
- Sonar / ultrasonic sensors
- Structured light

Pros: robust, simple

Cons: limited resolution



Carnegie Mellon ROBOTICS CLUB

Why Active Sensing?

- Cheap and easy to mount
- Works on reflective + non-reflective objects
- Only needs basic math
- Easy to visualize
- Similar in principle to LiDAR

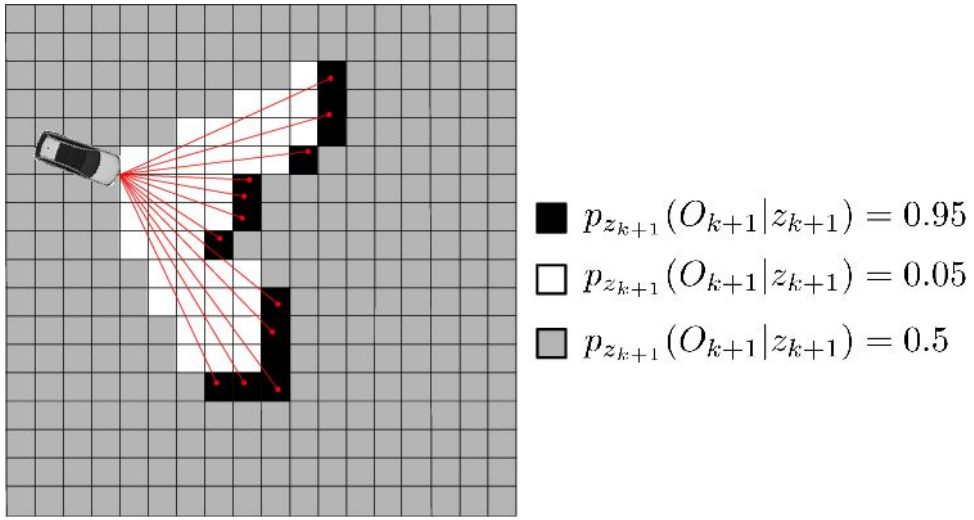
Carnegie Mellon ROBOTICS CLUB

From Sensing to Perception

Raw sensor data is not enough.

Robots must convert readings into useful structure:

1. Raw distances (“30 cm at 15°”)
2. Cartesian points (“Obstacle at x=29 cm, y=7 cm”)
3. Occupancy grid (“Cell (3,0) is occupied”)
4. World interpretation (“Left corridor has an obstacle halfway down”)



Carnegie Mellon ROBOTICS CLUB

What Is an Occupancy Grid?

An occupancy grid is a simple 2D map where space is divided into square cells:

- 0 = free
- 1 = occupied
- ? = unknown

This idea was introduced in the 1980s and is still used today in:

- Self-driving cars
- Drone mapping
- Warehouse robots
- Search & rescue robots
- SLAM systems

In this project, your grid is small (about 12×8 cells), but the concept is identical to what real robots use.

How Robots Choose Actions

Once a robot has perceived its surroundings, it must decide what to do next. This is called planning.

Levels of planning:

1. Reactive control
    a. “If something is close, slow down.”
2. Local planning
    a. “Go left or right around this obstacle.”
3. Global planning
    a. “Find the shortest path through a map.”

In this project, you’ll focus on local planning:

Decide which corridor (left or right) provides a clearer, safer path.

The Perception–Planning–Action Loop

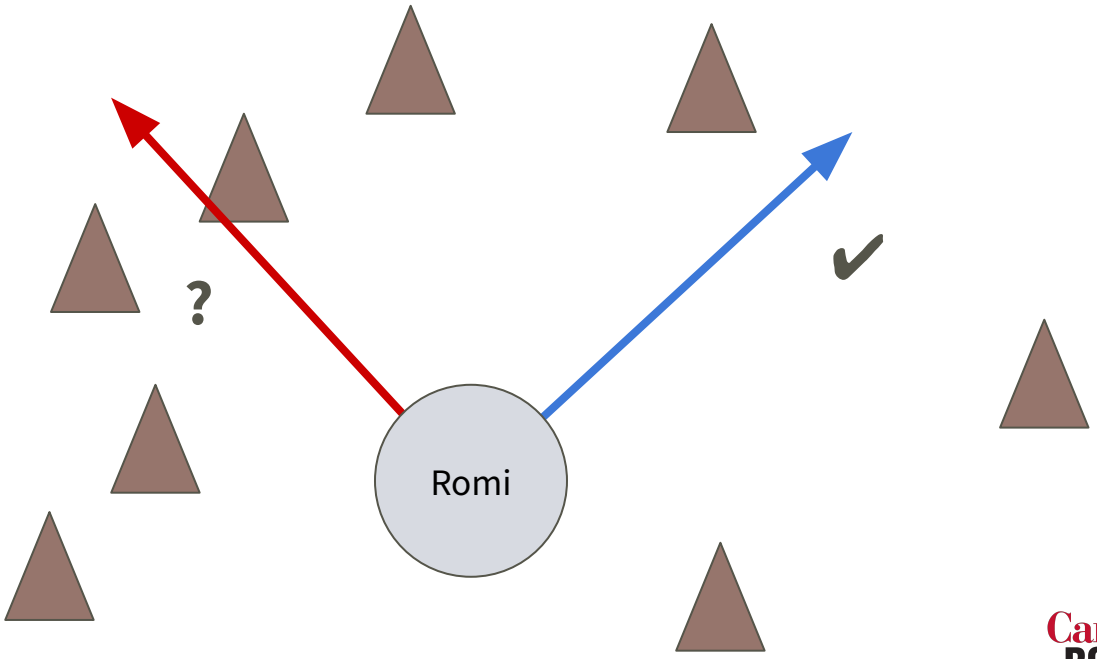
Real autonomous robots repeat this cycle: Sense → Map → Plan → Act → (repeat)

In your project, the loop looks like:

- Sense
  - Sweep the ultrasonic sensor
- Map
  - Convert distances to a 2D grid
- Plan
  - Score left vs. right corridor
  - Choose the best corridor
- Act
  - Turn into the corridor
  - Drive safely to the goal

Your Task

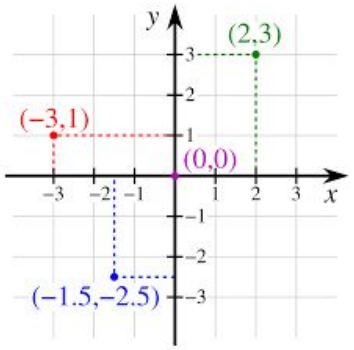
- Your robot will be placed in an unknown environment with obstacles
- You robot must autonomously map the scene
- Then, it must drive in the direction with fewer obstructions
  - Either left or right



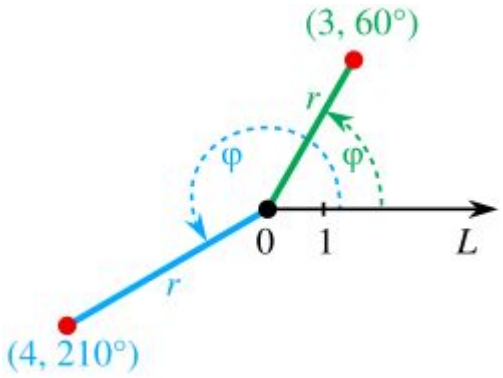
Carnegie Mellon ROBOTICS CLUB

Coordinate Systems

Some common coordinate systems used in robotics include cartesian coordinates and polar coordinates.



- Cartesian coordinates: (x, y)
  - x, y are distances from perpendicular axes
- Polar coordinates: (r, θ)
  - r is the distance from the origin, θ is the angle a ray to the point would make with the positive x axis



You will be working with both systems in pathfinder lab!

Carnegie Mellon ROBOTICS CLUB

Coordinate Systems Cont.

Recall the following formulas for converting from polar coordinates to cartesian coordinates:

Convert Polar to Cartesian

$$x = r\cos\theta$$

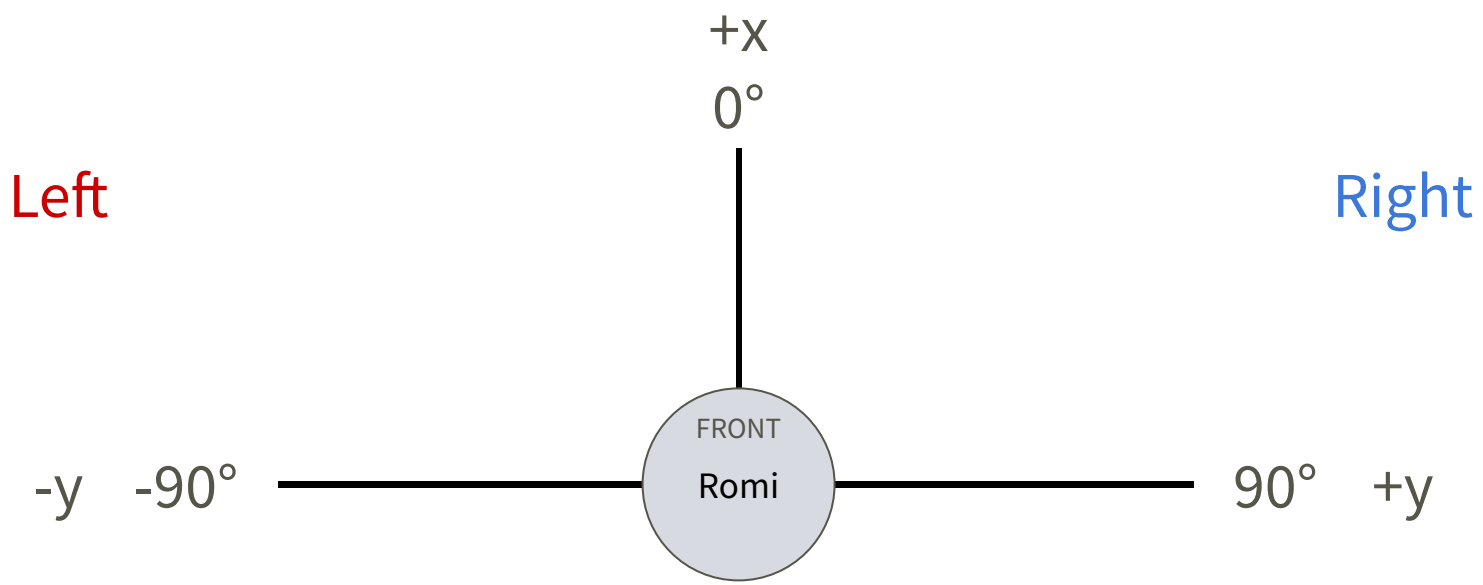
$$y = r\sin\theta$$

You will be converting your distance and angle into x,y coordinates to input into the occupancy grid!

Carnegie Mellon ROBOTICS CLUB

The Occupancy Grid

- In code, represented as a 2-dimensional array
  - The positive x-axis extends in front of the robot
  - The y-axis extends to the left and right
- Rotating left is a negative angle, rotating right is positive



Carnegie Mellon ROBOTICS CLUB

Occupancy Grid Cont.

Carnegie Mellon ROBOTICS CLUB

Code Runthrough

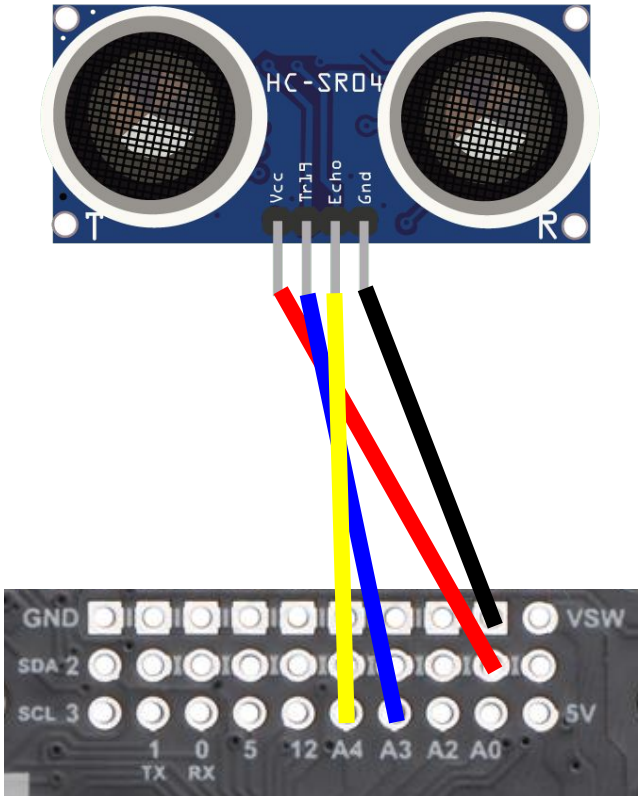
1. Sweep the ultrasonic sensor from -90° to +90°
   a. At each angle, read the distance from the sensor using the provided function
   b. Convert your (distance, angle) to (x, y)
   c. Mark the corresponding cell in the occupancy grid
2. Choose which direction to drive in
   a. Count up the number of occupied cells for left and right sides
   b. The side with fewer occupied cells is the direction you drive in!

Notes & Hints

- You may want to take multiple readings at each angle and use the average
  - Make sure to exclude any outliers! (distance < 5cm or > 150cm)
- The `grid` variable is your occupancy grid
  - We have set DEPTH = 8 and WIDTH = 12
  - Since the robot starts at WIDTH / 2, you will need to shift your coordinates by this amount
  - Each cell is a 10cm x 10cm block
  - The left side contains all points where y < WIDTH / 2, right side contains all points where y > WIDTH / 2
- Use the `Serial.println()` function while the robot connected to your computer to print out values!
- Use the `sin()` and `cos()` functions in Arduino
  - Remember to convert to radians

Carnegie Mellon ROBOTICS CLUB

Setup



Carnegie Mellon ROBOTICS CLUB