UOW
COLLEGE
HONG KONG
香港伍倫貢學院

AST 21118

Engineering Product Development

Project Report

Project Title:

"JetRover: A Self Navigator Educational Robot"

– Experiential + Applied Project –

Group Details:

Shadev Kharel(H10006790)
Lee Long To(H10007221)
Kao Fung Ting (H10006783)
Lee Chun Wing(H10006057)
Wang Yicheng(H10006993)

Submitted to: Dr. Junpei Zhong(Joni) **Date: 28th April, 2025**

Plagiarism Declaration

We understand that Plagiarism is regarded as a very serious offence in UOWCHK. Any related offence will lead to disciplinary action.

We declare that, to the best of our knowledge and belief, this assignment is our own work, all sources have been acknowledged and the assignment contains no plagiarism.

We further declare that we have NOT previously submitted this work or any version / part of it for assessment in any other course offered by UOWCHK or any other education institution in Hong Kong or overseas. If a clear case of plagiarism is found, penalties may include failure for this course, suspension from study, expulsion from UOWCHK, and debarment from re-admission.

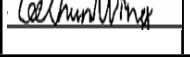
Name	Signature	Name	Signature	Name	Signature
1. Shadev Kharel*		**2. Lee Long To**		**3. Kao Fung Ting**	
4. Lee Chun Wing		**5. Wang Yicheng**			

Table of Contents

1. Introduction and Project Overview:

The AST21118 project: Jetrover, is an application oriented experiential robotics project, in which our team focused on the practical deployment of the autonomous navigation system using the Robot Operating System(ROS2). To secure the task requirements, we leveraged a pre-existing hardware solution in the form of Jetrover which is manufactured by Hiwonder in Shenzhen, China. Using the Jetrover, our primary objective was to utilize Simultaneous Localization and Mapping(SLAM), Navigation 2 Stack(Nav2), Teleop Key Control along with visualization tools like RviZ, to effectively understand, configure and integrate the complex systems which are already industry ready. Furthermore, after a brief understanding of the internal mechanisms of the Navigation subsystem in the Jetrover, we also tried to incorporate a secondary development to demonstrate a Human Robot Interaction(HRI) where a basic communication loop can be established via. a text input. Therefore, instead of programming and training the algorithms from scratch, our focus laid on fine-tuning and extending the already existing architecture of Jetrover.

Despite the core development environment being able to be set up only in a single machine due to resource constraints, each team member however was responsible for understanding a subsystem each, which led to modularization and an effective task division. The task pipeline for our team is highlighted below:

Kao Fung Ting: *Hardware Configuration & Troubleshooting + Testing*

↓ Assembly, LiDAR Issue, Speaker Integration

Lee Long To: *Led SLAM Configuration, Localization, and Mapping*

↓ Map Generated for Dance Room using SLAM

Shadev Kharel*: *System Integrator and Core Dev For Text_Nav System(TTS)*

↓ Tested Nav2 and Text_Nav Overlay System

Lee Chung Wing: *RViz Visualization and Overall Testing Coordinator*

↓ Visualized Whole Navigation System in RViZ

Wang Yicheng: *Documentation and Workflow Management.*

2. Technical Implementation

The implementation of our autonomous navigation system task on the Jetrover was a modular approach as specified in the introductory section. The central layer being the perception layer which involved: Lidar Sensors, Wheel Encoders, and Inertial Measurement Unit(IMU) related data. Data from the components in the perception layer was passed on to the SLAM stack of the Jetrover, which uses SLAM Toolbox tool by default as it is the most mature 2D SLAM implementation for ROS2. The Jetrover was then controlled manually in the physical location using Teleop Key Control functionality provided by default within the ROS2 System. The map generated by SLAM was then saved to the Navigation Server implemented in Jetrover, i.e. Navigation 2 Stack (NAV2) Stack. Utilizing the BT Navigator Stack, in the NAV2 Stack, and also using the preconfigured RVIZ Launch file, we were able to leverage GUI based Goal setting within RViZ to send navigation goals to the robot, and visualize the path planning as well. Further technicalities for the projects are categorically highlighted below:

1. Simultaneous Localization and Mapping(SLAM):

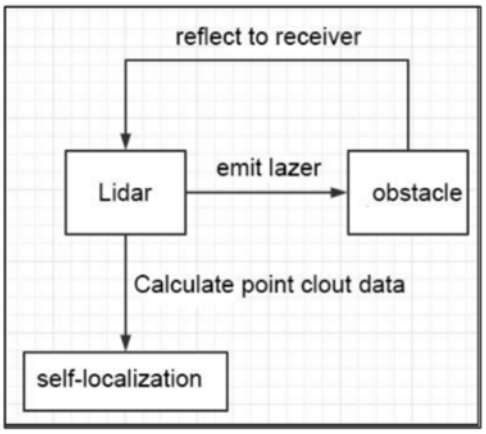
SLAM is an essential component for autonomous robots, enabling them to achieve navigation under unknown environments without prior knowledge and human control.

SLAM consists of two major processes:

1. Localization:
   Robot can determine its orientation and position data given by the coordinate system. This coordinate system can be obtained from the initial keyframe, predefined maps and landmark or GPS data, ensuring an accurate localization to track its movement that is relative to its surroundings.

2. Mapping:
   It allows the robot to create a map of its surrounding environment using the data perceived from the A1 rplidar. The map consists of geometric elements like points, which help the robot understand the obstacles like walls and available space for path planning.

Data Processing from Raw data to Mapping:



To get the raw data, the LiDAR will emit laser light pulses from the transmitter, scattering when hitting surrounding obstacles. Some of the light waves that reflect back to the receiver of the LiDAR will calculate its distance, scattered and accurate angle using the principle of laser ranging. Theses surrounding environment information is called “point cloud”, which allows the robot to estimate where it is located. But before getting the point cloud data, it will undergo optimization which can filter out problematic data (noises).

After that, “Matching” and “Map Fusion” will perform, which firstly matches the current point cloud data with the established map to find the corresponding position, then integrate new data from the lidar into the original map to keep updating a new structured map in Rviz.

Configuration for the SLAM system in ROS2:

Frames:

```
# ROS Parameters
odom_frame: odom
map_frame: map
base_frame: base_footprint
scan_topic: scan
use_map_saver: true
mode: mapping  #localization
```

Matching parameter:

```
# Scan Matcher Parameters
distance_variance_penalty: 0.5
angle_variance_penalty: 1.0

fine_search_angle_offset: 0.00349
coarse_search_angle_offset: 0.349
coarse_angle_resolution: 0.0349
minimum_angle_penalty: 0.9
minimum_distance_penalty: 0.5
use_response_expansion: true
```

The above are some configuration parameters examples obtained from slam.yaml that control the SLAM toolbox (SLAM algorithm) to process the raw data which further performs TF transformation from odom (sensor data e.g. LiDAR data, Wheel resolutions) -> base_link(position of robot) updates to create a 2D map of space.

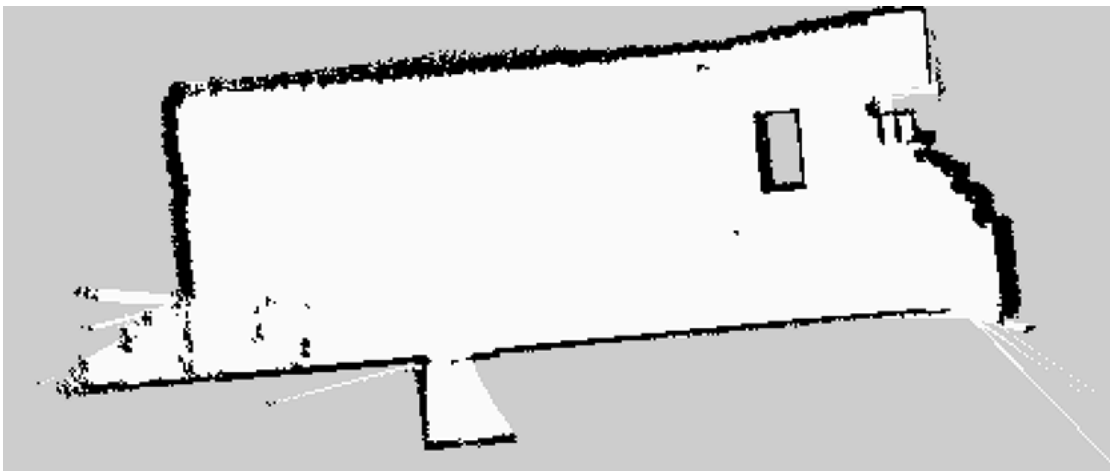
SLAM system launch file:

```
slam_launch = IncludeLaunchDescription(
    PythonLaunchDescriptionSource(
        os.path.join(slam_package_path, 'launch/include/slam_base.launch.py')),
    launch_arguments={
        'use_sim_time': use_sim_time,
        'map_frame': map_frame,
        'odom_frame': odom_frame,
        'base_frame': base_frame,
        'scan_topic': '{}/scan'.format(frame_prefix),
        'enable_save': enable_save
    }.items(),
)

if slam_method == 'slam_toolbox':
    bringup_launch = GroupAction(
     actions=[
         PushRosNamespace(robot_name),
         base_launch,
         TimerAction(
             period=5.0,  # 延时等待其它节点启动好(Delay and wait for other nodes to start up)
             actions=[slam_launch],
         ),
```

The upper slam.launch.py file initializes the SLAM system which first configures the major SLAM node by loading the slam_base.launch.py and passing the essential parameters e.g. TF transformation frames (map, odom, base_link) for accurate positioning, simulation time setting to parameterize for different environments (real or simulation), LiDAR scan topic and saving(Robotis E-Manual, n.d.). Following by the bringing_launch for sequencing the startup, which firstly assigning unique namespace to avoid topic conflicts, then launching the required nodes (LIDAR driver, wheel encoders, IMU), and critical delay to ensure stable data flow before activating the SLAM toolbox algorithm.

Our Final Map:

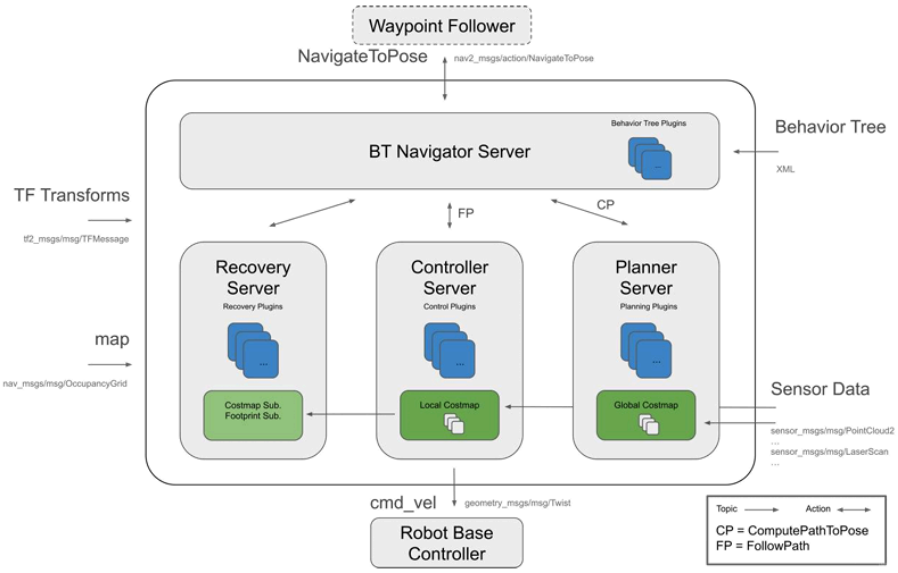


After several trials, we finalized our mapping conducted inside the dance room. As you can see it has a clear geometric structure, the rectangular layout is distinctly captured, having sharp boundaries and minimal noises. The well-defined corners and straight edges provide precise landmarks for the robot localization, reducing its pose estimation errors, making it easier for further navigation.

2. Navigation 2 Stack(Nav2): ROS2 offers its prebuilt Navigation Stack, which only needs to be configured according to the robots physical mechanics. For example: In the nav2 params.yaml for Jetrover, the topic for sensor data is defined as /scan which is a standard practice in ROS2 (ROS2, 2024), having said that, custom parameters involving robots mechanics, torque, and physical dimensions, has to be manually specified so that all the data are processed accurately for an effective navigation functionality. The hiwonder's Jetrover also uses Nav2 by ROS2.

```
navigation_launch = IncludeLaunchDescription(
    PythonLaunchDescriptionSource(os.path.join(navigation_package_path, 'launch/include/bringup.launch.py')),
    launch_arguments={
        'use_sim_time': use_sim_time,
        'map': os.path.join(slam_package_path, 'maps', map_name + '.yaml'),
        'params_file': os.path.join(navigation_package_path, 'config', 'nav2_params.yaml'),
        'namespace': robot_name,
        'use_namespace': use_namespace,
        'autostart': 'true',
        'use_teb': use_teb,
    }.items(),
)
```

In the navigation.launch file above , the Launch description launches bringup.launch.py which launches all the servers like map_server, slam_toolbox, bt_navigator and even the recovery_server. This is a default ROS2 feature provided by NAV2 Controllers, and even for the jetrover the same bringup.launch.py is cloned into the ros2_ws for navigation purposes.



(Fig: BT Navigator Server Stack:The Brain of NAV2, Source: Hiwonder)

The URDF of Jetrover in Jetrover_Description folder of Simulations directory gives a complete simulated model of the Jetrover, but owing to time constraints, and limited knowledge, we couldn't understand the complete simulated model for Jetrover. Despite that, after launching the navigation node, we also launched the RViZ launch file for visually providing goal targets, and seeing path plannings done by the Navigation system.

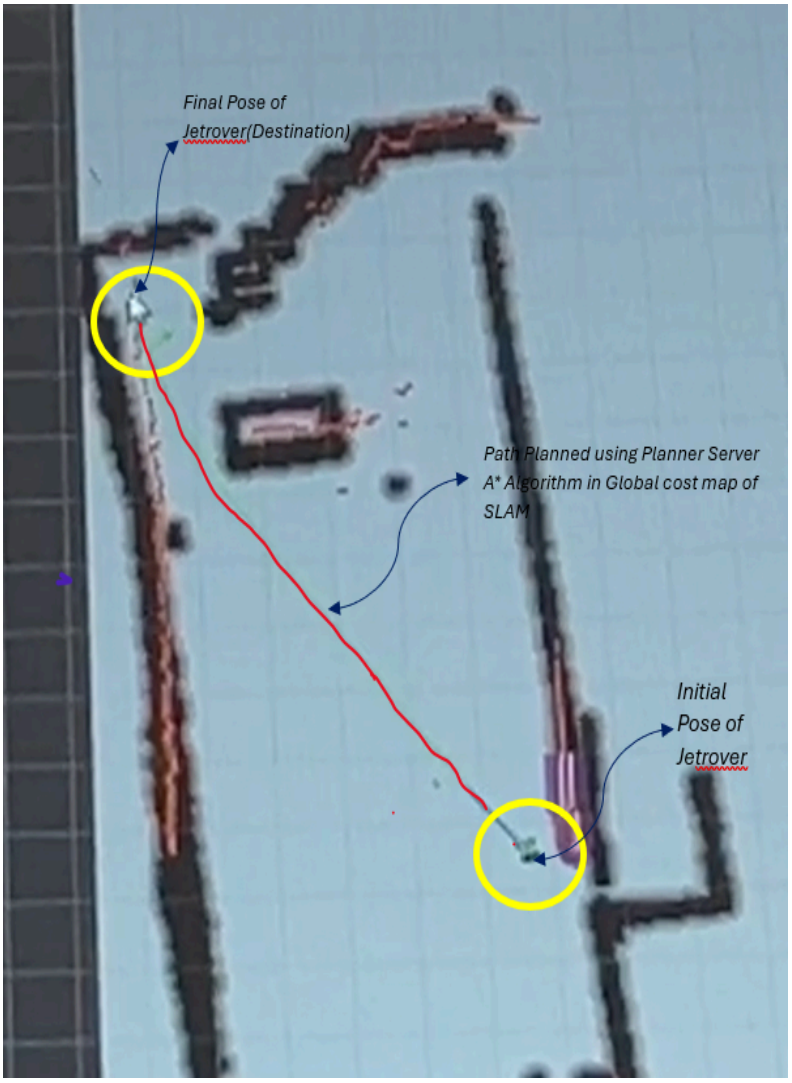
For Navigation, the step by step highlights for our project is mentioned below:

1. Configured the navigation launch files to work with our central controller Laptop Acer Nitro 5.
2. After a successful colcon build, we launched the Nav2 launch file using Raspberry Pi 5 terminal. Along with navigation, it also launches robot.launch.py as a base launch. The robot.launch.py is responsible to declare the topics needed for navigation like /cmd_vel, /scan, /odom, /base, which are responsible to send movement(velocity commands) to the robot , receive, and publish LiDAR scan data or robot wheel positions.(Kamath, 2022)

```
def launch_setup(context):
    map_frame = '{}map'.format(frame_prefix) if robot_name == master_name else '{}/map'.format(master_name)
    cmd_vel_topic = '{}/controller/cmd_vel'.format(topic_prefix)
    scan_raw = '{}/scan_raw'.format(topic_prefix)
    scan_topic = '{}/scan'.format(topic_prefix)
    odom_frame = '{}odom'.format(frame_prefix)
    base_frame = '{}base_footprint'.format(frame_prefix)
    lidar_frame = '{}lidar_frame'.format(frame_prefix)
    imu_frame = '{}imu_link'.format(frame_prefix)
```

(Fig: defining topics and frames, to which navigation nodes subscribe to.)

3. The map generated as specified in SLAM section of this report, was used as global costmap for path planning,



(Fig: Screenshot from RViZ demonstrating Initial and Final Pose)

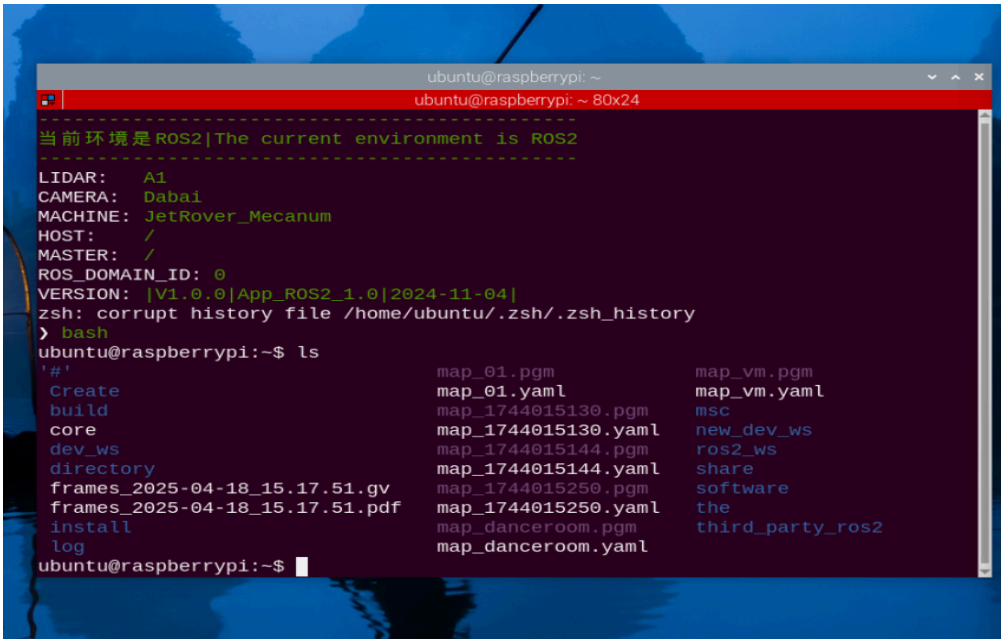
(For Clarity, please refer to appendix 2 of this report (Video Demonstration))

4. Then along the path, if the LiDAR detects any dynamic obstacles, the controller server which uses local costmap*(by subscribing to laser and wheel encoded data topics)*, detects it, and dynamically creates a new path for the robot to reach the destination, then sends messages to /cmd_vel topic for relocalized movement.

5. This way the robot was capable of navigating point to point autonomously.

3. Custom *'new_dev_ws'* Workspace for basic HRI :

To further expand the navigation functionality to work with an user input, I created an overlay workspace on top of the originally configured ros2_ws, named new_dev_ws, this workspace uses pre existing custom navigation nodes already available in the Jetrover, and extends it to be functional with user input.



(Fig. new_dev_ws alongside ros2_ws in raspberry pi file system)

In the new_dev_ws, we created a launch file to launch all the necessary packages for navigation, slam, and simulations functionality.

On top of that, a node which takes in string input via. ros2 topic pub command on the terminal was implemented. The node imported necessary dependencies and an action server needed for the input of this node to be sent out as a goal for the navigation system. In fact, I found the coordinates for two of the points in the map_house that I created using the SLAM system in the testing environment of my house.***(The map is attached to Appendix 3).***

```
# Action client for navigation
        self._action_client = ActionClient(self, NavigateToPose, 'navigate_to_pose')

        # Predefined locations (example)
        self.destinations = {
            "door": (0.420722, 0.618713, 0),
            "eating": (2.27117, 0.886419, 0)
        }
```

One point is named the *door*, and another called *eating*, and both have their effective RViZ coordinates hardcoded in the python script. When the user opens a new terminal and sends this command:

ros2 topic pub /text_nav_command std_msgs/String "data: 'door'"

Then, the text_nav_node checks out and maps the coordinate in accordance to the user input, then converts the euler (x,y,z) formatted coordinates into quaternion form which is required for sending out goals to navigate to pose action server.

```
 def send_goal(self, destination_name):
    # Extract coordinates for the destination
      x, y, yaw = self.destinations[destination_name]
      qx, qy, qz, qw = self.euler_to_quaternion(0.0, 0.0, yaw)
```

Also, to integrate the speaker functionalities, we did the `import os` to leverage the default speaker using espeak module.[4] The speaker along with the sound card is an external module we bought from a third party store.

```
        self.get_logger().info("TextNav node started and waiting for commands on /text_nav_command.")
        self.speak("Text navigation node started successfully.")
self.speak("Hello Professor, I am Ready to Navigate.")
```

When user gets prompted waiting for commands in the terminal, there is a introductory feedback from a robot stating the node has successfully launched and it is ready for navigation, and when user successfully inputs a destination via string input, i.e. ‘Door’ or ‘Eating’ in this case, the robot says: *“ALRIGHT, Going to + ‘User Input’”*

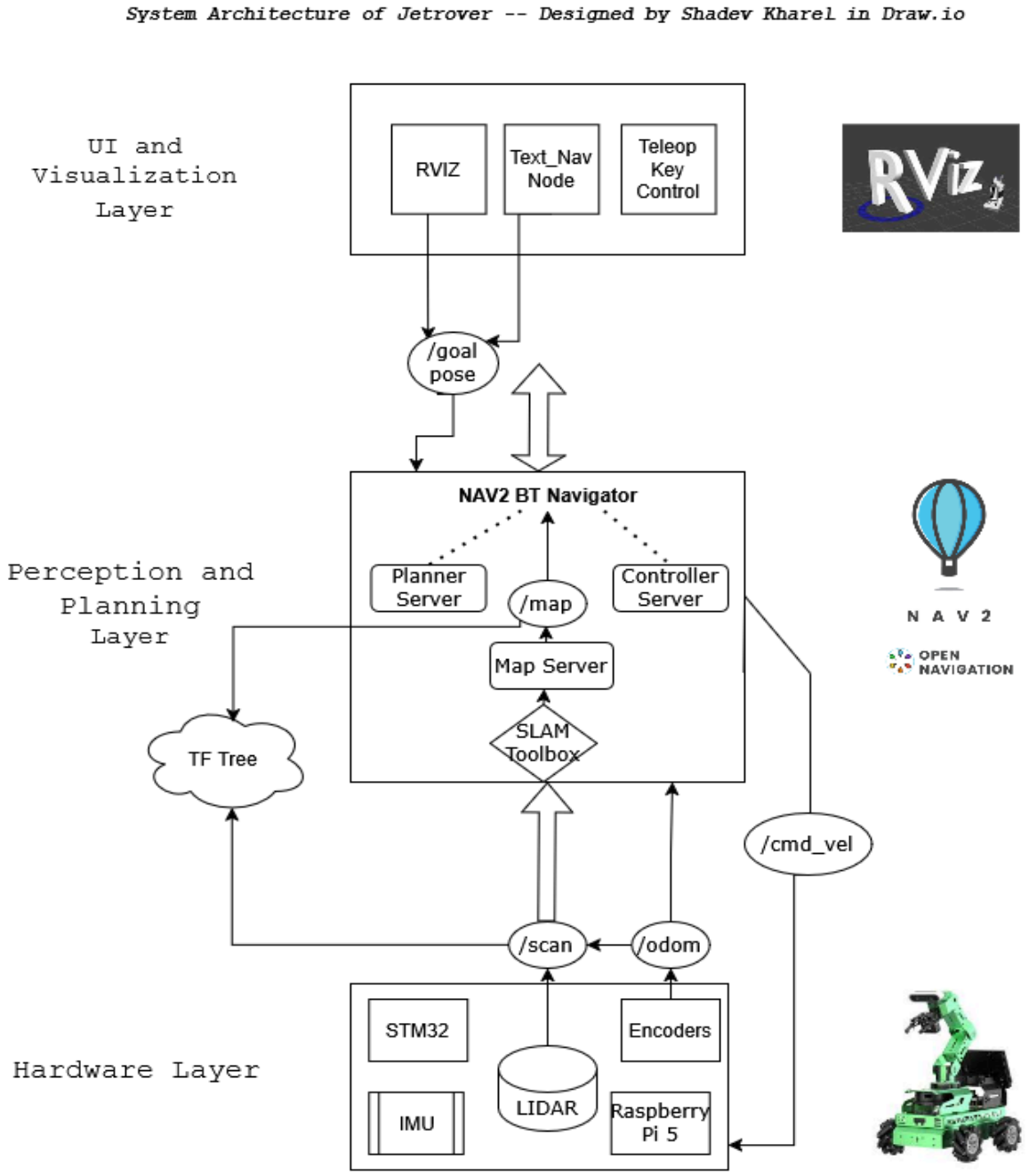
```
destination_name = msg.data.strip().lower()
self.speak("Alright, Going to:" + destination_name)
```

This is how we utilized the underlying navigation subsystem and added a basic text based navigation system on top of it, with a timely speaker feedback system. However, this is very scalable with a microphone module and Speech-to-text system in the future, but we weren't able to implement that functionality currently due to time, knowledge and resource constraints.

Please Refer to Section 3 in the Next Page for the System Architecture…

3. System Architecture of Jetrover for Task 2.

Oval → Topics



4. Technical Failures and Learnings

Failure	Description	Learning/Fix	Severity
Connection failure	The ROS2 system couldn't connect to the virtual machine.	Because of the Laptops resource constraints, i.e. Low Memory, Incompatible Graphics Card, not everyone of us could set up the Virtual Machine environment to work with the Robot Remotely. We got to learn that ROS2 is resource intensive, and was not feasible to work without prior setup, and understanding.	**High.** Inability to connect to the Raspberry Pi's system through the computer, made it difficult to work on the system, and utilize the time efficiently.
LiDAR communication failure	The LiDAR couldn't communicate with the Raspberry pi. *(Error Code: 80008004 leading to operation timeout.)*	We rewired the Jetrover's Hardware, and made sure the USB cable connecting LIDAR to the Pi had enough power, and sudo permissions to read. After we reinstalled the LiDAR and reduced the default baud rate, the LiDAR started communication with Raspberry Pi.	**High.** If the LiDAR cannot communicate with the raspberry pi, the Raspberry pi cannot transmit /scan data to the RViz and the RViz cannot create the map of the room(TF Transform Failure). It caused the mapping error.
RViz transmission failure	The data from LiDAR couldn't be transmitted to RViz.	After the LiDAR starts communication with Raspberry Pi, the RViz starts generating the map.	**High.** RViz couldn't create a map of the room. It caused a delay in testing and configuring the Navigation subsystem.

Network Connection Failure(Fixed.)	The VNC Viewer couldn't connect to the visual display of the Raspberry Pi 5 of Jetrover, due to the lack of Internet connection.	External Keyboard was directly connected to the RaspberryPi 5 and inside the Ubuntu System we visually connected to the college Wi-Fi. We got to learn about PEAP, an authentication networking protocol used by enterprise networks like UOWCHK_SS.	**Low.**

5. Future Scope of the Jetrover:

The Hiwonder Jetrover is an excellent platform for students like us to learn more about ROS2 Systems, Microcontrollers like RaspberryPi 5, and various hardwares like: LiDAR, Robotic Arm, Servo Motors, and so on. Jetrover is already an industry efficient solution, and could be leveraged for various purposes. For example: In an ongoing Indian Premier League (IPL), A Robotic Dog named *'Champak'*(Olympics, 2025), capable of hand gestures, visual processing, and leveraging smart AI is used to gain attraction. In fact with the extension of some other custom interfaces, and various deep learning models, we can further commercialize the Jetrover like *'Champak'*, considering the fact that Jetrover is an Open Source Rover offered by Hiwonder. Despite that, two of the largest domain, where Jetrover finds its application are:

1. **Education & Research:**

   - STEM education inspiration:

   Jetrover's versatility encourages innovation, and its modular design allows modification to complete different tasks. It provides a platform for students' like us, an adaptive learning to hold the whole project and find their way to achieve the goal. For example, we could leverage interchangeable chassis for achieving different specific environments such as Mecanum wheel, tank treads, and Ackermann steering. At the same time, those changes could also have a corresponding update to the software. Building up a concept of a project process and an appropriate division of labor.

2. **Industrial and Service Robotics Applications:**

   - Logistics and Warehousing: With installing suitable hardware and upgrade, Jetrover could move different objects and accomplish warehousing systems and Human-Robot Collaboration. Autonomous material transport with multi-robot coordination for warehouse automation. However, the operating hours of it are not enough for warehousing. The power support of the Jetrover should be upgraded for stable working efficiency.

6. Individual Reflections:

6.1. Reflection of Shadev Kharel :

The AST21118 Engineering Project Development was challenging, and initially an overwhelming project for me who had limited or near zero experience with ROS2. Infact, I didn't even have any prior experiences with Linux based systems, therefore jumping straight to ROS2, while also considering the hardware configuration of the Jetrover was perplexing, and frustrating at the beginning. However, as a group we started to break down tasks, and worked together in a modular approach. We followed online tutorials to work with the Jetrover, and understood the basics of the underlying hardwares and software configurations of it, and had built up the confidence to complete the fundamentals of the task specification 2.

Personally, working on the architecture, specifically, the navigation based system, gave me a first hand exposure to how robotics components integrate. I got to understand the core ROS2 functionalities like simultaneous localization and mapping, NAV2 , and RViZ. With all this, we completed the point to point navigation, with dynamic obstacle detection, and I also understood the basic theory behind path planning i.e. the Dijkstra Algorithm, whose theory I had learnt while studying the course Data Structures and Algorithm in the Semester A of AY24/25. While the original task specification was the integration of voice controlled navigation system, due to time and system complexity, we rather rescaled it to a much smaller text-based input system. Similarly, during the development of the custom coded workspace, new_dev_ws, I got to learn the need for adaptability in engineering. Not only that, as the individual posessing the central computer for developmental tasks, I learnt soft skills like leadership to bring all group members together and troubleshoot issues on hand.

Therefore, it was a chaotic and perplexing experience for us initially, but by the end of the semester, we as a group have managed to demonstrate our understanding regarding the task specification, make timely decisions, and complete what was within our understanding. Therefore, it was a fruitful experience to work with, and will serve as a strong foundation for my future robotics journey.

Signature: 

Date: 25th April, 2025

6.2. Reflection of Lee Long To :

Robotic research projects have always been an exciting yet intimidating frontier for me, joining this course is my first in-depth and hands-on experience in this field. As a beginner, I found it quite challenging and stressful during the first stage of the project. However, after persistent trials, dedicated learning and methodical problem solving, we overcame each obstacle one by one, getting us right on track and ultimately led us to successfully complete our autonomous navigation.

My role primarily involved comprehensive understanding of the SLAM process, to optimize the mapping for the Jetrover to further navigate efficiently, and provide technical support of both hardware and software. Initially, I was struggling with the complexity of ROS2 and the SLAM algorithm. We also missed launching the robot state publisher and some other essential drivers causing several failed maps. However, after doing some research and learning, I developed a deeper comprehension of how the algorithms interact with LiDAR data and odometry during the TF transformation, that we figured out using pre configured SLAM launch file which comes with the SLAM toolbox package and tuning some general SLAM parameters like scan buffer, distance and laser range, which highly increase the accuracy of the map. On top of that, we eventually chose a simple rectangular layout for our testing area which was easier for environment recognition. This decision simplified the path planning, enhanced the robot's ability for localization, and most importantly reduced errors during navigation. And finally, collaborating with my teammates on hardware configurations like integrating different components, fixing LiDAR and connection issues has also strengthened my problem-solving skills and teamwork.

This project actually reshaped my understanding of robotics engineering. Through this hands-on experience, I have gained not only specific technical knowledge like the SLAM operating process but also valuable engineering skills to approach complex challenges with patience and analytical thinking, leaving me excited to continue developing in robotic and autonomous systems in the future.

Signature: 
Date: 26th April, 2025

6.3. Reflection of Kao Fung Ting :

My duty is separated into two parts, hardware lead and troubleshooting lead. As the hardware and troubleshooting lead for the Jetrover project, I played a role in ensuring that hardware components operated seamlessly while addressing key issues during the system's setup and deployment. This experience is a great learning experience for me acting as an engineer in the future.

My responsibilities involved addressing hardware failures and integrating components such as the LiDAR and speaker. One of the most significant challenges I encountered was the LiDAR communication failure, which resulted in the inability to generate maps in RViz. If the maps cannot be generated, the Jetrover cannot navigate automatically. To resolve this, I debugged the LiDAR setup by reassembling the Jetrover, reinstalling the LiDAR by checking the USB connection, reducing the baud rate, and ensuring power supply sufficiency. This hands-on troubleshooting developed my diagnostic skills and taught me the importance of methodically addressing hardware issues. Besides working on hardware, I also deal with the connection failures caused by resource-intensive ROS2 virtual machines. After understanding and prior setup by us, we can work on the ROS2 virtual machines. These experiences emphasized the criticality of balancing hardware resource constraints with software dependencies.

Another challenge was resolving a VNC Viewer connection failure to the Raspberry Pi’s visual display. By applying networking knowledge, we established a PEAP-based Wi-Fi connection and provided direct control access to the Ubuntu system. Because of the Wi-Fi connection, we need to work on everything in the college within a limited time. This task enhanced my understanding of network authentication and strengthened my adaptability under constrained conditions.

Overall, the project taught me the value of systematic troubleshooting, attention to detail, and clear communication with team members to solve complex issues. It also reinforced my problem-solving abilities. This project is a valuable experience on the pathway to becoming a hardware engineer.

Signature: 

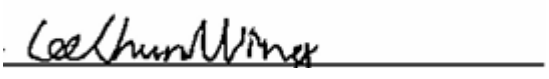
Date: 26th April, 2025

6.4. Reflection of Lee Chun Wing :

The AST21118 Engineering Project Development was challenging for me. Before this experience, it was very stressful to take the first step in a robotic field I had never had contact with. However, through collaborative problem-solving and iterative testing, I developed a greater appreciation for robotic perception, visualization tools, and system validation.

My duty is to configure RVIz for real-time visualization of the Jetrover's sensor data, navigation paths, and obstacle detection. Issues we encountered were that localization was inaccurate of the robot, which frequently caused the Jetrover to stop unexpectedly. Due to errors in pose estimation, the robot often believed it was in a different location than its actual position, leading to unnecessary halts—especially when a wall was detected in front of it. This problem highlighted the critical role of precise sensor calibration in ensuring reliable navigation. Additionally, debugging in RViZ allowed me to visualize issues like misaligned LiDAR scans, odometry drift, and incorrect cost maps, which were key to refining navigation. For solving this problem, we are verifying LiDAR and odometry data consistency to minimize drift and simplifying the testing environment to reduce mapping noise. The effect is very significant after removing the tables and chairs in the testing site and resetting the position.

While the project was chaotic at the beginning, the hands-on exposure to real-world robotic testing was invaluable. Whether it meant revisiting ROS2 documentation, testing incremental changes, or collaborating with teammates to cross-verify hypotheses. The iterative process of fail-test-refine resolved technical hurdles and improve my problem solving skills . By the end of the semester, I felt accomplished in demonstrating our understanding of the task specifications and contributing to successful navigation outcomes. This experience has laid a solid foundation for my future in robotics and has ignited my enthusiasm for further exploration in this field.

Signature: 

Date: 26th April, 2025

6.5. Reflection of Yicheng Wang :

Looking back at my experience with the group project of designing a self-navigator educational robot, I am reminded of the importance of proper documentation and workflow management in ensuring our success as a team. My role was not just to coordinate information but also to make sure that communication and coordination among team members were done smoothly. From this experience, I have learned some important lessons that I will carry with me in the future.

Emphasis on Clean Documentation: One of the most significant responsibilities I had was the development and maintenance of clean documentation. This included project requirements and design specifications, meeting minutes, and progress reports. I quickly realized that clean documentation is essential for several reasons. First, it serves as a reliable point of reference for team members, ensuring everyone is aligned when it comes to project goals and timelines. By documenting our conversations and decisions, we were able to reduce misunderstandings and remain focused on the project.

Workflow Management: Keeping the Team on Track: In addition to documentation, I also did workflow management. This involved using project management tools to create timelines, assign tasks, and track progress. I realized that a well-managed workflow is necessary to maintain momentum in a group project. By breaking down the project into small tasks and creating realistic deadlines, we were able to keep each other on our toes and maintain a steady pace.

Challenges and Learning Opportunities: Of course, the experience was not without its challenges. One of the biggest challenges was adapting to different communication styles and preferences within the team. I learned the importance of being adaptable and finding a balance that worked for everyone. This experience taught me the importance of empathy in teamwork and the need to be responsive to the team's diverse needs.

Signature: *Yicheng Wang*

Date: 26th April, 2025

7. Conclusion:

To sum up, the project successfully integrated key robotic features like: Navigation, SLAM, basic Text-to-Speech(TTS) speaker module. The robot can explore unknown areas, build maps, and move autonomously, and also provide voice feedback. Although challenges were faced along the way, such as configuring the system settings, and integrating the modules to work together, In the end however, the project showed the potential of smart building, and set a foundation for our group to step our foot in the world of robotics development.

References:

1. ROS2. (2024). *Navigation2: Configuring the costmap*. ROS2 Documentation. https://navigation.ros.org/configuration/packages/configuring-the-costmap.html

2. Kamath, A. (2022, September 4). *Digging into ROS2 Launch Files*. https://adityakamath.github.io/2022-09-04-akros2-drive-and-bringup/

3. *Creating a workspace — ROS 2 Documentation: Humble documentation*. (n.d.). https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Creating-A-Workspace/Creating-A-Workspace.html

4. GeeksforGeeks. (2024, March 29). *Using Espeak with OS in Python*. GeeksforGeeks. https://www.geeksforgeeks.org/using-espeak-with-os-in-python/

5. ROBOTIS e-Manual(n.d.). ROBOTIS e-Manual. https://emanual.robotis.com/docs/en/platform/turtlebot3/slam/

6. Karnan, H. (2018, February 14). *Connecting a Raspberry Pi 3 to enterprise wifi*. https://hareshmiriyala.wordpress.com/2018/02/13/connecting-a-raspberry-pi-3-to-enterprise-wifi/

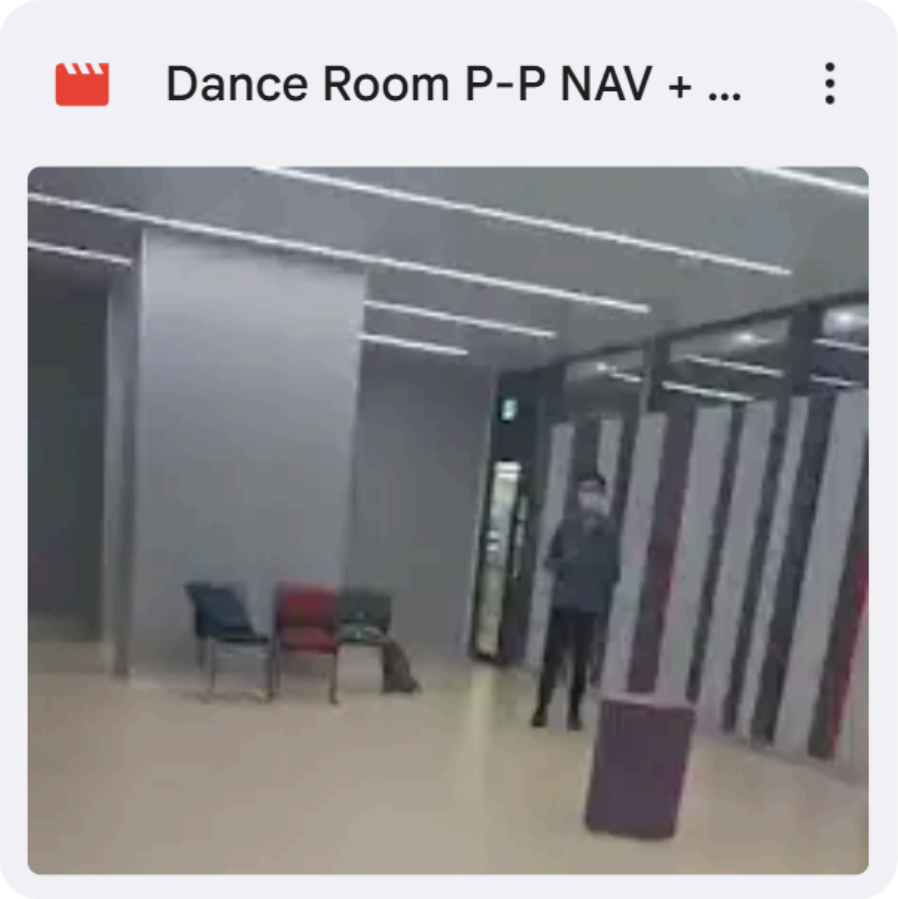
7. Olympics(2025). *IPL robot dog - all you need to know*. https://www.olympics.com/en/news/ipl-robot-dog-camera

Appendice(s):

Appendix 1:

(Pic: Dance Room IRL)

Appendix 2:



(Link: https://drive.google.com/file/d/1NaLjojUy8jR8WI70f0RSr2UxC7AJpUGL/view?usp=drive_link)

Appendix 3

For More Detailed Documentation(**Map_House** + Video + Pictures), Please Refer to this Google Drive Link: *AST 21118 - Documentation + Videos - Google Drive*

——----------END——----------