Lab #0: TurtleBot3 and Its Simple Control

Introduction

In this experiment, you will explore the capabilities of wheeled robots by controlling a **TurtleBot3 Waffle Pi**. You will also develop an understanding of the basic sensor data, coming from **camera**, **LiDAR** and **wheels encoders**.

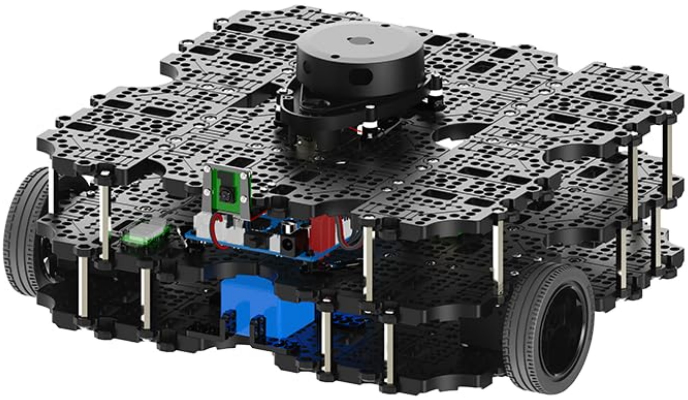

Figure 1: A TurtleBot3 Waffle Pi wheeled robot.

Getting Started

1. Connecting the Power to the Robot

1.1 Power via Adapter (SMPS)

While programming the **TurtleBot3 Waffle Pi**, it is recommended to power it through the AC/DC adapter (SMPS). Connect the adapter to the **OpenCR** board as shown in Figure 2.

1.2 Power via Battery Pack

When the robot is moving untethered, use the **Li-Po battery pack** supplied with TurtleBot3. Insert the battery cable into the **OpenCR power port** as shown in Figure 3.

2. Powering On the Robot

Locate the power switch on the **OpenCR Board** as shown in Figure 4 and switch it on.

Once the lights of the **OpenCR Board** and the **Raspberry Pi 4** start flickering, and the **LiDAR** starts spinning, wait for at least 30 seconds for the robot to boot.

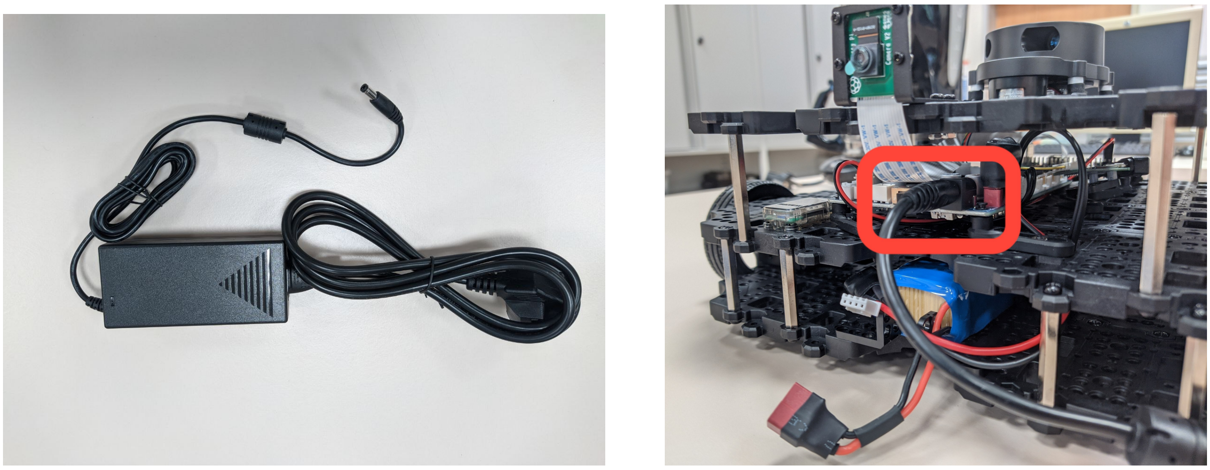

Figure 2: Connecting the SMPS to the OpenCR for tethered power.

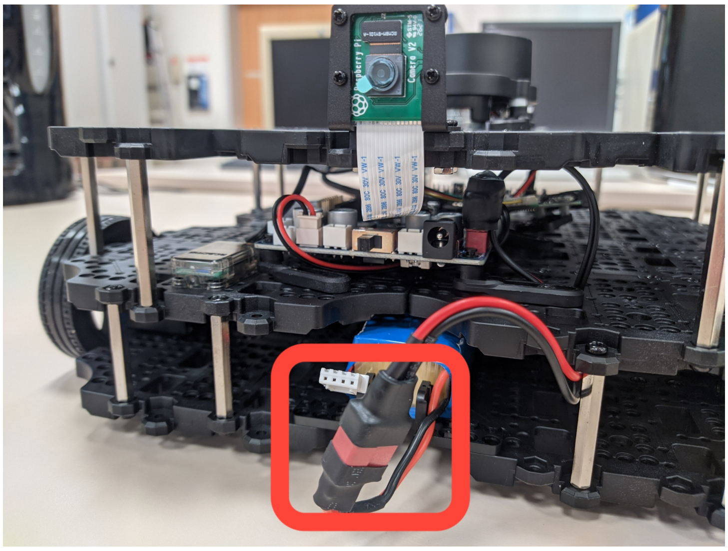

Figure 3: Connecting the battery pack for untethered power.

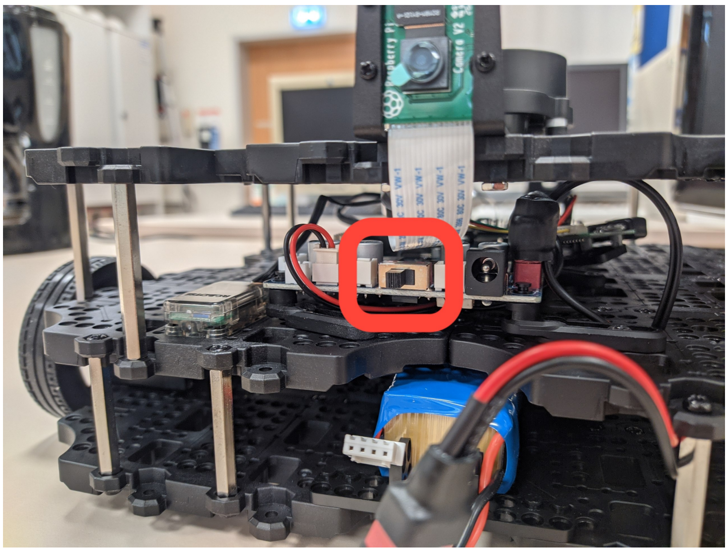

Figure 4: The power switch on the OpenCR Board.

3. Connecting the PC to the Robot

3.1 Wi-Fi Connection

For real-time autonomous operation, Wi-Fi is the preferred connection method. It allows the robot to move freely without cables, making it ideal for navigation and mapping tasks.

Wi-Fi Setup Steps:

- Connect your PC/Laptop to the **TurtleBot3 Waffle Pi**'s Wi-Fi hotspot:

  ```
  Name: TBX
  Password: 12345678
  ```

  (Wi-Fi hotspot information is written on the **TurtleBot3's ID Card**)

  Follow the steps shown in Figure 5.

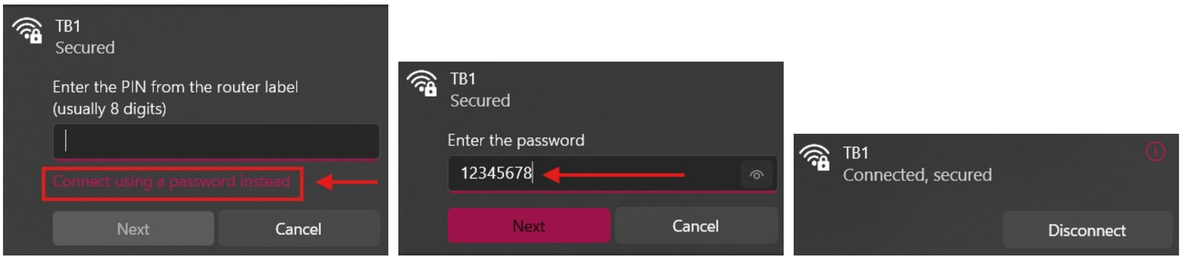

Figure 5: Connecting to the TurtleBot3 Wi-Fi hotspot.

- Open a **Command Prompt (cmd)** or **PowerShell** window on your PC by searching it from the Start menu.

- Test connectivity to the robot's Raspberry Pi by typing:

  ```
  ping 192.168.XX.1
  ```

  (Wi-Fi IP information is written on the **TurtleBot3's ID Card**)

- Open an SSH connection to the robot by typing:

  ```
  ssh me425-1@192.168.XX.1
  ```

  (Login Username and Wi-Fi IP information are written on the **TurtleBot3's ID Card**)

  You will then be prompted to enter the password:

  ```
  me425-1@192.168.XX.1's password:
  ```

  (Password is `1234` for all the **TurtleBot3**s.)

- Once connected successfully, you should see a terminal similar to Figure 6.

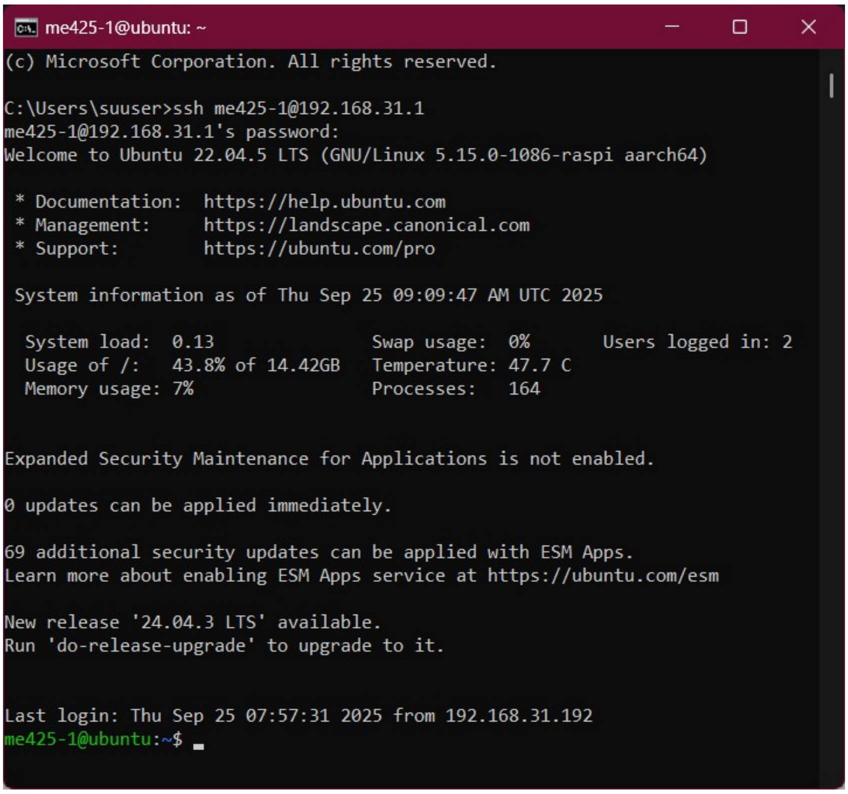

Figure 6: Successful SSH connection to the TurtleBot3 over Wi-Fi.

3.2 Ethernet Connection

Note

For now, you will not need to connect via Ethernet if you are already connected through Wi-Fi. Check appendix to see the full setup steps in case of Wi-Fi connection loss.

4. ROS Nodes Bring-up on the Robot

Once your connected robustly to the **TurtleBot3** via Wi-Fi or Ethernet, you can now control it via your ssh connection from the **Command Prompt (cmd)** or **PowerShell** on your PC. By default, no ROS nodes are running. You must launch the bring-up process to start publishing the required sensor data.

Bring-up Setup Steps:

- Connect your PC to the TurtleBot3 (either via Wi-Fi or Ethernet tethering) and make sure that you are seeing a terminal similar to Figure 6 or Figure 9.
- In the same **Command Prompt (cmd)** or **PowerShell** window you already have on your PC, launch the **TurtleBot3** bring-up node by typing:

```
bringup
```

- Once the bring-up is successful, you should see a terminal similar to Figure 7.

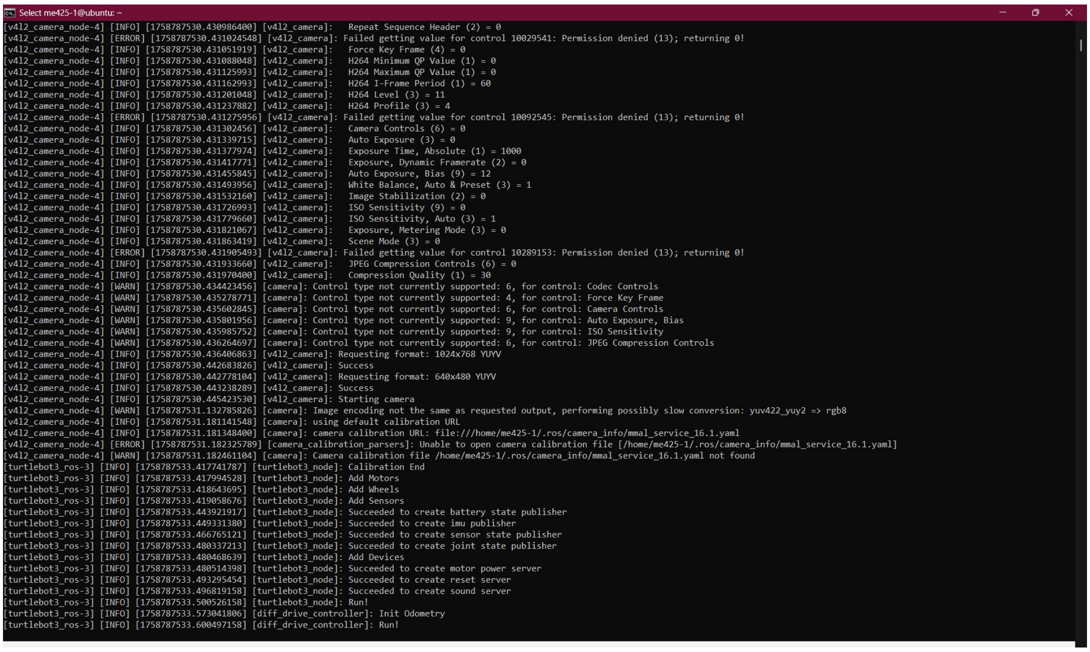

Figure 7: Successful launch of the bring-up node.

- Leave this terminal window open and running. You are now ready to continue with **MATLAB**!

5. MATLAB Setup for TurtleBot3

Before running any MATLAB code, configure your ROS 2 environment to match the **TurtleBot3**'s Raspberry Pi.

MATLAB Preparation Steps:

- Add this block at the beginning of any MATLAB script:

```
%% --- MATLAB  TurtleBot3 ROS 2 setup ---
clear node; % clear previous node handle if it exists
setenv('ROS_DOMAIN_ID','XX'); % Set to your robots ROS_DOMAIN_ID
setenv('ROS_LOCALHOST_ONLY','0'); % Must be 0 for multi-host
    communication
setenv('RMW_IMPLEMENTATION','rmw_fastrtps_cpp'); % Middleware

% Create a unique node name for this MATLAB session
node = ros2node('ANY_NAME_YOU_WANT');
```

  (ROS Domain ID is written on the **TurtleBot3's ID Card**)

- *Optional:* Check your environment and node:

```
% --- Quick sanity check: display the node name and current environment
    ---
disp("ROS 2 up. Node: " + node.Name);
disp("ROS_DOMAIN_ID = " + string(getenv('ROS_DOMAIN_ID')));
disp("RMW_IMPLEMENTATION = " + string(getenv('RMW_IMPLEMENTATION')));
```

- *Optional:* Verify ROS traffic from the robot:

```
% Create a lightweight /scan subscriber to confirm traffic:
subScan = ros2subscriber(node,"/scan","sensor_msgs/LaserScan", ...
    "Reliability","besteffort","Durability","volatile","Depth",10);

msg = receive(subScan,5); % Set waiting time in seconds

if isempty(msg)
    disp("Waiting for /scan... (Start the bringup on the robot from the
        cmd!)")
end
```

Important Note

Never power off the **TurtleBot3** using the power switch located on the **OpenCR Board** shown in Figure 4! (this will cause the system to be corrupted)
The proper way is by typing:

```
sudo poweroff
```

into the already running cmd terminal shown in Figure 6 or Figure 9.
If your ROS nodes are up and running like in Figure 7, you will first need to stop them by pressing **CTRL + C** inside the cmd terminal, then you can power off the system using:

```
sudo poweroff
```

once the LiDAR stops spinning, this indicates that the robot is safely shut down, so now you can turn off the switch shown in Figure 4 and disconnect the battery or SMPS.

The Experiment

Goals

- Use onboard sensors: **LiDAR**, **camera**, and **wheel encoders**.
- Visualize sensor streams in MATLAB.
- Command the robot to execute **forward**, **backward**, and **circular** motions at different speeds.
- Record and plot **encoder** data for each maneuver.

Checklist Before You Start

- You are connected to the robot via **Wi-Fi or Ethernet** (only one at a time).
- You can SSH into the robot and have launched bringup on the robot.
- MATLAB ROS 2 environment is configured.

Part A — Sensor Bringup & Visualization

Implement `cameraLidar.m` to:

1. Subscribe to the LiDAR topic (`/scan`, `sensor_msgs/LaserScan`).
2. Subscribe to the camera topic (`/image_raw/compressed`, `sensor_msgs/CompressedImage`).
3. Display a camera view and a simple LiDAR visualization (e.g., polar plot or range vs. index).
4. Take a screen-shot of the output (image and LiDAR plot) at specific instance.

Part B — Simple Open Loop Motion Control & Encoders

Implement `controlledMotion.m` to:

1. Publish velocity commands on `/cmd_vel` (`geometry_msgs/Twist`) to move:
   - Forward for 3–5 s,
   - Backward for 3–5 s,
   - Circle: send nonzero angular velocity for 8–12 s.
2. Subscribe to `/joint_states` (`sensor_msgs/JointState`) and log left/right wheel positions (ticks/ radians).
3. Plot both wheel encoder traces vs. time for each maneuver and comment on symmetry/offsets.

Safety: Lift the robot wheels or keep a clear area when testing `/cmd_vel`.

Things to do:

- Submit your report **via SUCourse** until the report submission deadline.

Post-Lab Report Deadline: 15 October 2025, 23:55 via **SUCourse**

- Your report must include:
  - **Introduction**
  - **Procedure**
  - **Results**
  - **Conclusion**
  - **Discussion**
  - **Appendix**
  - Provide your **MATLAB** codes in Appendix section appropriately.

Answer the following questions in the Discussion section of your Post-lab report:

- Plot the data that you obtained from LiDAR. How do you think these data are related to the surrounding environment?
- Take a screen shot of your camera output. Comment on how you think camera is used in autonomous mobile robotics.
- Can your robot move straight and rotate properly? Plot the encoder values of both motors to clarify your answer.

Appendix: Ethernet Connection (if needed)

Ethernet Setup Steps:

For a more stable and faster connection, connect the robot directly to your PC/Laptop using an Ethernet cable.

- Connect one end of the Ethernet cable to your PC/Laptop and the other end to the **Raspberry Pi 4** on the TurtleBot3.
- Set up your PC's Ethernet settings as shown in Figure 8.

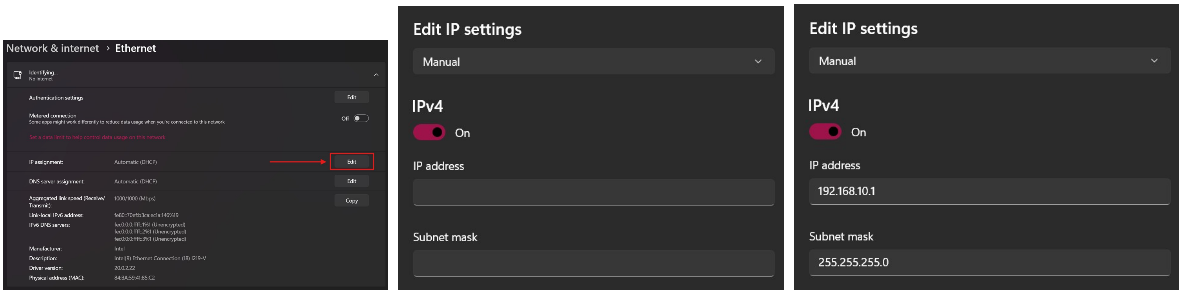

Figure 8: Setting up PC's Ethernet IP address

- Then your PC will automatically detect the connection. The robot will use the IP address:

  ```
  192.168.10.2
  ```

  (Same Ethernet IP for all the **TurtleBot3**s)

- Open a **Command Prompt (cmd)** or **PowerShell** window on your PC by searching it from the Start menu.
- Test connectivity to the robot's Raspberry Pi by typing:

  ```
  ping 192.168.10.2
  ```

  (Same Ethernet IP for all the **TurtleBot3**s)

- Open an SSH connection to the robot by typing:

  ```
  ssh me425-1@192.168.10.2
  ```

  (Same Login Username and Ethernet IP for all the **TurtleBot3**s)

  You will then be prompted to enter the password:

  ```
  me425-1@192.168.10.2's password:
  ```

  (Password is `1234` for all the **TurtleBot3**s.)

- Once connected successfully, you should see a terminal similar to Figure 9.

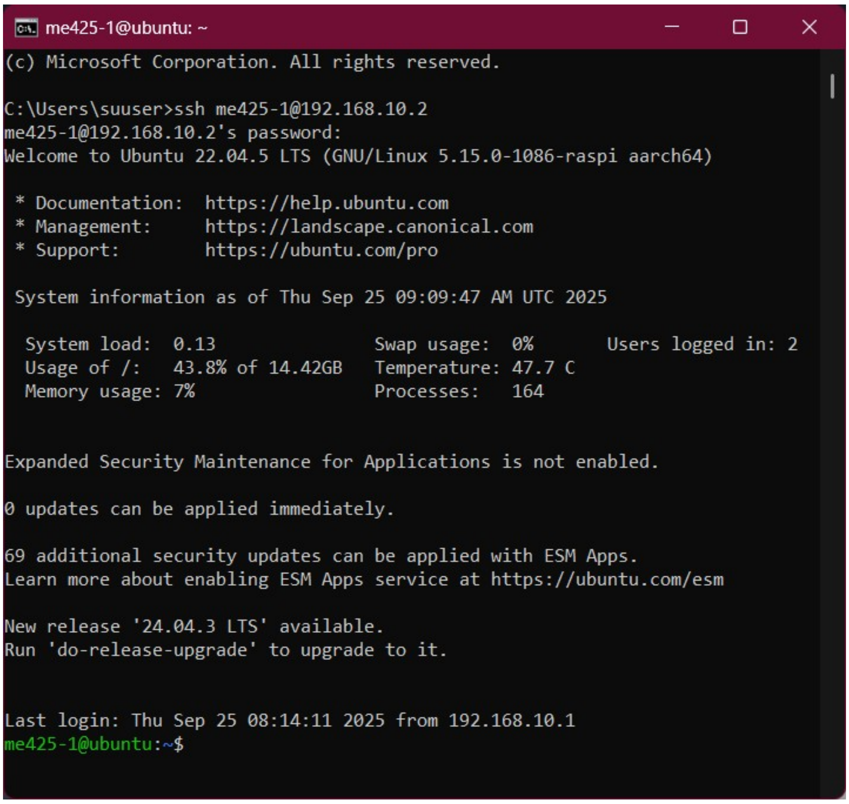

Figure 9: Successful SSH connection to the TurtleBot3 over Ethernet.

Important Note

Only use **one connection method at a time** (either Wi-Fi *or* Ethernet). Having both connected simultaneously to your PC will cause conflicts in MATLAB and lead to communication errors.