

# MATLAB-ROS-Gazebo Simulation Platform

**Lili Ma**

Assistant Professor

Dept. of Computer Engineering Technology (CET)

CUNY-New York City College of Technology

[LMa@citytech.cuny.edu](mailto:LMa@citytech.cuny.edu)

EdgeRobot Research Seminar

April 4<sup>th</sup>, 2023

# ROS and Gazebo

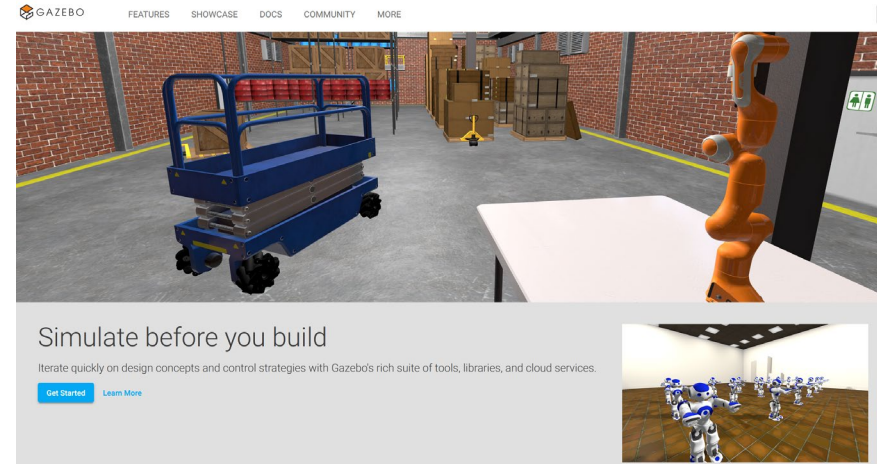
- **ROS:** Robot Operating System
- <https://www.ros.org/>

## ROS - Robot Operating System

The Robot Operating System (ROS) is a set of software libraries and tools that help you build robot applications. From drivers to state-of-the-art algorithms, and with powerful developer tools, ROS has what you need for your next robotics project. And it's all open source.

- Open source
- State-of-art algorithms
- Compatible to simulators & physical robots

- Gazebo Simulator:
- <https://gazebo.org/home>



- Complete toolbox of development libraries
- Realistic Environments
- Sensors of high fidelity

# MATLAB-ROS-Gazebo

## Control of a simulated robot in the Gazebo simulator on MATLAB utilizing MATLAB's ROS toolbox

The IP address of the computer running MATLAB. Can be obtained by typing ipconfig in the host computer's command window:

```
Command Prompt
Microsoft Windows [Version 10.0.22621.963]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Lilim>ipconfig
```

The IP address shown on the virtual machine:

```
user@ubuntu: ~
user@ubuntu:~$ ifconfig
```

If instead an IPv6 IP address is displayed, use the following commands on the virtual machine's terminal to disable the IPv6 IP address:

```
sudo systemctl -w net.ipv6.conf.all.disable_ipv6=1
sudo systemctl -w net.ipv6.conf.default.disable_ipv6=1
```

List of available ROS topics

The MATLAB interface displays a list of ROS topics on the left, including `/camera/parameter_descriptions`, `/camera/parameter_updates`, `/camera/rgb/camera_info`, `/camera/rgb/image_raw`, `/camera/rgb/image_raw/compressed`, `/clock`, `/cmd_vel`, `/gazebo/link_states`, `/gazebo/model_states`, `/gazebo/parameter_descriptions`, `/gazebo/parameter_updates`, `/gazebo/performance_metrics`, `/gazebo/set_link_state`, `/gazebo/set_model_state`, `/imu`, `/joint_states`, `/odom`, `/rosout`, `/rosout_agg`, `/scan`, `/tf`, `/tf_static`, `/turtlebot_move/cancel`, `/turtlebot_move/feedback`, `/turtlebot_move/goal`, `/turtlebot_move/result`, `/turtlebot_move/status`, `/turtlebot_move`, and `/turtlebot_move/status`. A red box highlights the `/turtlebot_move` topic. In the center, a script window shows the following code: `% laser scan: https://www.mathworks.com/help/ros/ug/using-ros-bridge.html`, `% need to start action server on VM: ~/start-turtlebot-move`, `clear; close all; close all;`, `rosshutdown;`, `rosinit('192.168.1.158', 'NodeHost', '192.168.1.154');`, and `rostopic list`. A red box highlights the IP addresses `192.168.1.158` and `192.168.1.154`. Below the script, a 'Figure 1' window shows a 360-degree laser scan plot with axes X and Y ranging from -4 to 4. A red dashed arrow points from the IP addresses in the script to the VM's IP address in the screenshot below.

ROS Action List

Image captured by the robot's onboard camera

360° laser scan data

MATLAB

The Virtual Machine (Gazebo Simulator) interface shows a desktop environment with various icons for Gazebo simulations like 'Gazebo VEX Clawbot', 'Gazebo Warehouse Robot', 'Gazebo Pioneer 3-DX', 'Gazebo ROS2 Maze', 'RoboCup Manipulation 1', 'ROS Foxy Terminal', 'Gazebo RecyclingWorld', 'Gazebo PX4 SITL', 'RoboCup Manipulation 2', 'ROS Noetic Core', 'Ubuntu and ROS licenses', 'Gazebo Office', 'ROS Noetic Terminal', and 'Gazebo Sign Follower ROS'. A 'Gazebo' window is open, showing a 3D simulation of a robot in a square environment. A blue arrow points from the VM's IP address in the screenshot to the MATLAB script above.

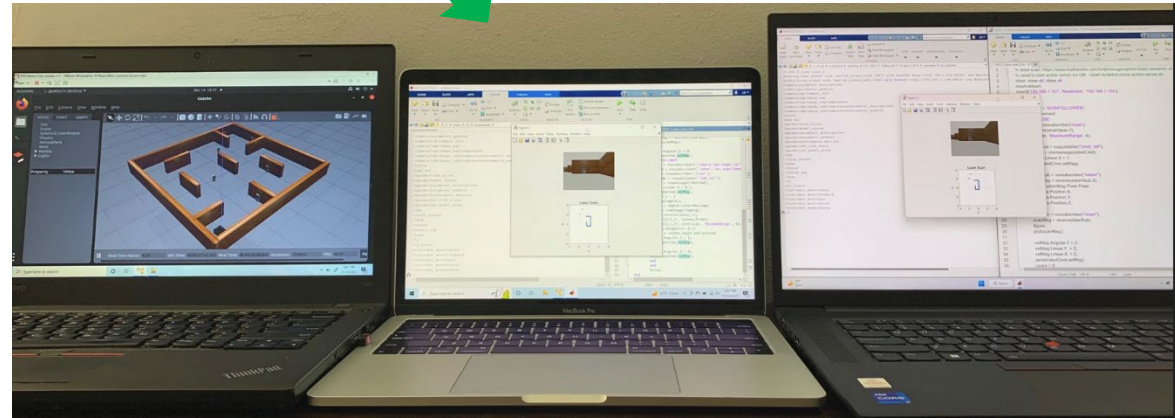
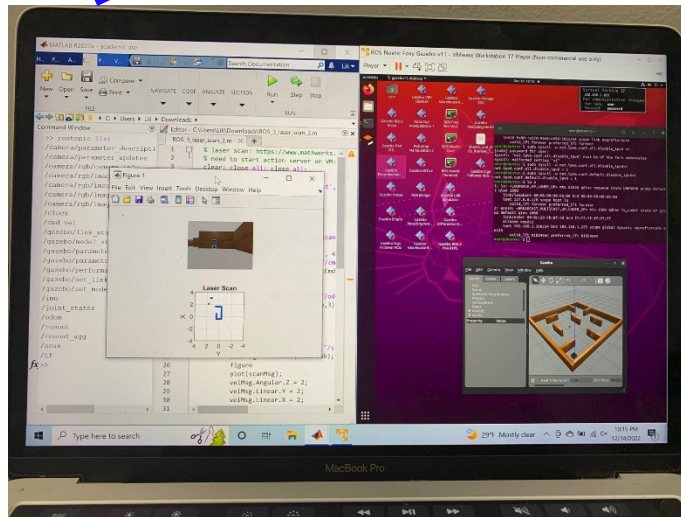
Virtual Machine (Gazebo Simulator)

# Setting Up the Platform

- <https://www.mathworks.com/help/ros/ug/get-started-with-gazebo-and-a-simulated-turtlebot.html>
- <https://www.mathworks.com/support/product/robotics/ros2-vm-installation-instructions-v8.html>
- Requires:
  - **MATLAB** with **ROS** toolbox
  - **VMware Player** that plays a **virtual machine**

# Setting Up the Platform

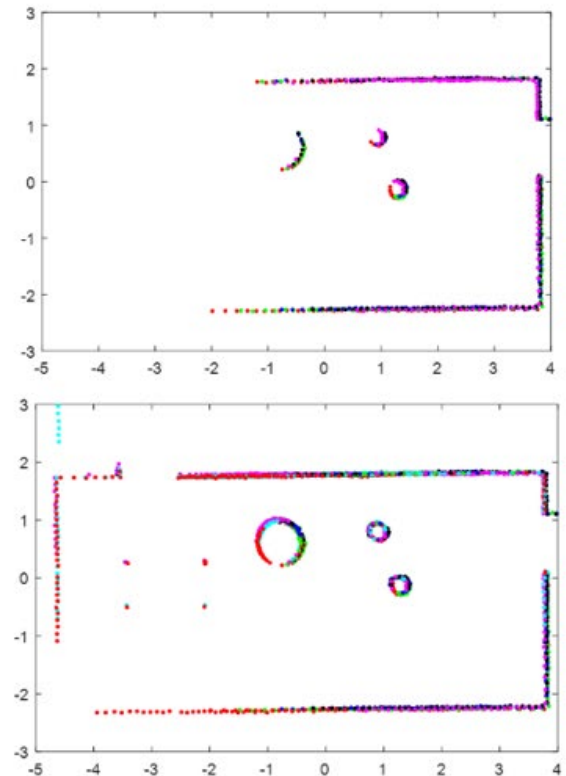
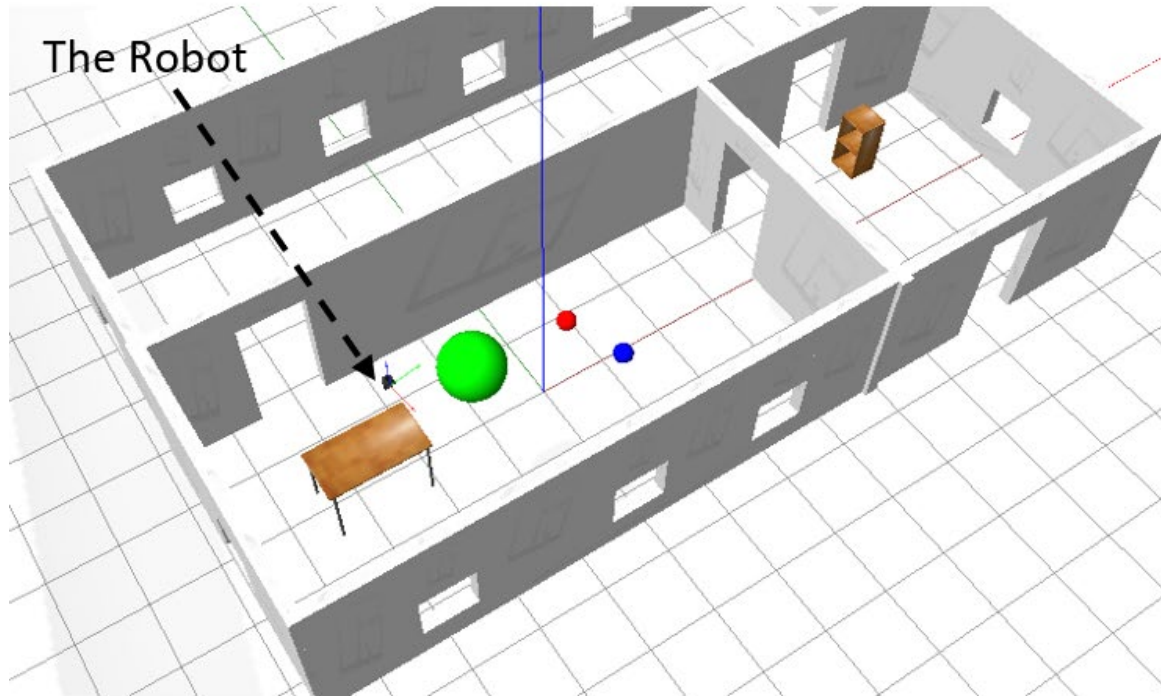
- So far, I have setup on **Windows**
  - Both MATLAB and Gazebo on one computer running Windows
  - Both MATLAB and Gazebo on one Mac (Windows is installed via Bootcamp)
  - Gazebo on one computer with two other computers running MATLAB



# #1: Receiving Laser Scan Data

The **Office** environment:

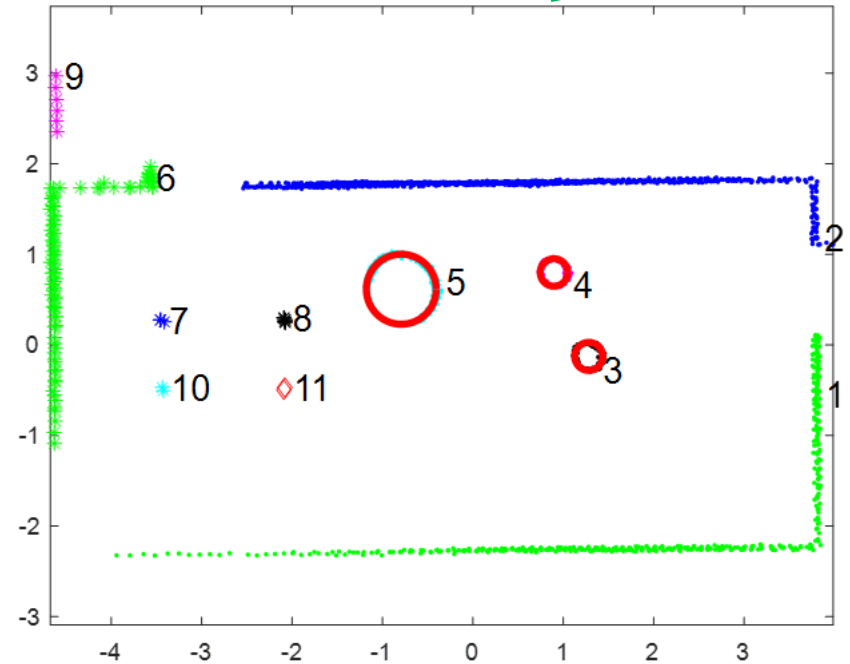
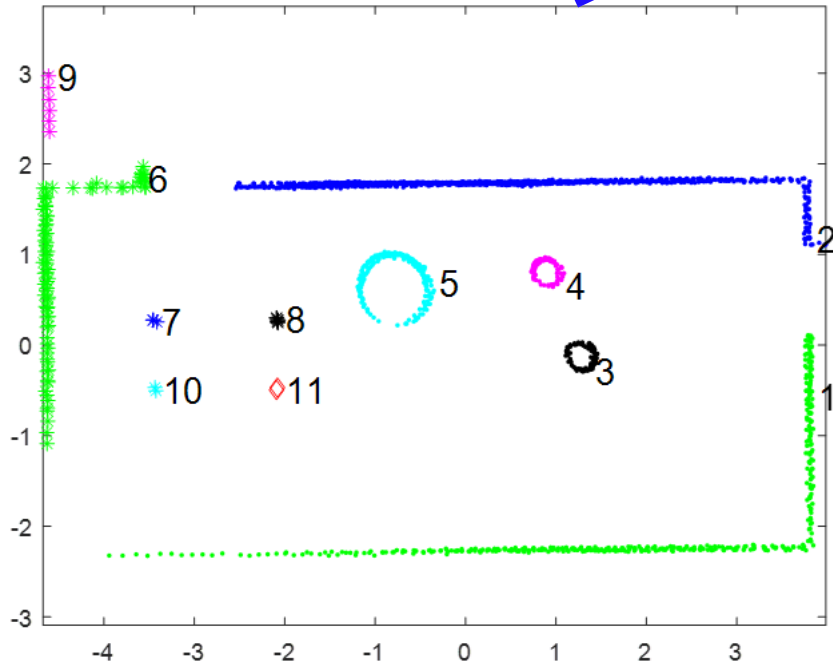
1. The robot is commanded to move around.
2. The collected laser scan data are cumulated together, yielding a better representation of the robot's environment.





# #1: Receiving Laser Scan Data

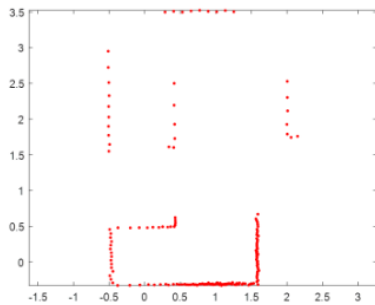
- Allowing data **clustering** and model-based **fitting**



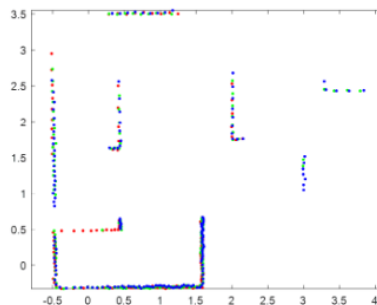
# #2: Path Planning

The **Gazebo Sign Follower ROS** environment:

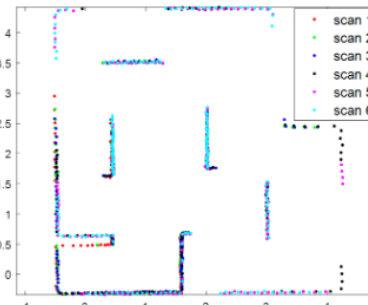
1. The robot moves around, collecting laser scans.
2. Implementation of home-made path planning algorithm
3. Command the robot to reach the goal location.



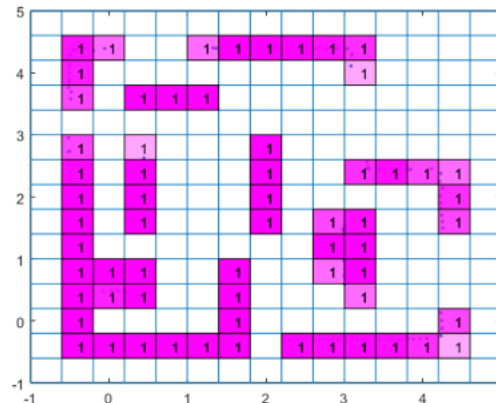
(a) 1 Scan



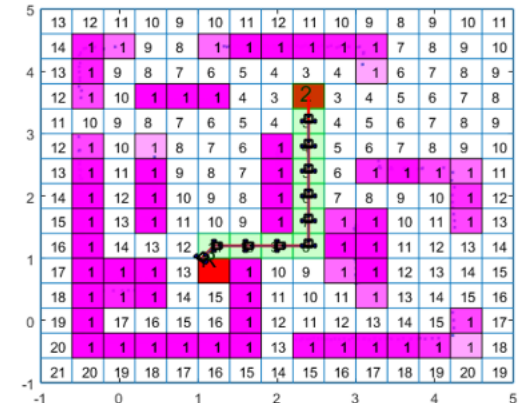
(b) 3 Scans



(c) 6 Scans



(a) Setting Up Wavefront Map



(b) Wavefront Propagation



# #2: Path Planning

EDITOR - C:\Lib\4\_CityTech\teaching\CET\_4952\Online\_HD\Project\_3\_ROS\exampl... - □ ×

EDITOR Figure 1 File Edit View Insert Tools Desktop Window Help

Open Save FILE

ROS\_3\_laser\_scan\_2.m

```
% [cli  
% waitFo  
getOdom;  
initial_  
  
count =  
for k =  
    star  
    endi  
    v =  
    dist  
    fina  
AOR  
  
% ro  
goal  
goal
```

5	13	12	11	10	9	10	11	12	11	10	9	8	9	10	11
4	14	1	1	9	8	1	1	1	1	1	1	7	8	9	10
3	12	1	10	1	1	1	4	3	2	3	4	5	6	7	8
2	11	10	9	8	7	6	5	4	3	4	5	6	7	8	9
1	12	1	10	1	8	7	6	1	3	5	6	7	8	9	10
0	13	1	11	1	9	8	7	1	3	6	1	1	1	1	11
-1	14	1	12	1	10	9	8	1	3	7	8	9	10	1	12
-2	15	1	13	1	11	10	9	1	3	1	1	10	11	1	13
-3	16	1	14	13	12	1	1	1	1	1	1	11	12	13	14
-4	17	1	1	1	13	1	1	10	9	1	1	1	12	13	14
-5	18	1	1	1	14	15	1	11	10	11	1	13	14	15	16
-6	19	1	17	16	15	16	1	12	11	12	13	14	15	1	17
-7	20	1	1	1	1	1	1	13	1	1	1	1	1	1	18
-8	21	20	19	18	17	16	15	14	15	16	17	18	19	20	19
-9	-1	0	1	2	3	4	5								

TF-8 CR

```
TurnDistance : 0.6303907632827759  
ForwardDistance : 0  
Goal active  
Final state succeeded with result:  
TurnDistance : 0  
ForwardDistance : 0.2725736200809479
```

ROBO VEXR SOLID Busy

only)

Dec 6 20:34

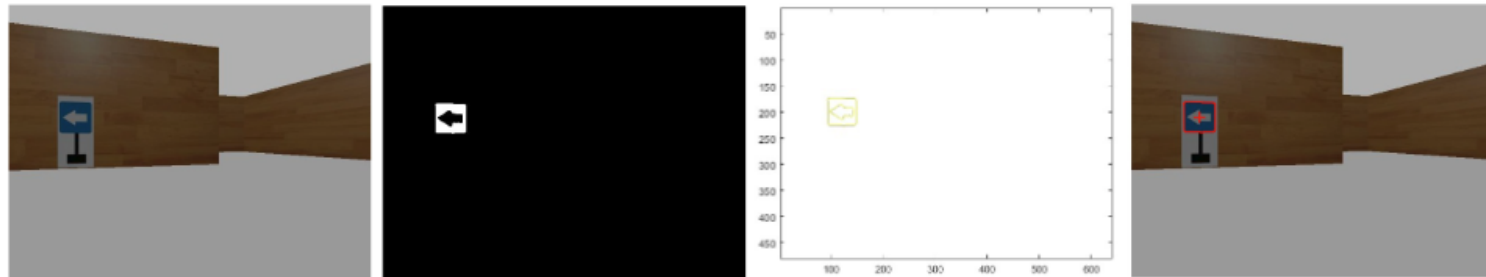
Gazebo

Real Time Factor: 0.31 Sim Time: 00:00:26:53.046 Real Time: 00:00:00

# #3: Vision-Based Control

The **Gazebo Sign Follower ROS** environment:

1. The robot captures images using its onboard camera.



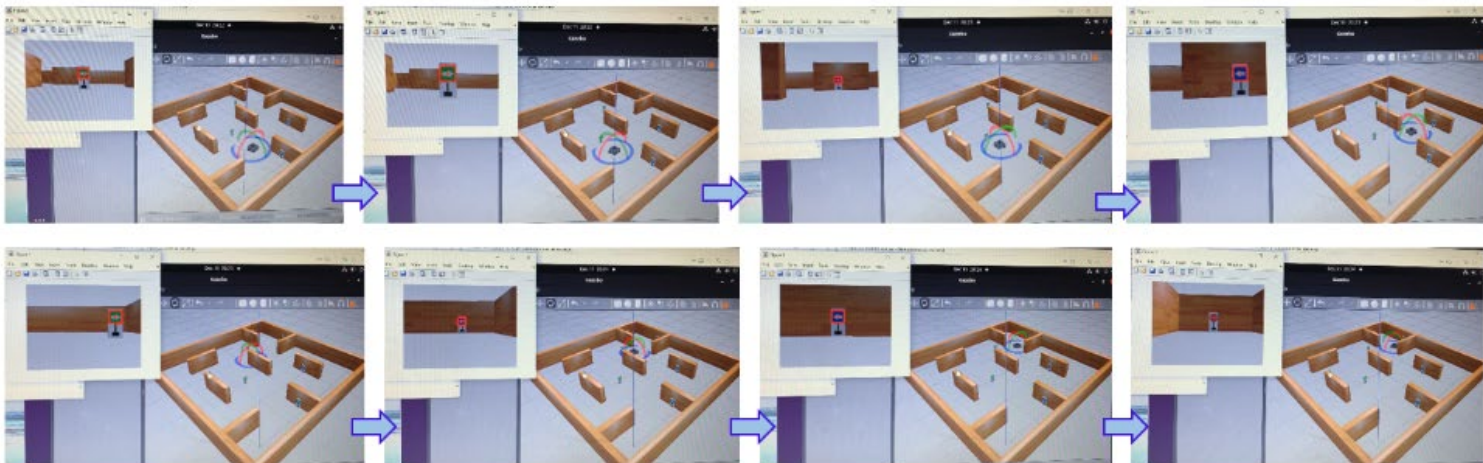
(a) Original Image

(b) Thresholding Hue

(c) Detected Contours

(d) Center

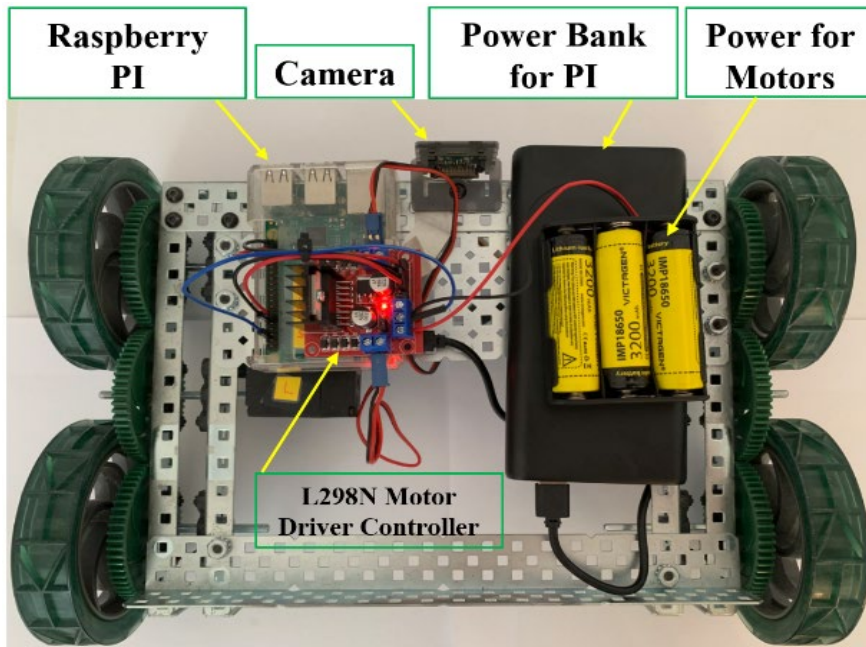
2. The robot is controlled to turn left/right according to the sign





# Physical Robots

- We have several VEX robots and one TurtleBot3.
- We are now working on controlling these physical robots via ROS.



**TURTLEBOT3**  
Waffle Pi

