ME 392, Undergraduate Research
05-09-2025, Spring 2025
CHICAGO, IL, USA

HIWONDER MENTORPI A1 ACKERMAN CHASSIS AUTOMOMOUS VEHICLE (CONCENTRATION ON SLAM MAPPING)

Mubashir Farooqi
Robotics and Motion Laboratory
Dept. of Mechanical & Industrial Engineering at University of Illinois Chicago
Chicago, IL, USA
mfaro5@uic.edu

ABSTRACT

The given problem statement was to build a similar autonomous vehicle as RoboRacer(formerly known as F1Tenth) that follows a track and avoid collisions with obstacles in cast effective way. By proposing and developing a Hiwonder MentorPi A1 Ackerman Chassis vehicle, there was an 86% reduction in bill of materials cost while achieving similar functionalities as a RoboRacer vehicle.

SLAM mapping was the primary focus but Lane keeping was an essential first step for autonomous driving. Lane keeping was achievable since access to Virtual Network Computing (VNC) app was only needed. A map can be made with yellow tape which will represent lanes in which the robot will follow.

However, SLAM mapping was a more difficult process since communication between a virtual machine and VNC had to be made. A virtual machine was necessary for communication between a person's laptop and robot when SLAM mapping occurs. SLAM mapping maps an environment based on the robot's surroundings; Rviz is short for ROS visualization which is a 3D visualization tool used in ROS (Robot Operating System). Rviz will be used on a personal laptop or computer through a virtual machine. Proper communication must be made between the robot, and a virtual machine, or SLAM mapping will not function.

Much debugging had to occur for SLAM mapping to work since the Hiwonder documentation does not mention key details such as having a working virtual machine ready to use. Instead, they explain that you must import files from the Hiwonder robot onto your own personal virtual machine which can confuse users. Another crucial step is to ensure that the robot and virtual machine are connected to the same WIFI.

Once achieving the proper connections, SLAM mapping will function, and the robot will be able to map out any environment it's in which will be broadcasted onto Rviz, located on a virtual machine. The robot can be moved from a laptop's keyboard in which the environment will be mapped as it moves around. In addition, obstacles will be detected automatically.

There were some shortcomings which consisted of the Ackerman chassis not making sharp turns during Lane Keeping. In addition, obstacle avoidance and mapping out a larger environment can be explored further.

1. INTRODUCTION

Proposed and developed 1/10 scaled autonomous vehicle with an 86% reduction of the bill materials cost compared to the industry standard RoboRacer (formerly known as F1Tenth). The autonomous vehicle is a Hiwonder MentorPi A1 Ackerman Chassis powered by Raspberry Pi 5, compatible with ROS2, programmed in Python, and utilizes advanced AI capabilities such as SLAM mapping.

2. METHODS

First, assembling the autonomous vehicle consisted of many parts such as the lidar sensor, 3D depth camera, and Raspberry Pi. For reference, Hiwonder delivers a google drive folder with documentation and YouTube videos for a physical setup. [1]

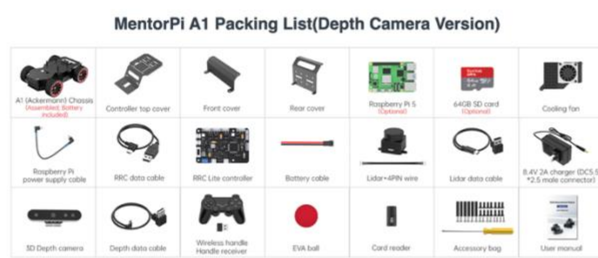

Figure 1. Parts List for Hiwonder Mentor Pi A1 Ackerman Chassis

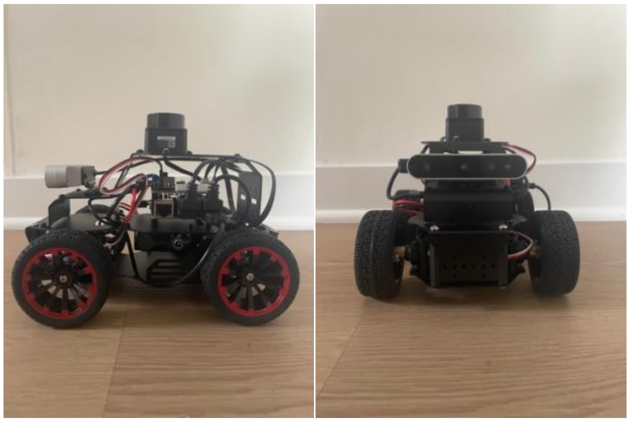

Figure 2. Front and Side View of Hiwonder Robot

Once assembled using the parts from **Figure 1**, the final product should appear as shown in **Figure 2**. The Hiwonder robot should now be able to power up and be connected from your phone or laptop using the hotspot generated. The hotspot starts with "HW" and can be found in nearby networks.

To verify functionalities, the user should download the WonderPi app which can be used to connect a cell phone to the robot. Once connected, lidar tracking, obstacle avoidance, and a remote control can be used with your cell phone.

Figure 3. Mode Selection in WonderPi App

When using the robot control in the WonderPi app, shown in **Figure 3**, the Ackerman chassis was not working, and the steering was disabled. The robot version configuration must be changed, and the Ackerman chassis option should be chosen. Since they're two versions of chassis, we must verify the robot version configuration.

To change the robot version configuration, we must access the raspberry pi software. There are two methods of connection, one being through downloading Virtual Network Computing (VNC), or a micro-USB to HDMI can be connected from the Raspberry Pi 5 to a monitor. For easier accessibility, downloading the VNC software and connecting a laptop to the Hiwonder hotspot is recommended.

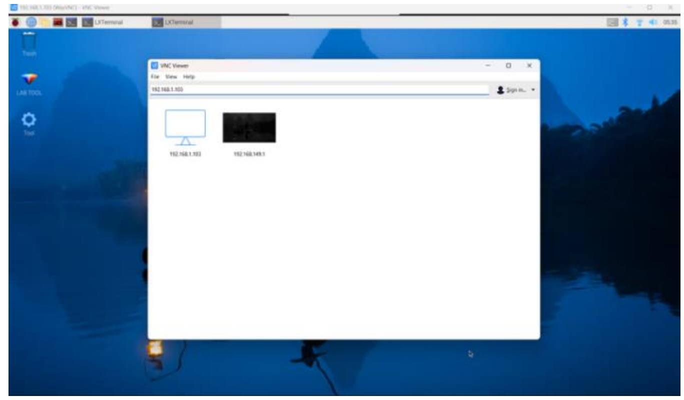

Figure 4. VNC access to Raspberry Pi 5

When analyzing **Figure 4,** the IP address "192.168.149.1" must be entered in the search bar. The username and password will be asked as shown in **Figure 5**.

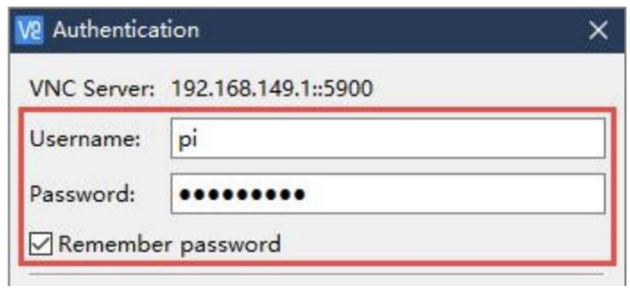

Figure 5. Username and Password for VNC

As shown in **Figure 5**, the username will be "pi", and the password will be "raspberrypi." Now, the raspberry pi software can be utilized where you have access to inserting commands for the robot. Shown later, the software will be used for commanding the robot for SLAM mapping and Lane Keeping.

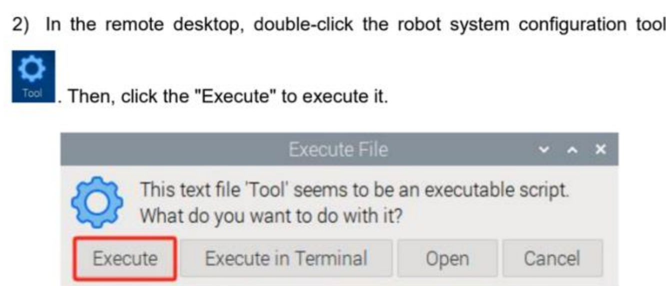

Figure 6. Robot System Configuration Tool

Figure 7. Changing Robot Version Configuration

In our case the robot version was in "MentorPi_Mecanum." You must manually change the configuration of the chassis to "MentorPi_Acker," shown in **Figure 7,** since the robot is using an Ackerman chassis. The Mecanum-wheel chassis is for omni-directional movement which will not work in this case.

Lane Keeping

Lane keeping can be established by making a map first. All that is necessary is colored tape which can be used on any floor.

Figure 8. DIY Map with Yellow Tape

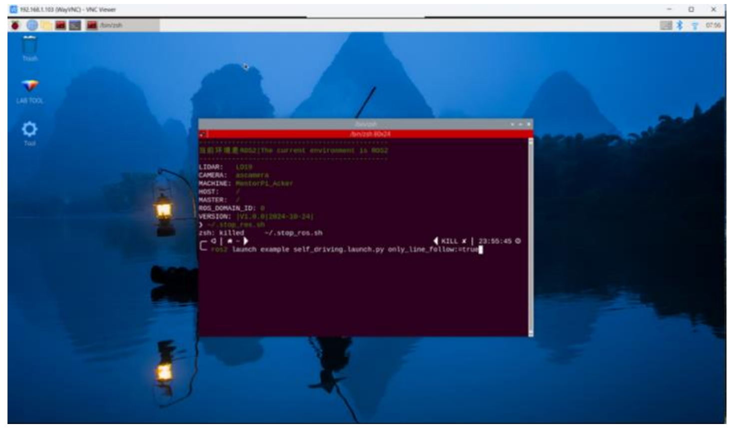

Figure 9. Self-Driving and Lane Keeping Command in Terminator

Once making a map as shown in **Figure 8**, the command can be inputted into VNC, referring to **Figure 9**, the robot will start self-driving once put onto the track and lane keeping will occur. Keep in mind, any map layout can be made according to the user's choice.

SLAM Mapping

For SLAM mapping, it's crucial to download Workstation Pro 17 instead of 16 because Hiwonder's preinstalled virtual machine setup for SLAM mapping will not work.

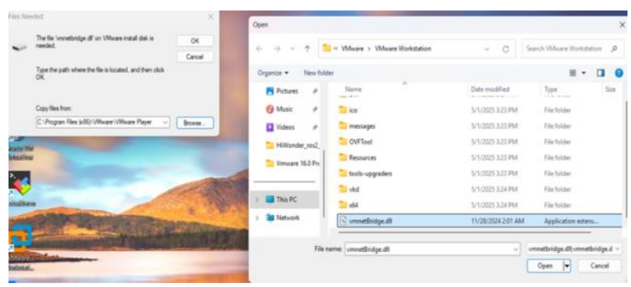

Figure 10. "vmnetbridge.dll" error message

Once installing Workstation Pro 17.5 or the other latest version, you will likely experience an error in installation. NOTE: This is extremely important to address or SLAM mapping will not work. The "vmnetbridge.dll" is used for the network bridge adapter which will be utilized in Workstation Pro 17. When receiving this error, find the correct path in the VMware Workstation folder, shown in **Figure 10.**

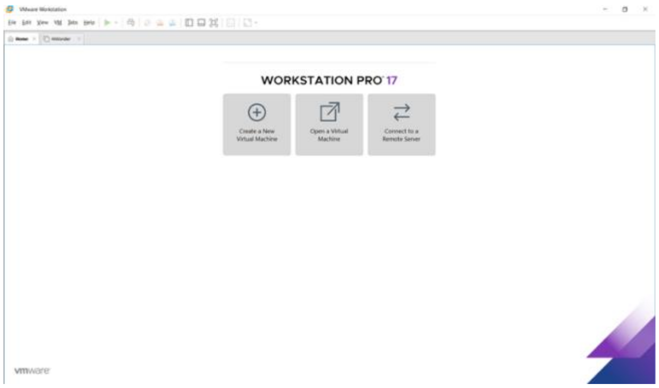

Figure 11. Workstation Pro 17

Figure 12. Zipped Folder for Hiwonder Virtual Machine

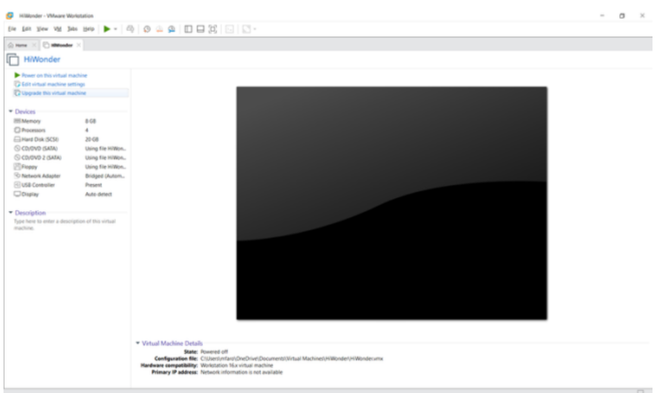

Figure 13. Hiwonder Virtual Machine

To access the Hiwonder virtual machine designated for SLAM mapping, it is located within the Appendix of the google drive folder and found within “Virtual Machine Images.” Once found, download the folder as shown in **Figure 12.** Once downloaded, click on “Open a virtual machine,” shown in **Figure 11.** Once opened, you can title the virtual machine anything you would like; I titled the virtual machine “Hiwonder,” which is shown in **Figure 13.** Now, the virtual machine is ready for use and can now communicate with the robot.

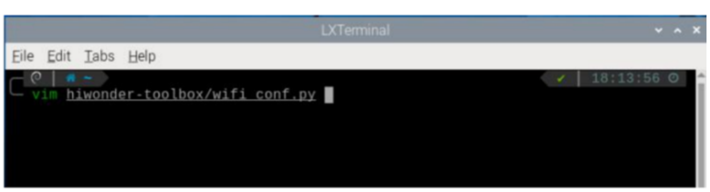

Figure 14. VNC Terminal to Access WIFI Settings

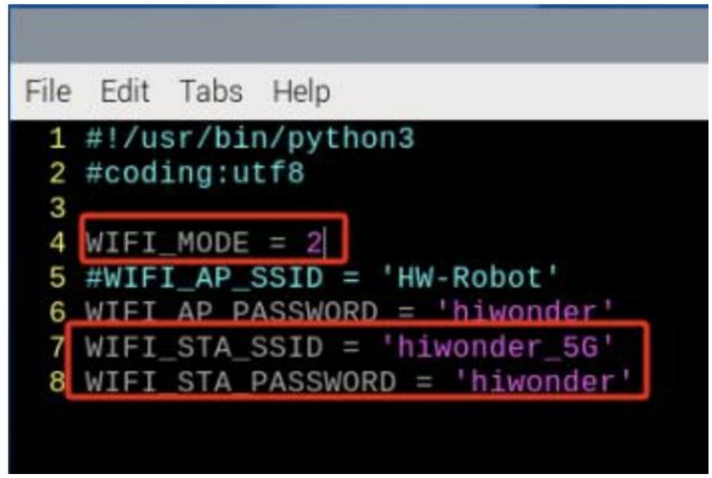

Figure 15. LAN Mode Configuration

Next, to establish connection between the robot and virtual machine, place the robot in LAN mode. Going to VNC and clicking on the terminal, using **Figure 14** for reference, and typing in the code to edit the WIFI settings for the robot, change the WIFI Mode from 1 to 2, as shown in **Figure 15.** Next, enter your home WIFI username and password in the STA ID and password.

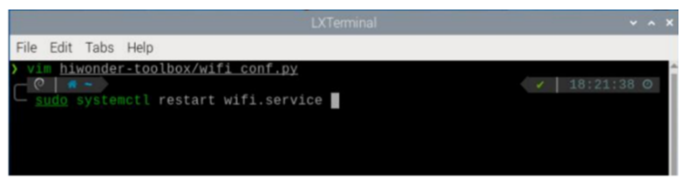

Figure 16. Restart Robot Command

Looking at **Figure 16,** once entering your personal WIFI settings, enter the command as shown. The robot will reboot and will be connected to your personal WIFI. Since you will be connected to your home WIFI, your IP address will change, and you will have to login through VNC with your new IP address.

Figure 17. Option #1 of Finding New IP Address

Figure 18. Method #2 of Finding New IP Address

There are two methods of finding the new IP address since you cannot connect to VNC currently. Looking at **Figure 4** in my case, I have two IP addresses because I connected the robot to my WIFI and I found the new IP address through method #2, shown in **Figure 18**, when I connected a micro-USB to HDMI cable from Raspberry Pi to a monitor.

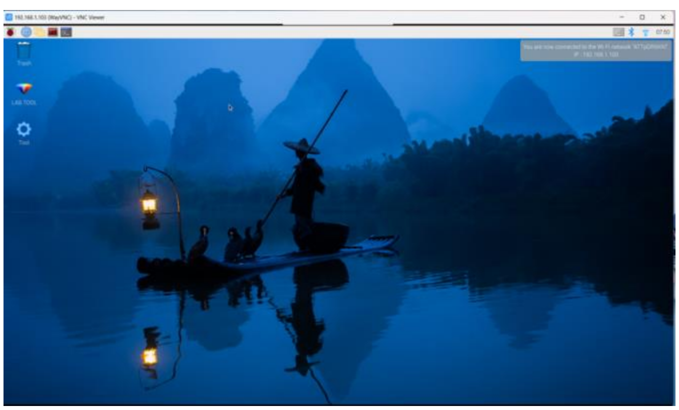

Figure 19. Raspberry Pi Screen with New IP Address

As shown in **Figure 19**, the new IP address is located at the top right in which you can enter this into the VNC. The username and password, shown in **Figure 5**, remain the same as before. Now, SLAM mapping will work correctly; communication between the virtual machine on a personal laptop and Hiwonder robot will occur.

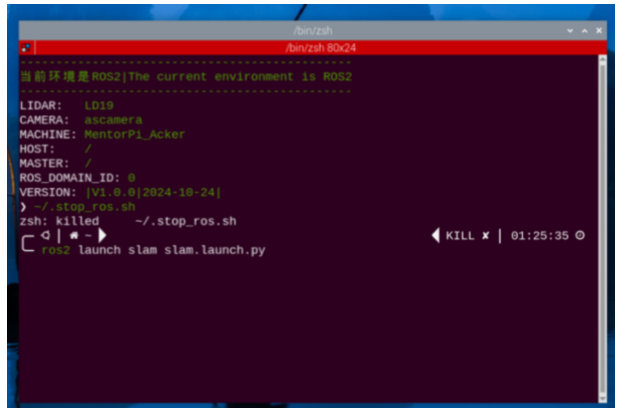

Figure 20. Insert Command for SLAM Mapping in VNC

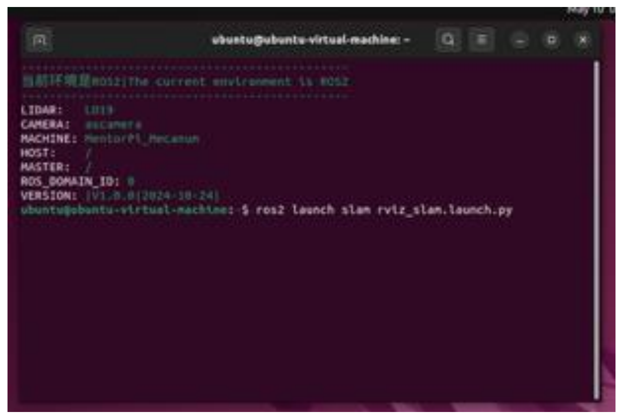

Figure 21. Insert Command for SLAM Mapping in Virtual Machine

Figure 20 and **Figure 21** are the last steps which allow the robot and virtual machine to be connected to each other. Once the command through VNC is inputted, the robot will start SLAM Mapping. Then, once inputting the command into a virtual machine, Rviz will open, and the environment will be shown and detected by the Hiwonder machine.

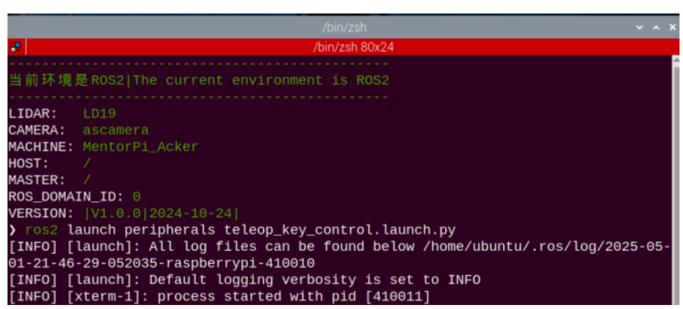

Figure 22. Insert Command to Control Robot from Keyboard

Figure 22 shows the command to input in a separate terminator within VNC to control the robot from a keyboard. Shown in the results below with a video, the robot will be able to move from your personal computer which will be able to map out the environment, shown in Rviz.

3. RESULTS

Lane Keeping

[Lane Keeping Demo]

In the Lane Keeping demo, the map used is referenced in **Figure 8**. The Hiwonder robot is successfully able to accomplish lane keeping by staying in between both yellow lines. However, towards the very end of the video, the autonomous vehicle is unable to make the sharp turn. Unfortunately, the Ackerman chassis cannot make sharp turns but making a map with a gradual turn, will allow the robot to successfully demonstrate lane keeping. For users attempting to replicate lane keeping, ensure the lanes are about 15 to 20 inches wide. In **Figure 8**, the lanes were 17 inches wide.

SLAM Mapping

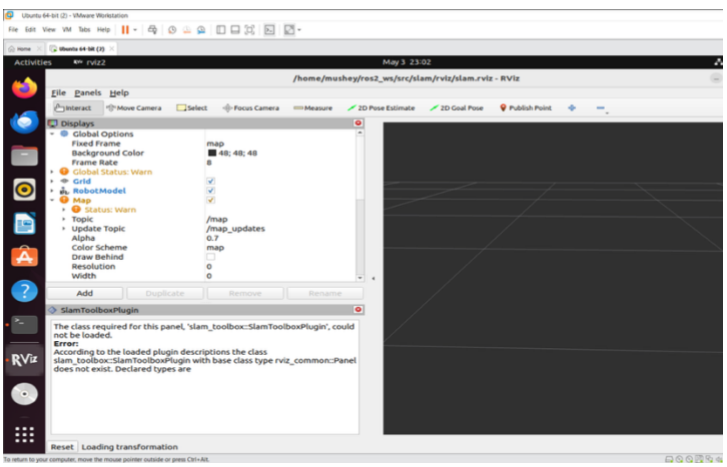

Figure 23. Rviz showing slam_toolbox Error

When first beginning SLAM mapping, I ran into a slam_toolbox error. According to Hiwonder's documentation, you must use the Hiwonder virtual disk/machine for SLAM mapping to work, as shown in **Figure 12** and **Figure 13**. Hiwonder has a lot of text which is unnecessary which requires a user to import files from the robot onto your own personal computer. It is unnecessary as the Hiwonder virtual machine has all the files required for SLAM mapping and does not require any importing.

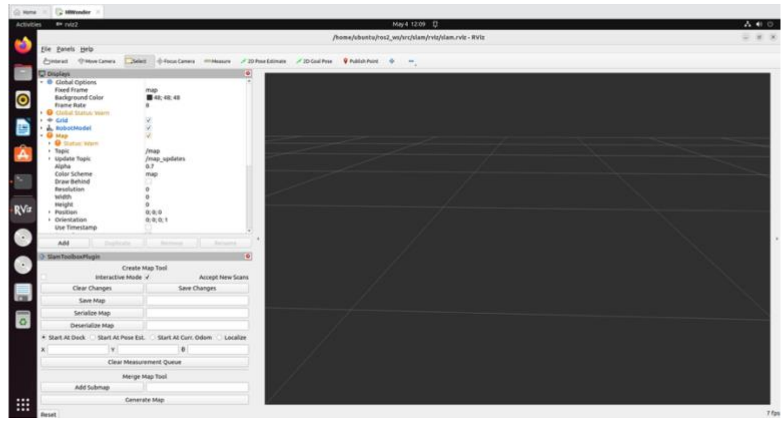

Figure 24. Rviz showing Map Not Received Error

In **Figure 24**, the map not being received indicates a communication error between the Hiwonder robot and virtual machine. In this case, I was using Hiwonder's virtual machine but was not able to establish communication. Referring to **Figure 10**, when I first installed VM Workstation Pro 17, I came across an error, "vmnetbridge.dll, " which was related to the network bridge adapter that Workstation uses to communicate with other devices in a network. Uninstalling and reinstalling Workstation Pro 17, I came across the error once again and was able to give a path to the "vmnetbridge.dll" file which allowed SLAM Mapping to work.

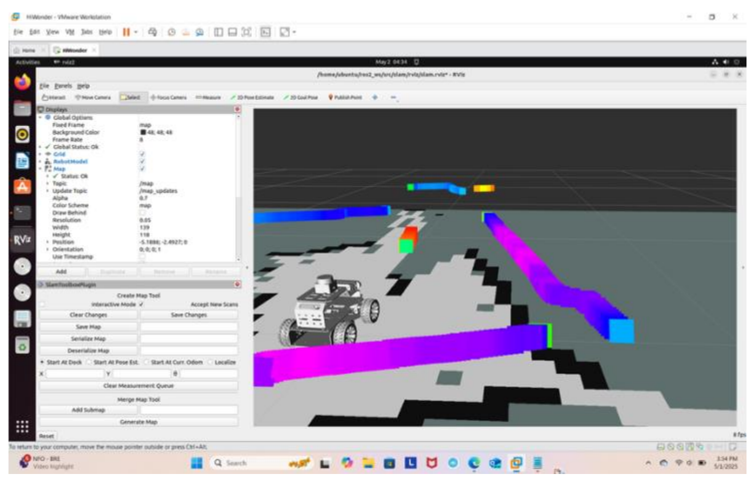

Figure 25. Rviz showing Environment of Robot

Now, capable of SLAM mapping, Rviz shows the environment of the robot. With a connection between a user's virtual machine and Hiwonder robot, the environment of the robot can be detected. In addition, the keyboard from a user's personal computer can be used to move the robot to map out the entire environment. Furthermore, we can verify there are no errors in Rviz because all of boxes are checked green such as the robot model, map, and global status.

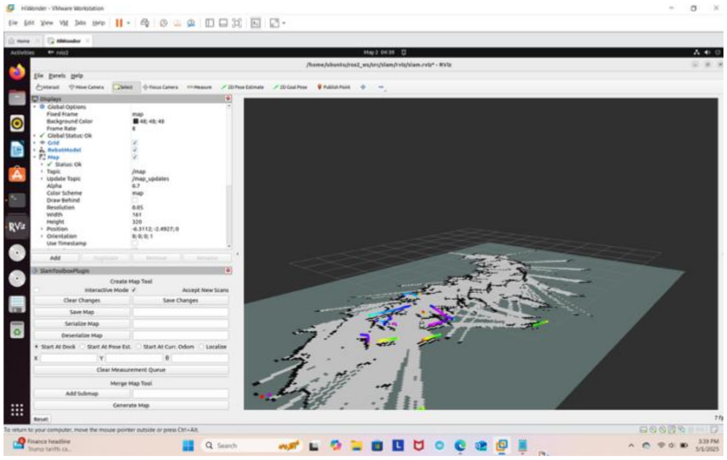

Figure 26. Entire view of Rviz Environment

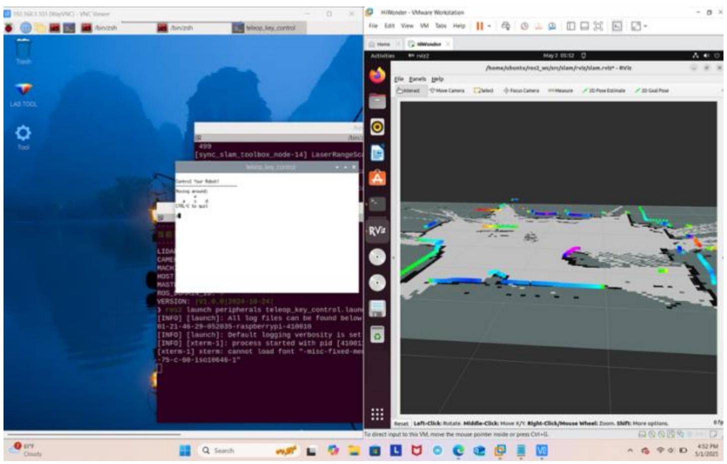

Figure 27. Side by Side View of VNC and Rviz (Virtual Machine)

When analyzing **Figure 26** and **Figure 27**, the robot can now be controlled by the keyboard which will map out the entire environment. As the robot moves, the Rviz environment will expand to reflect the current environment of the robot. Due to the Lidar sensor that is equipped, obstacles will be detected.

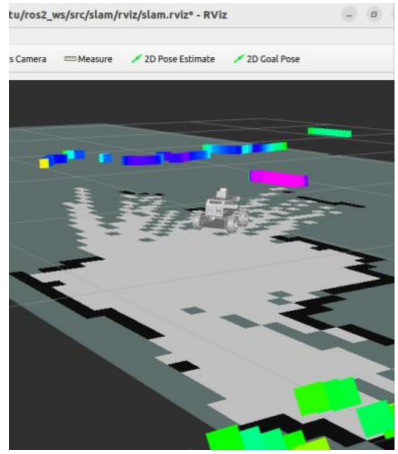

Figure 28: Obstacle Placed in Front of MentorPi Robot

Figure 29: Obstacle Placed in Front of MentorPi Robot (Realtime Environment)

Figure 28 and **Figure 29** are the comparison of real-time vs Rviz environment. Placing a box in front of the Hiwonder Robot, it will detect the obstacle and stop movement, while using the keyboard as shown in **Figure 27**. Analyzing **Figure 28**, the pink line is represented by the cardboard box in front of the Hiwonder vehicle. This proves obstacle avoidance is detected during SLAM mapping. The SLAM mapping demo video will give readers a better idea of SLAM mapping.

SLAM Mapping Demo

When watching the video, the robot can detect my glass table which is located at the bottom right of the video which explains the lines it creates to indicate an obstacle in Rviz. Also, the video demonstrates the keyboard usage as well as SLAM mapping. To explain why the vehicle runs into the monitor, the robot must see the obstacle further in advance to avoid the obstacle.

4. DISCUSSION

Lane keeping was established which allows the Hiwonder robot to function as an autonomous vehicle as shown in the Lane Keeping Demo. By creating a track, shown in **Figure 8**, which utilizes yellow tape to establish 17-inch-wide lanes, the autonomous vehicle was able to successfully perform lane keeping until a sharp turn occurred. For SLAM mapping, through rigorous debugging, communication between a user's virtual machine and robot was established which maps out a real environment in Rviz. Utilizing the keyboard, shown in **Figure 27**, the robot can be moved utilizing the "w" and "s" keys while turning uses the "a" and "d" keys.

For the Lane keeping demo video, the Hiwonder robot can perform lane keeping until a sharp turn occurs. We can conclude that the Ackerman chassis cannot perform sharp turns. To remedy this issue, the map must take into consideration that the robot cannot make sudden turns. Gradual turns through the map can be made to test lane keeping even further.

As shown in the SLAM Mapping demo video, the Hiwonder robot can successfully map out an environment and detect obstacles. Furthermore, as shown in **Figures 28** and **29**, obstacle avoidance is found within SLAM mapping which demonstrates automatic emergency braking to an extent.

Some limitations to my approach involved many debugging issues which resulted in SLAM mapping to take longer than expected. In addition, obstacle avoidance within SLAM mapping could have been explored even further. Lastly, for Lane keeping, the map shown in **Figure 8**, could have accommodated the Ackerman chassis to a greater degree resulting in more gradual versus sharper turns.

5. CONCLUSION AND FUTURE WORK

To conclude, lane keeping was straightforward to run since VNC is only needed. The lane keeping demo highlights the robot's 3D depth camera which recognizes the yellow lines. However, the Ackerman chassis is where the robot falls short since it is not capable of sharp turns. For SLAM mapping, much time was put into the debugging process since it can be quite difficult to establish a connection between the robot and virtual machine. With conflicting documentation, it can be a challenging process. Nonetheless, SLAM mapping was able to be run, and the environment can be mapped around the robot. However, going forward, future research can go into obstacle avoidance within SLAM mapping and mapping out even bigger environments.

For future work, research can be extended towards turning decisions and automatic emergency parking. Looking at **Appendix C**, Hiwonder has a map available for purchase which simulates traffic scenes, parking lots, and sidewalks. In addition, the map comes with four traffic signs and one traffic light which highlight the vehicle's turning decisions. Since there are empty parking lots, automatic parking can be achieved as well. Within the google drive folder of autonomous driving for Hiwonder, the robot is capable of road sign detection, traffic light recognition, turning decisions, and autonomous parking.

REFERENCES

[1] Hiwonder. MentorPi 2024. Google Drive Folder. Available:
https://drive.google.com/drive/folders/1Ox5xN5zpxXqDK-9ruDwwgcQgePXvMvHr?usp=sharing

APPENDIX

A. Keyboard Controls for SLAM Mapping

Key	Robot Action
W	Short press to switch to forward state and continuously move forward
S	Short press to switch to backward state and continuously move backward
A	Long press to interrupt the forward or backward state and turn left
D	Long press to interrupt the forward or backward state and turn right

Figure 30. Keyboard Controls

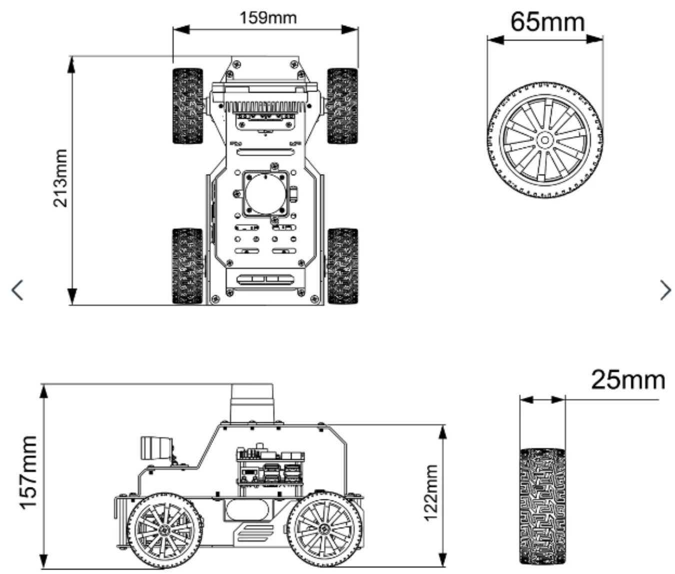

Figure 32. Dimensional Diagram

B. Specs of Robot

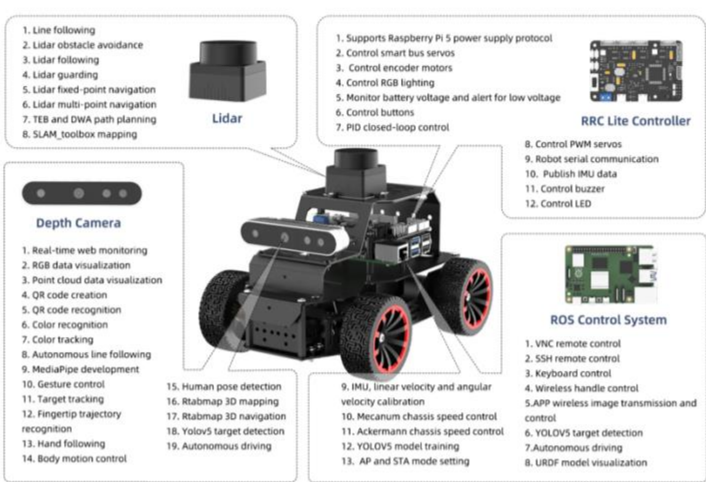

Figure 31. Function List of Hiwonder Robot

Size	213*159*157mm(Depth Camera Version)
Weight	1.2kg
Servo type	LD-1501MG servo and LFD-01 anti-blocking servo (Monocular camera version)
Motor	310 metal gear geared motor
Encoder	AB-phase high-accuracy quadrature encoder
Material	Full metal aluminum alloy chassis, anodizing process
ROS controller	RRC Lite controller + Raspberry Pi 5 controller
Control method	App, wireless handle and PC control
Camera	Angstrong binocluar 3D depth camera
Lidar	ldrobot STL-19P
OS	Raspberry Pi OS + Ubuntu 22.04 LTS + ROS2 Humble (Docker)
Software	iOS/ Android app
Communication method	WiFi/ Ethernet
Programming language	Python/ C/ C++/ JavaScript
Storage	64GB TF card

Figure 33. Product Parameters

C. Hiwonder Map

Figure 34. Hiwonder Map Utilizing Autonomous Driving