Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #46434

Fostering Student Learning and Innovation in Robotic Navigation: Field Trials and Virtual Simulations

Dr. Abhijit Nagchaudhuri, University of Maryland Eastern Shore

Dr. Abhijit Nagchaudhuri is currently a tenured professor in the Department of Engineering and Aviation Sciences at the University of Maryland Eastern Shore(UMES). Dr. Nagchaudhuri received his baccalaureate degree from Jadavpur University (India) with honors in mechanical engineering in 1983. Thereafter, he worked in a multinational industry for a little over three years before joining Tulane University as a graduate student in the fall of 1987. He received a master's degree from Tulane University in 1989 and a doctoral degree from Duke University in 1992. He is a member of the American Society for Mechanical Engineers (ASME), the American Society for Engineering Education (ASEE), and, has also been involved with the American Society for Agricultural and Biological Engineers (ASABE). He is actively involved in teaching and research in the fields of (i) robotics and mechatronics, (ii) remote sensing and precision agriculture, and,(iii) biofuels and renewable energy. He has been involved with several extramurally funded projects as PI or Co-PI to support outreach, education, and research activities from various state, federal, and private agencies; served in leadership roles in professional societies; and has received several awards and certificates from his home institution as well as professional societies. He has published more than 90 refereed articles in journals, book chapters, and conference proceedings. He has been involved with several divisions of ASEE including the Minorities in Engineering Division (MIND), and served as the division chair of the Energy Conversion, Conservation, and Nuclear Engineering Division (ECCNED) in 2016. He is a member of the technical committee of IEEE/ASME Mechatronics and Embedded Systems Application(MESA) and has served as the division chair in 2022. He serves as a program committee member of the Maryland Space Grant Consortium.

Lance Ward, University of Maryland Eastern Shore

Lance is an undergraduate student researcher at the University of Maryland Eastern Shore currently working on a bachelor's degree in engineering with a concentration in computer engineering.

Danny Pham, University of Maryland Eastern Shore

Danny Pham is a sophomore at the University of Maryland Eastern Shore, majoring in Computer Engineering. His work focuses on robotics applications using MicroROS, including line following, gesture recognition, and more. He utilizes OpenCV and Mediapipe for computer vision and AI-driven interactions.

Anubhav Dixit, University of Maryland College Park
Christopher Snyder, University of Maryland Eastern Shore

Fostering Student Learning and Innovation in Robotic Navigation: Field Trials and Virtual Simulations

Abstract

Autonomous and remote-controlled navigation trials with mobile robots equipped with various sensors including LiDAR and stereo depth cameras will be highlighted in this paper. Realistic simulation models of the devices have also been developed and employed for virtual trials, utilizing Gazebo and ROS-based open-source platforms running on the Ubuntu/Linux operating system. The project efforts reported here were largely conducted during the summer of 2024 by undergraduate engineering students at the University of Maryland Eastern Shore (UMES) and a visiting student from the University of Maryland College Park (UMD) who came to UMES for a 10-week internship at the robotics laboratory supported by Maryland Space Grant's Summer Exchange Internship Program (SEIP). Standardized software libraries and packages integral to the Robot Operating System (ROS) facilitated the implementation efforts. The students overcame significant challenges to familiarize themselves with appropriate ROS1 and ROS2 environments and assimilated them with compatible versions of the Ubuntu(Linux) operating system to advance the delineated goals. The ROS2-compatible micro-ROS was also utilized for one of the educational mobile robotic platforms. Integration of artificial intelligence and sensor feedback was emphasized. Collaborative field trials analogous to autonomous and remote-controlled robotic navigation in lunar environments were also demonstrated on the UMES campus during the summer by visiting UMD students with the support of UMES students.

The three-year UMES-UMD collaborative project titled "DREAM: Developing Robotic Explorations Using Agrobots and Moonbots" funded by the NASA-MSTAR program provides the framework and the funding for the educational and research goals of the endeavors that are in broad alignment with NASA's Artemis mission. Kolb's experiential learning paradigm continues to provide a framework for student learning and engagement as well as continuous refinement and advancement of project objectives. Current efforts in the fall of 2024 will extend and build on the summer efforts.

1.0 Introduction

The UMES-UMD (University of Maryland College Park) three-year collaborative project titled "DREAM: Developing Robotic Explorations using Agrobots and Moonbots" was initiated in the fall of 2023 and provides a challenging platform for education and research aligned with the broad scope of the objectives of NASA's Artemis mission. The work reported in this paper builds upon earlier efforts reported previously[1].

The Robot Operating System (ROS) is an open-source software framework that provides middleware for robotics applications[2,3]. It allows different components (*nodes*) of a robot to communicate with each other using messages sent over a network. The communication happens through a *publishing-subscribing* model where one node sends a message by posting it to a *topic* and other nodes receive the message by *subscribing* to that topic.

One of the challenges that the DREAM project participants have faced is to keep up with the evolution of ROS. Some of the platforms acquired by the UMES project team use ROS 1 while others have adopted ROS 2. The Robot Operating System (ROS) was designed to simplify the development process of various robotics tasks, such as motion planning, perception, and control. It was designed to work with a wide range of hardware. ROS 1, the original version, was released in 2007. It introduced concepts such as message-passing between components, device drivers, and simulation environments. Some of the limitations identified by researchers, especially in areas like real-time performance, security, and scalability led to the development of ROS2 which was released in 2017 and uses modern technologies such as DDS (Data Distribution Service) to address these issues[4].

ROS-based robotic platforms acquired by the DREAM project team for education and research include Agilex Limo and Agilex Scout Mini with a compatible R&D kit developed by Agilex Robotics. Additional robotic platforms such as ROSMASTERX1 and Micro-ROS Pi-5 were acquired from Yahboom. All these platforms use ROS and are equipped with several internal navigational sensors as well as LiDAR and stereo depth cameras. The platforms offer cutting-edge technology for engineering and computer science students, immersing them in artificial intelligence and autonomous navigation applications crucial for lunar and planetary rovers[5], as well as terrestrial utilization[6].

Agilex Limo (Fig 1(a)) and Scout Mini (Fig 2(b)) used in this work are multi-modal mobile robot development platforms that connect to open-source ROS and the Gazebo simulator, granting access to significant reserves of free demos and sample programs for artificial intelligence development, autonomous navigation, mapping, obstacle avoidance, and route planning. A specially designed simulation table (Fig. 1(b)) has also been utilized for simplifying the process of testing model applications.


Fig1(a): LIMO

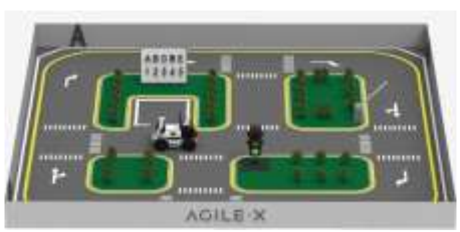

Fig1(b): Simulation Table


Fig1(c):Rosmasterx1


Fig1(d):MicroPi5

UMES robotics lab has also acquired a ROSMASTER X1(Fig 1(c)) a 4-wheel drive mobile robot with a pendulous suspension chassis compatible with ROS. Like Agilex Limo, it uses Jetson NANO and is also equipped with LiDAR and depth cameras. It was used alongside the Limo for obstacle avoidance trials. Both the Limo and ROSMASTER X1 utilize ROS1 MELODIC.

This MicroROS-Pi5 ROS2 car (Fig1(d)) is developed based on Raspberry Pi 5. It consists of a Micro ROS robot expansion board with ESP32 co-processor, 4 encoder motors and tires, LiDAR, and a stereo depth camera. It has adopted ROS2 HUMBLE development environment. Python3 programming used with it allowed importing OpenCV image processing functions and MediaPipe machine learning algorithms.

Fig 2(a): Vertex/Biobot

Fig 2(b):Scout-Mini

Fig 2(c): Field Trial (July 11, 24)

The robustness, versatility, and efficient maneuverability of the Scout Mini platform proved to be extremely useful for preliminary field trials with the UMD-designed VERTEX/Biobot (Fig 2(a)) platform [5]. During the initial field trial in the summer of 2024(Fig2(c)), communication was established between VERTEX and Scout Mini. Additional efforts that will facilitate future advanced maneuvers envisioned for the VERTEX/Biobot with image and LiDAR data of the surroundings provided to the platform by the Scout Mini are reported in a subsequent section of this paper (section 4.0).

Figure (a) in the Appendix provides an overview of how Kolb's Experiential Learning paradigm [7] was envisioned to be adapted for the project. In the following sections, experimentations with the robotic platforms discussed above in laboratory and field settings on the UMES campus will be elaborated. The focus will be largely on efforts undertaken in the summer of 2024. Difficulties and challenges encountered by the students and the learning outcomes will be highlighted. The plans for the future will also be outlined in brief.

2.0 Obstacle Avoidance Trials with Agilex LIMO and ROSMASTER X1

Autonomous navigation through static and dynamic obstacles in unknown environments is one of the key emphases for robotic exploration in lunar as well as other unknown terrains. Mapping efforts using advanced mobile robots equipped with LiDAR, stereo depth camera, and IMU (Inertial Measuring Units) sensors undertaken by the project team were reported in reference[1]. Concepts related to SLAM (Simultaneous Localization and Mapping) and Gmapping integral to ROS-based implementation were outlined. The schematic move_base ( ) package integral to ROS is shown in Figure (b) in the Appendix. The project team devoted significant effort to learning about the move_base ( ) package and how it was utilized for autonomous navigation through obstacles using both simulation and laboratory experimentations with the actual LIMO and ROSMASTER X1 mobile robots delineated below.

A 3m by 3m square boundary was set up in the UMES robotics laboratory to conduct the obstacle avoidance trial using the move_base ( ) package in ROS 1. The move_base ( ) uses a global_costmap and local_costmap as well as a global and local planner in conjunction with sensor data from external sensors such as LiDAR and camera and internal IMU sensors for odometry along with previously generated map data to find an optimal path from start to goal (specified by the user) that the robot autonomously navigates to while avoiding obstacles. Several adjustable parameters including minimum and maximum linear and angular velocities and acceleration, inflation radius, as well as the robot configuration and turning radius play a crucial role and determine the path the robot chooses autonomously. The ROS Wiki page (https://wiki.ros.org/move_base) and other online resources were consulted extensively to execute the trials both in Gazebo simulations and laboratory trials.

Figures 3(a) and 3(b) show two of several configurations that were used in the 3m by 3m square enclosure using buckets and boxes as obstacles that the LIMO robot had to navigate through to the goal location. These configurations were also created in Gazebo where the simulated robot was launched to compare simulation results with actual laboratory trials. Simulations also provided insight into observed behavior in actual runs to inform laboratory trials about parameter settings.

Fig 3(a): Obstacle layout with buckets

Fig 3(b): Obstacle layout with boxes

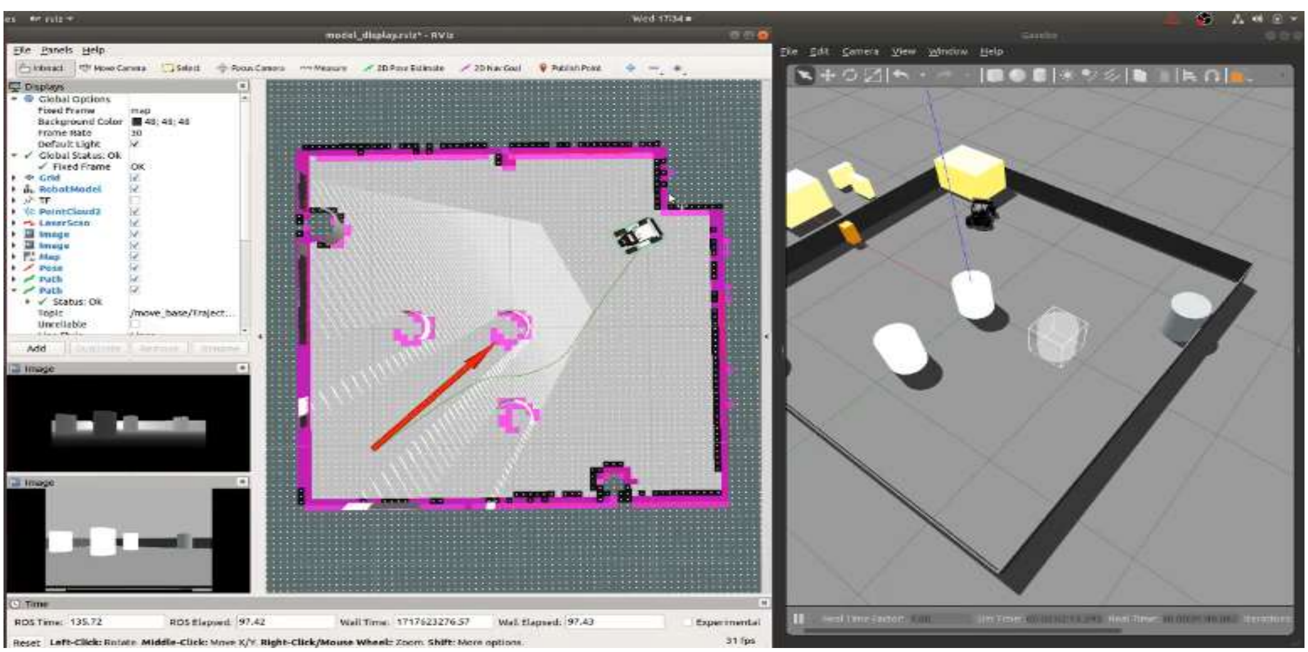

Fig 4: Gazebo simulation of a bucket set-up trial and view of the global path (Goal- back of the red arrow)

Several trials were performed with these set-ups, an example is shown in Figure 4, the back of the red arrow is the goal location. During these trials, the project team adjusted several parameters in the move_base ( ) algorithm. Emphases were largely devoted to adjusting the global and local inflation radii of the obstacles and the commanded forward velocity of the robot. The inflation radius grows the obstacles by a specified amount to avoid collision with the robot. In our setting, in the 3 m by 3 m square enclosure with multiple medium-sized obstacles scattered about, an effective range for global inflation radius was between 0.4m and 0.6m, while an effective range for local inflation radius to complement this was between 0.05m and 0.10m. The maximum and minimum velocity also affected the ability of the LIMO to successfully reach its navigational goal. With higher velocities, the LIMO struggled to stay “on track”. At lower velocities, the LIMO remained closely on track with its global path planner; combined with the proper global and local inflation radius, the LIMO indicated less likelihood of being stuck.

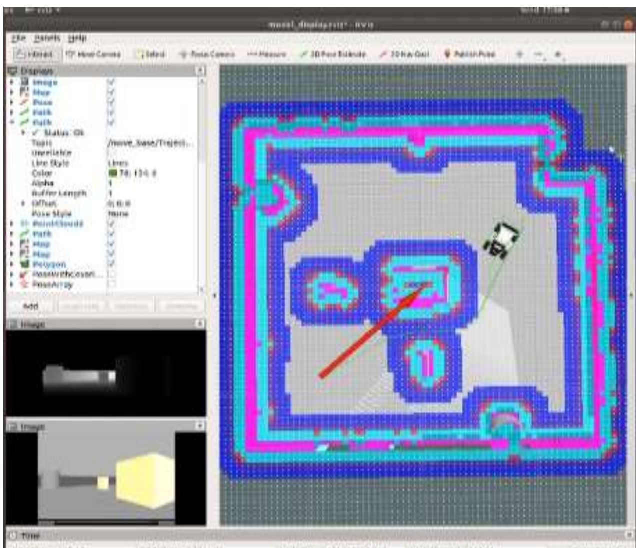

Fig 5(a):Global costmap with a box set-up trial

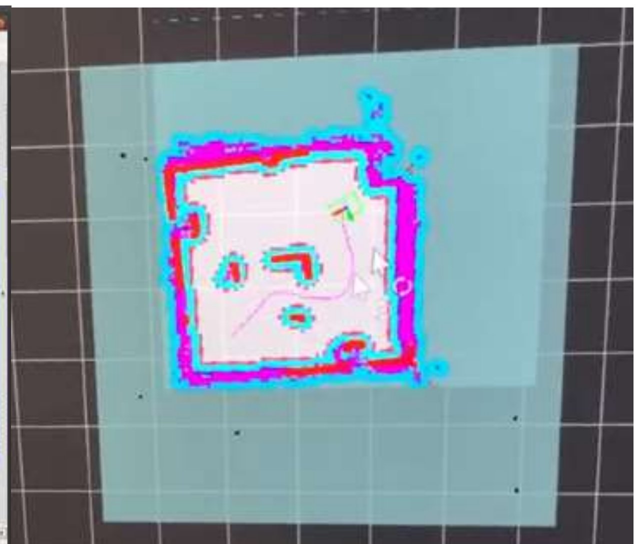

Fig 5(b): Corresponding local costmap

In the trials involving the boxes (Figure 3(b)), in both the physical setting and simulation, the LIMO consistently preferred to "turn left" around the center box and navigate through the middle-left path, rather than taking the middle-right path. This bias can be explained by the costmap_2d calculations within ROS. The cost map uses inflation to propagate cost values outward from occupied cells, with values decreasing as the distance from the obstacle increases. The calculation determined that turning left and taking the shorter path around the box resulted in a lower cost compared to the longer right path, since the left path involves fewer high-cost cells due to the shorter edge of the box, making it a more efficient route. In Figures 5(a) and 5(b), one can observe that in either direction the LIMO approaches from, the shorter side of the box will be on the left, and the smallest cost is incurred from taking the middle-left path. Changing maximum and minimum velocities had identical results with obstacles with boxes as it did with the buckets. Higher velocities caused the LIMO to veer off from its global path plan, which led to critical errors that could result in the LIMO getting stuck, while lower velocities allowed the LIMO to follow its global path closely, reducing the error greatly. Similar efforts were also undertaken with the ROSMASTER X1 robot; in the interest of brevity, they are not included here.

These results underscore the importance of appropriately configuring the global and local inflation radii in the ROS move_base package. Proper tuning of these parameters can enhance the robot's navigation efficiency and obstacle avoidance capabilities, which is crucial for optimizing autonomous navigation in a variety of environments.

3.0 Experiments with the LIMO Robot using the Simulation Table and Accessories

Figure (6a) shown below is the Agilex Simulation table that was also acquired in the early part of summer (2024) to provide students with a structured set-up to use the various sensors and capabilities of the Agilex Limo as well as facilitate comprehension and utilization of the features of ROS (Robotic Operating System). The figure shows the Limo robot, a lift barrier, and a street lamp with red, yellow, and green lights positioned in the “Simulation Table” with trees and other road signs and transportation markers. Under the auspices of the ongoing Summer Exchange Internship Program (SEIP) supported by the Maryland Space Grant, an intern from the University of Maryland College Park (UMD) came to work in the UMES Robotics Laboratory for 10 weeks. He was tasked to learn the fundamentals of ROS, and work in a team setting with a couple of UMES students to explore sensor data processing, localization and mapping, decision-making, and control actuation with the Limo robot utilizing the “Simulation table” and some of its components. Particular emphasis was devoted to developing a preliminary framework for an intelligent automated transportation system with the Limo robot utilizing the lift barrier with an AR (Augmented Reality) tag and a traffic light with red, yellow, and green settings that were integral components of the “Simulation table” (see Figures 6(b) and 6(c)).

The project goal was to have the LIMO robot use move_base( ) to navigate from a select starting point on the “Simulation Table” to a goal point and on the way encounter the lift barrier with AR tag and the street lamp. The LIMO had to stop in front of the barrier and signal to lift the barrier and then proceed forward and turn left to encounter the street lamp, wait till the red light turns green before proceeding towards the designated goal. The students successfully implemented the preliminary objectives as stated. The Limo’s camera with the vendor-provided image processing script, identified the AR tag and activated the associated code to trigger a Bluetooth signal to open the barrier. The ROS framework of publishing to a ‘topic’ (by the Limo) and subscribing to it (by the barrier) was utilized to achieve the intended results. Some tweaking of delay or pause time had to be included to ensure the barrier opened before the Limo proceeded forward.

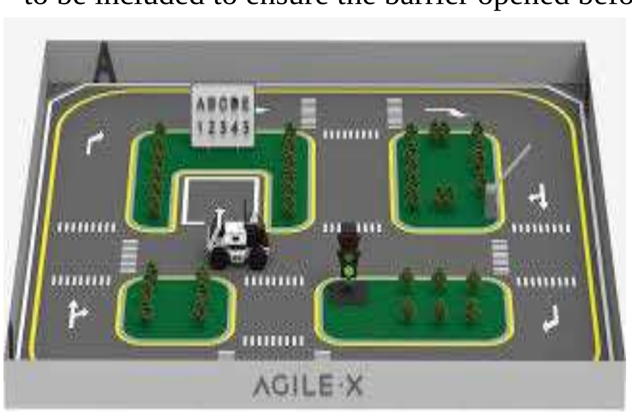

Fig 6(a):Agilex Simulation Table & Accessories

Fig 6(b):Lift-Barrier AR Tag, Fig(c) Street Lamp

After negotiating the barrier, the LIMO moved forward to a wall, turned left, approached the stoplight, and stopped. A separate image processing program allowed the detection of the traffic light and its color. The robot stopped as directed by the move_base command when the light was red. When the light turned green, the robot automatically triggered the move_base command to continue.

Challenges of this trial included inconsistent blue tooth communication delays with the lift barrier automation, errors in traffic light color detection due to variable ambient lighting conditions in the laboratory, and the appropriate parameter settings with the move-base( ) algorithm. The student participants undertook several trials and used an additional 30-second pause to circumvent issues related to Bluetooth communication delays for the lift barrier, adjusted the detection range for the traffic light colors, and selected suitable navigation parameters for the move_base ( ) algorithm to execute successful trial run with the “simulation table” as tasked. Figure 7 displays an example of the outputs of the traffic light color detection algorithm observed from the camera on the Limo robot.

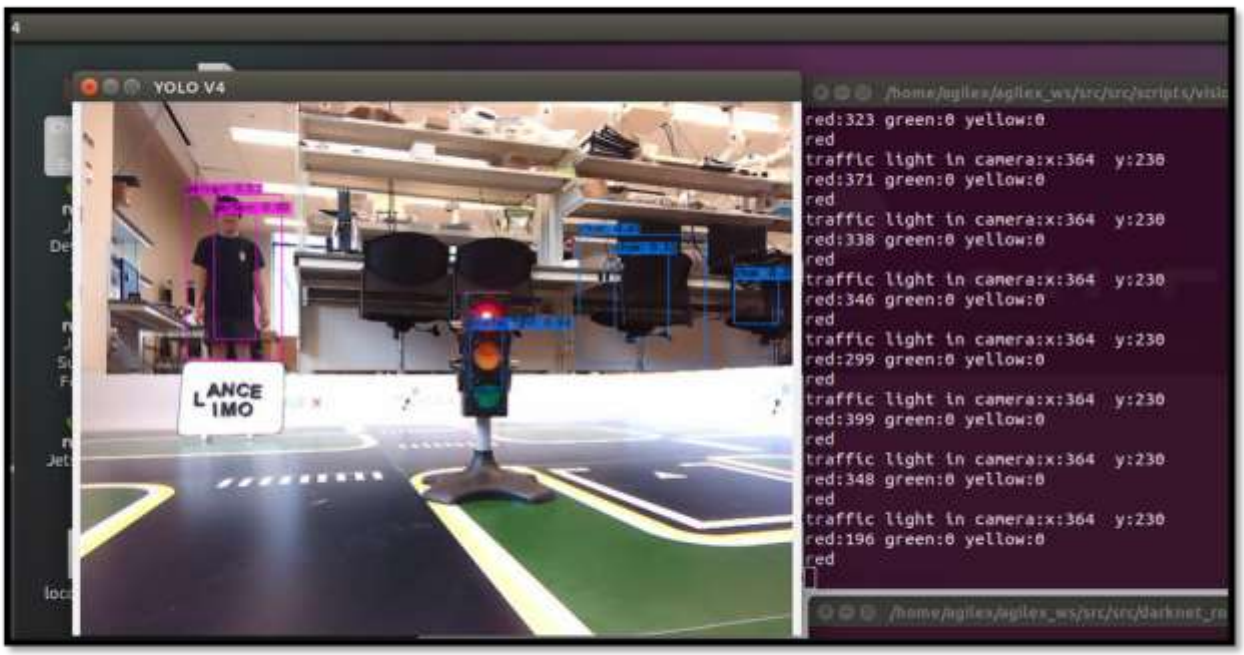

Fig 7: Display related to the Limo Camera running the traffic light color detection algorithm

3.0 ROS2 Humble, OpenCV, and Mediapipe Integrated Trials with MicroROS Pi5

As mentioned earlier the project team has also acquired a few MicroROS Pi 5 educational robots (see Fig 1(d)). The MicroROS Pi 5 robot cannot negotiate rough terrains in the open fields outside the laboratory but provides students with an excellent experiential learning and research platform to explore intelligent navigation capabilities inside the lab. It uses advanced features that integrate ROS2 humble, OpenCV(Open Source Computer Vision Library) functions to process images, and Mediapipe (an open-source framework from Google that helps developers build machine learning (ML) and artificial intelligence (AI) applications). A UMES sophomore student took the lead on working with the platform in cooperation with other project participants as needed. Several demonstration tasks were performed during the summer of 2024 and the fall semester that followed. Line following using visual feedback and PID control as well as intelligent navigation involving face detection and gesture recognition using Mediapipe are outlined below.

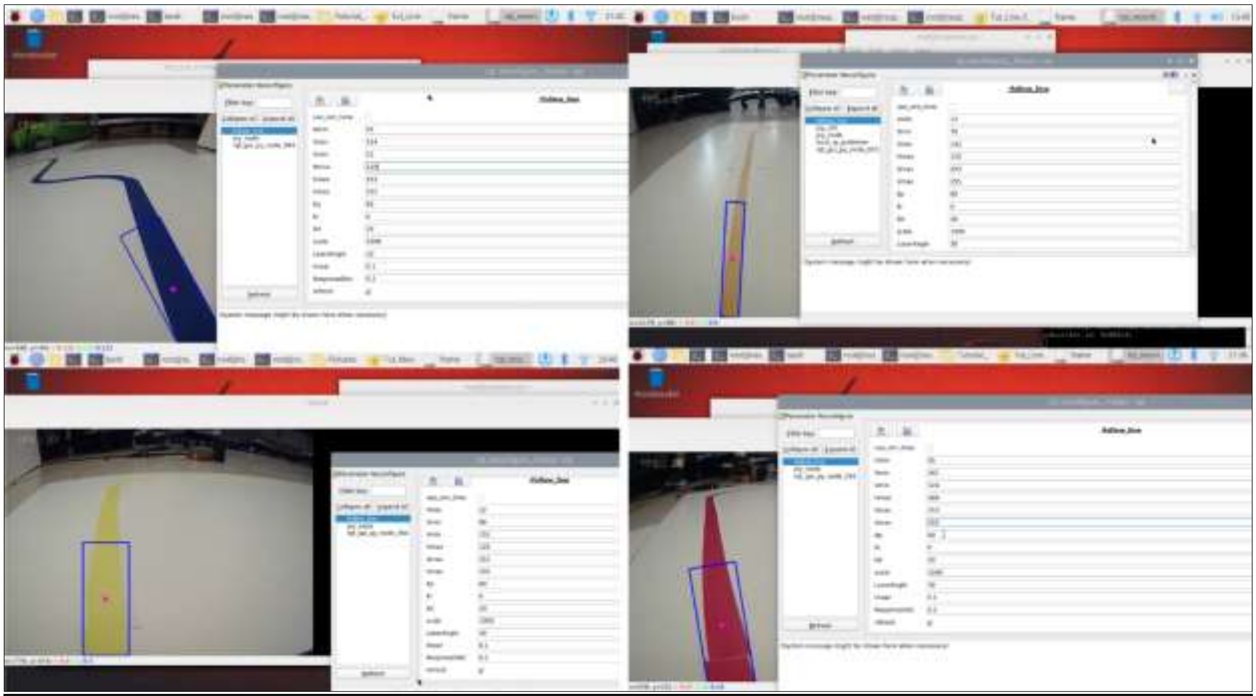

Fig 8: Colored (purple, yellow, orange and red) Line Following Using MicroROS Pi 5

The angle of the camera mounted on the MicroROS Pi 5 robot was adjusted to look ahead at the lines of different colors that the robot needed to follow as shown in Figure 8. The captured image was converted from RGB to HSV (Hue, Saturation, and Value). The "Hue" information allowed the isolation of the colored portion of the applied rectangular mask from the background and obtaining its centroid (ratio first moment and zeroth moment m_{01}/m_{00}; m_{10}/m_{00} [8]) using an existing function in OpenCV. The error could be easily computed from the horizontal offset between the line's centroid and the center of the camera frame (desired) to appropriately determine the angular velocity by adjusting the proportional, integral, and derivative gains while holding the linear velocity constant using the cmd_vel( ) topic in ROS 2. It may be noted that while HSV conversion worked well with colored lines, for a black line it was easier to convert the camera image to grayscale and obtain a binary image with appropriate thresholding to implement the line following.

To demonstrate the face detection capability using OpenCV and Mediapipe, an image of a face (in this case President Obama's face) was printed out and attached to the back of a mobile robot in the lab and moved in an "L" pattern; the MicroROS Pi 5 robot used OpenCV and Mediapipe to detect the face and follow it around using appropriate integration of cmd_vel( ) and ROS 2. OpenCV captured and preprocessed the video frames of the image of the face captured by the camera on the MicroROS Pi 5 robot, MediaPipe results are drawn on the frame and the bounding box of the detected face provides the position of the face in the frame (Figure 9a). Based on this commands are sent to the cmd_vel( ) topic of ROS 2 to control the MicroROS Pi 5 robot's movement to follow the robot with the face attached to it.

Fig 9a:Face Detection(left) and Hand Gesture (right) Recognition Using Media Pipe and OpenCV

Mediapipe uses 21 landmarks including wrist, finger joints, and tips to process and recognize hand gestures. The video input is processed through OpenCV for hand gestures made in front of the camera mounted on the MicroROS Pi 5 robot. Hand gestures were sorted out with the 21 landmarks in Mediapipe to recognize gestures and integrate with motion commands in ROS 2 for the MicroROS Pi 5. For preliminary trials, if all five fingers were raised the robot moved forward whereas if five fingers pointed down the robot moved backward, and if the hand was closed in the form of a fist it stopped. Additionally, if the fingers were formed to make an 'OK' gesture as shown in Figure 9b, the robot recognized the gesture and was programmed to move in a circle.

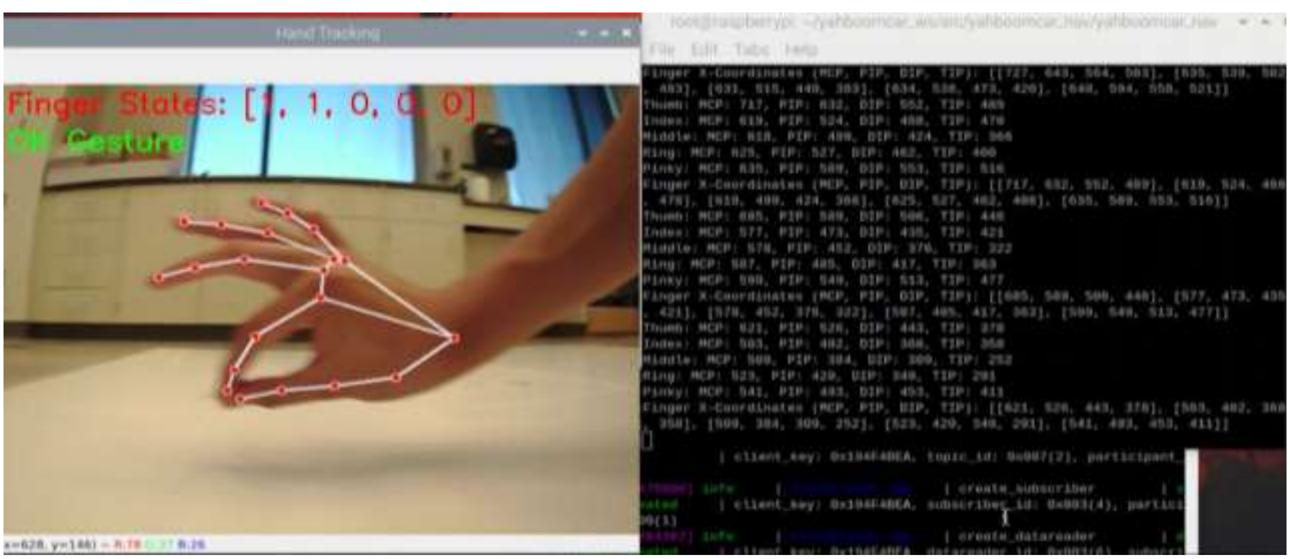

Fig 9b: Hand Gesture ('OK') Recognition Using Media Pipe and OpenCV

Additional demonstration efforts have also been performed with QR code detection but are not included here for brevity.

4.0 UMES-UMD Collaborative Field Trial with VERTEX, Scout-Mini, and other MoonBot Rovers

A field trial was conducted in the late summer of 2024 on the UMES campus with the VERTEX and other MoonBot platforms that have been designed and built at UMD [9](see Figure 10). The Agilex Scout Mini and Limo robots acquired by the UMES project team are also shown in the photograph in Figure 10 in front of the UMD robots. Besides demonstrating the capabilities of each of the robotic platforms, a collaborative effort has been initiated to implement the following objectives :

- To establish a reliable TCP connection between UMES's robotics platforms and UMD's VERTEX bio-bot.
- To stream real-time image data from UMES's Scout Mini to VERTEX using ROS nodes written in Python and C++.
- To develop and test a ROS2 client node for UMES's Scout Mini that can handle various data streaming commands, and to develop a ROS1 server node for UMD’s VERTEX that can receive and deserialize incoming data streams.
- To familiarize UMD and UMES with each other's robotic systems and their capabilities.

Fig 10:VERTEX (back), Scout Mini (2nd from front), Limo(front) and other MoonBots

- To lay the groundwork for future data exchanges, including point cloud terrain data, map data, navigational commands, and location data, from multiple UMES robots to VERTEX in a peer-to-peer network.

To this end, a ROS2 Python client node was successfully developed for the UMES Scout Mini. This node was designed to subscribe to an RGB image data topic */color/image_raw* and a /*commands* topic. The /*commands* topic accepted arguments inserted to the console manually by an additional node to start the data stream *START_STREAM*, stop the data stream *STOP_STREAM*, or send just a snapshot of the data stream *SEND_SNIPPET*, allowing for flexible control over data transmission by the user of the Scout Mini. Image data was converted from ROS .msg format to JSON format, the portion that was real image data was encoded from binary to string format, and the entire JSON object was serialized, and sent over a TCP connection in 1024-byte chunks. The server node on VERTEX captured this data and published it to a special topic for further processing. A secondary node on VERTEX successfully deserialized the incoming data back into JSON format, decoded the portion that was image data back into binary form, and converted the entire JSON object back into the ROS .msg type for real-time publication to a new image data topic, in a format (.msg) that is compatible with the rest of the VERTEX's ROS ecosystem. These capabilities are being refined for future field testing.

5.0 Learning Outcomes

Learning can be categorized into developing skills in three broad domains - cognitive, affective, and psychomotor. Higher education typically focuses largely on the cognitive domain following Bloom's taxonomy – knowledge, comprehension, application, analysis, synthesis, and evaluation [10]. ABET outcomes for engineering education integrate developing student abilities in both the affective and cognitive domains [11].

Recent advancements in robotics, AI, and sensor technologies for monitoring and control have transformed the field. These developments are not only shaping the future of technology but also providing platforms for enriching the student learning experiences. At UMES, the DREAM project has allowed participating students to engage with cutting-edge tools like Gazebo and ROS open-source software platforms in the Ubuntu/Linux operating system and advanced robotic devices equipped with LiDAR, cameras, and various odometry sensors. Participating students have learned to use ROS packages written in Python and C++ for autonomous navigation and other applications. Several advanced concepts including SLAM (Simultaneous Localization and Mapping) and AMCL (Adaptive Monte Carlo Localization) were successfully utilized in intelligent navigation trials with mobile robotic devices. They also got valuable to exposure to OpenCV( ) functions for image analysis as well as, machine learning and AI using Mediapipe. OpenCV and Mediapipe algorithms were utilized for face detection, gesture recognition, and QR code recognition. Using both grayscale thresholding and HSV color filtering, the students implemented line-following algorithms. In the process, they also got exposure to control systems and PID gain tuning for differential steering. The hands-on learning and real-time application development endeavors have provided graduate students with thesis topics and undergraduate students with enriching university experiences that are looked upon favorably by employers. Some efforts have been made to integrate aspects of the project with undergraduate courses in instrumentation and control systems; also planning and discussions are underway to develop a mobile robotics-centered course that will integrate aspects of the project fundamentals and implementation framework. The higher-order synthesizing skills of the students were also brought to bear with the open-ended creative project with the robotic systems and helped reinforce the ABET learning outcomes related to teamwork, analyzing and interpreting data, and self-directed acquisition of new knowledge[11].

6.0 Future Work

The student efforts outlined in this paper offer considerable potential for advancement within the context of the experiential learning and research framework. The "concrete experiences" described have facilitated "reflective observation" and "abstract conceptualization," which are now paving the way for further "active experimentation," in alignment with Kolb's experiential learning model[7].

Moreover, the project team has recently acquired a legged robot to complement navigational challenges that wheeled platforms may find insurmountable. The legged robot is equipped with 3-D LiDAR, a stereo depth camera, odometry sensors, and a variety of other instrumentation for intelligent navigation. It can also be mounted with a multi-degree-of-freedom robotic arm for combining pick and place operations with intelligent navigation. Significant effort will be devoted by the project team to develop relevant applications aligned with lunar navigation by both the wheeled and legged robotic platforms.

7.0 Acknowledgement

The authors would like to acknowledge the NASA MSTAR Grant # 80NSSC23M0206 titled "DREAM: Developing Robotic Explorations with AgroBots and MoonBots" and the AIRSPACES (Autonomous Instrumented Robotic Sensory Platforms to Advance Creativity and Engage Students) project supported by Maryland Space Grant Consortium.

Bibliography

[1] Nagchaudhuri, A., et al., "Experiential Learning with Mobile Robots: Bridging Physical and Virtual Environments", Proceedings of American Society for Engineering Education's Annual Conference, June 23-26, Portland Oregon.

[2] Quigley, M., et al, "ROS: an open-source Robot Operating System", ICRA Workshop on Open Source Software (2009)

[3] Guimarães, R.L., de Oliveira, A.S., Fabro, J.A., Becker, T., Brenner, V.A. (2016). ROS Navigation: Concepts and Tutorial. In: Koubaa, A. (eds) Robot Operating System (ROS). Studies in Computational Intelligence, vol 625. Springer, Cham. https://doi.org/10.1007/978-3-319-26054-9_6

[4] Macenski, Steve & Foote, Tully & Gerkey, Brian & Lalancette, Chris & Woodall, William. (2022). Robot Operating System 2: Design, Architecture, and Uses In The Wild. 10.48550/arXiv.2211.07752.

[5] Kraft, M., et al, "Integration of Heterogeneous Computational Platform-Based, AI-Capable Planetary Rover Using ROS 2," IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium, Pasadena, CA, USA, 2023, pp. 2014-2017

[6] Baek, E.-T.; Im, D.-Y. ROS-Based Unmanned Mobile Robot Platform for Agriculture. *Appl. Sci.* **2022**, *12*, 4335. https://doi.org/10.3390/app12094335

[7] Kolb, D. A. (1984). Experiential learning: Experience as the source of learning and development (Vol. 1). Englewood Cliffs, NJ: Prentice-Hall.

[8] Nagchaudhuri, A., Kuruganty, S., and Shakur, A., "Introduction to Mechatronics using an Industrial SCARA Robot and Integrated Vision System", Mechatronics (Journal, Elsevier Science Ltd.)Vol. 12, 2002, pp. 183-193.

[9] Akin, D., C. Hanner, N. Bolatto, D. Gribok, and Z. Lachance, "Development of an EVA Assistance Roving Vehicle for Artemis and Beyond" 50th International Conference on Environmental Systems, ICES-2021-101.pdf (1.09 MB)

[10] Bloom, B.S., (1984). Taxonomy of Educational Objectives. 1. Cognitive Domain. New York, Longman

[11] ABET, "Criteria for Accrediting Engineering Programs, 2020 – 2021" Available Online: https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2020-2021/

APPENDIX

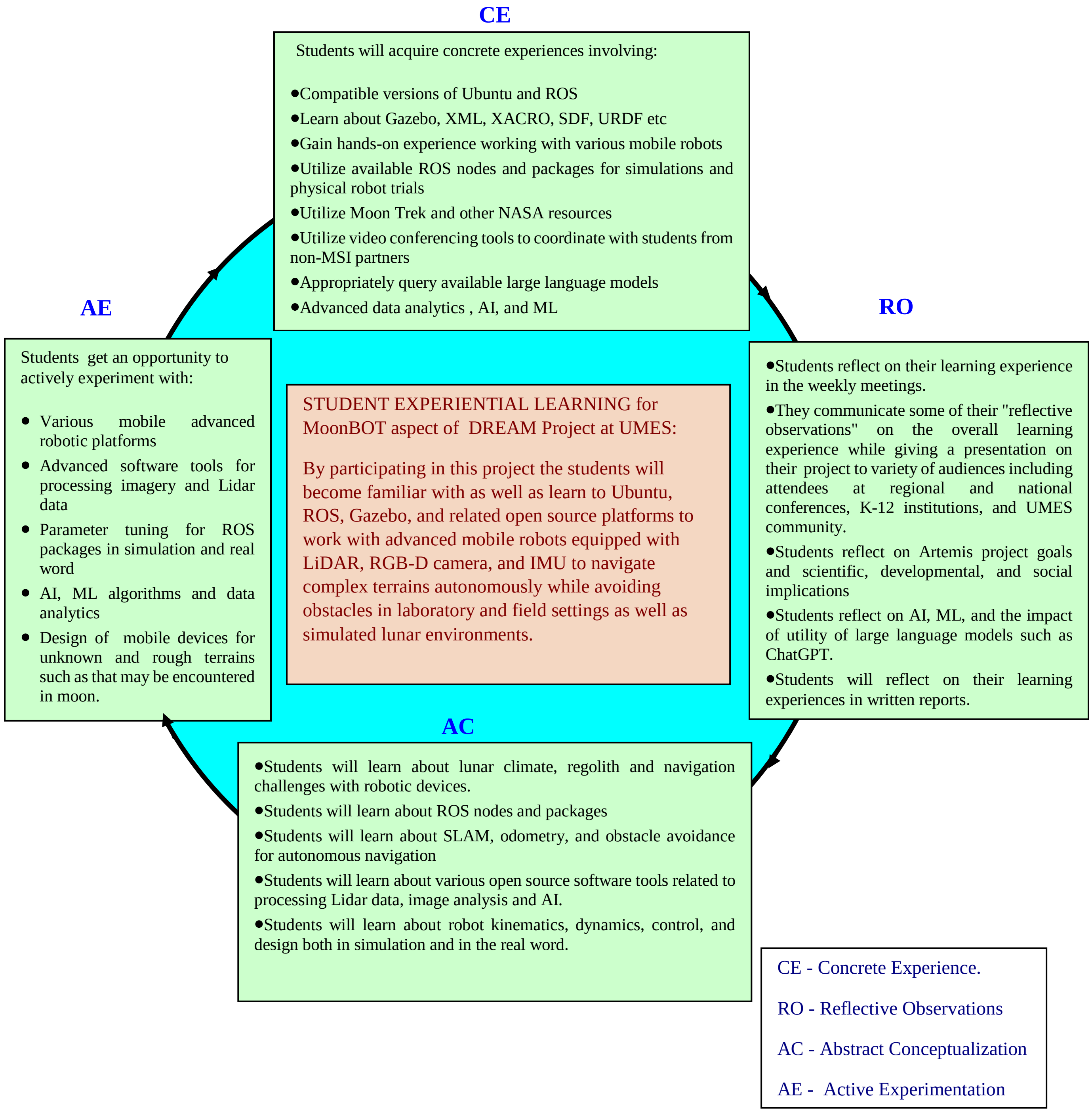

Figure a: Kolb's Experiential Learning Cycle adapted for the Student Experiential Learning Activity in the MoonBot aspect of the DREAM Project

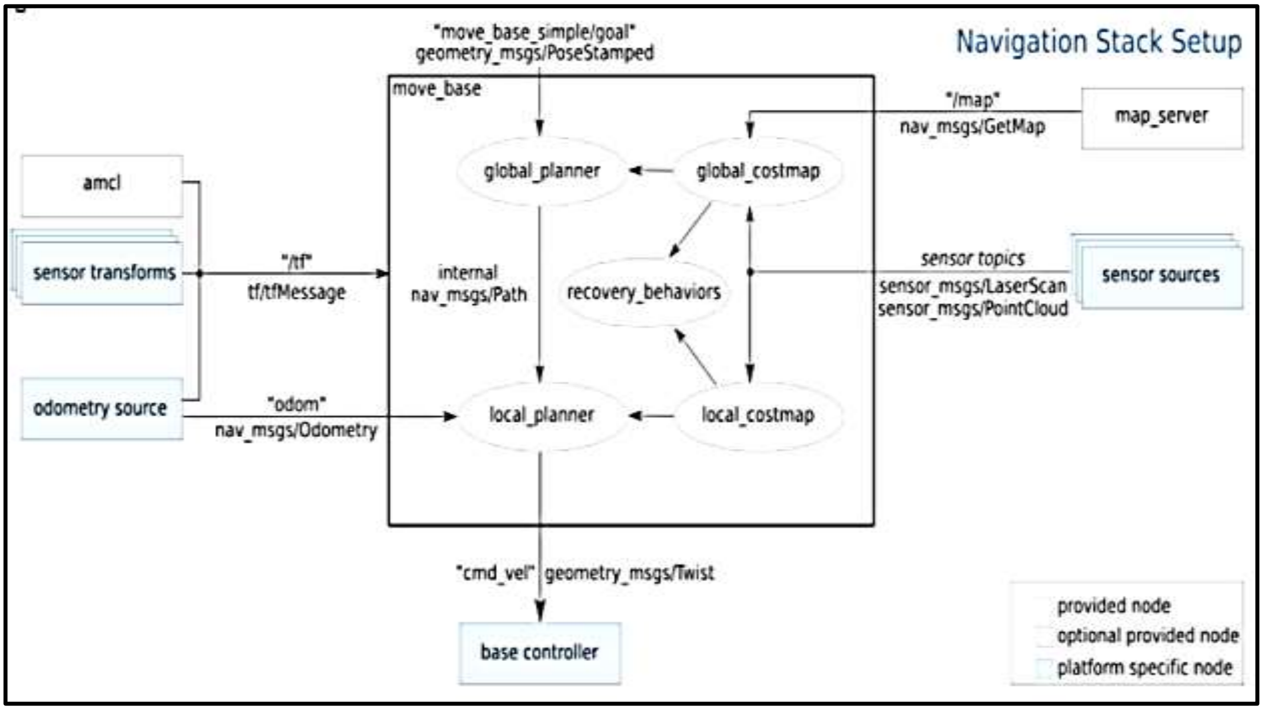

Figure (b) : Move_base Package (https://wiki.ros.org/move_base)