ISBN: 978-628-96613-3-0. ISSN: 2414-6390. Digital Object Identifier: https://dx.doi.org/10.18687/LACCEI2026.1.1.2800

A Modular Mobile Platform with Dual-Camera and LiDAR-based Collision Avoidance Software for Autonomous Robotics Competitions

Autumn Peterson[1]; Michael Stalford[1]; Gianni Dragos[1]; Abigail Clerget[1]; Gabriela Strevay[1]; Molly Ruley[1]
Faculty Mentors: Dr. Luis Felipe Zapata-Rivera[1] and Dr. Joel Schipper[1]
[1]Embry-Riddle Aeronautical University, AZ, USA, *petera56@my.erau.edu, stalform@my.erau.edu, dragosg@my.erau.edu, clergeta@my.erau.edu, stregayg@my.erau.edu, ruleym@my.erau.edu*

***Abstract*– This paper describes the design and realization of a mobile platform for autonomous vehicles to participate in robotics design competitions, such as the Intelligent Ground Vehicle Competition (IGVC). The goal of constructing this platform is to develop future designs of similar platforms and to provide modular software and hardware structures for re-use and adaptability. The software system is loaded onto a Jetson Orin Nano Super and utilizes the Humble distribution of Robot Operating System 2 (ROS 2). It separates each core processing task into a threaded node where data topics are published and/or subscribed to as variable arguments for processing. New nodes can be integrated into the system by linking them to published data topics. In this implementation, sensor integration, data processing, and waypoint planning are all separated into discrete nodes. The mechanical and hardware systems of the robot include a flexible power distribution system using adjustable voltage regulators to support a wide family of use cases. Additionally, an emergency-stop system is built into the frame of the robot and includes a button interface on the robot and a remote transmitter to stop the system at a distance. The frame is an aluminum construction that sits atop a motorized wheelchair base and is connected via a mast that is inserted into a pole in that wheelchair base. This allows for quick changes to the fundamental design structure to be made. A dual-motor differential drive provides the robot with the ability to rotate in place and the ability to perform routine maneuvers necessary to complete navigation tasks. The design is mostly effective in navigation through an obstacle course, especially during bright and even midday lighting. The obstacle avoidance system is robust and consistently identifies on-course obstacles and maneuvers around them. Future improvements may include a front-facing stereo-vision camera, motor encoders, and more intentional weatherproofing. Keywords-- autonomous vehicle, obstacle recognition, collision avoidance, development platform, robotics, modular.**

I. Introduction

This project aims to develop a modular and cost-effective software and hardware development platform for the construction of robotic systems to participate in autonomous vehicle competitions. In designing a development structure that is easy to re-use and expand upon, participation in this emerging field is made more accessible for students and researchers.

The Intelligent Ground Vehicle Competition (IGVC) served as a baseline ruleset in developing the robot. IGVC is a yearly-held competition sponsored by RoboNation and hosted by Oakland University in Rochester, MI. The competition offers two courses: autonomous and self-drive [1]. The autonomous section of the competition places the robot—of medium size, being no smaller than two by three feet in the horizontal directions—into an obstacle course that includes barrels, potholes, lanes, and a ramp. The robot uses sensors, image processing, and a drive system to navigate the course. This project derives its requirements for software and physical design from the IGVC specifications for the autonomous course. While other competitions may differ somewhat in the requirements placed upon the participating teams, the flexibility of the solution helps ensure that it can be modified easily to meet these alternative demands.

II. Competition Overview

Based on reports from past events and information provided by the IGVC hosts, a competition course outline was constructed [1]. Other than an instruction to begin navigation, the robot is expected to navigate the course autonomously. The course design is described below, and an example course is illustrated in Fig. 1.

The course must be completed within six minutes. The track's boundary lanes are spaced at a minimum of ten and maximum of twenty feet apart, between which the robot must always remain. The total distance a competing robot can expect to travel is approximately five-hundred feet for a given run. The outer lanes may be solid or dashed—the system must react to both accordingly. Construction barrels (approximately thirty pounds and forty inches tall) will be placed in various positions along the course to act as vertical obstacles for the robot to avoid. Additionally, "potholes" (black circles outlined in white) will appear along the course with a maximum diameter of two feet. The robot's speed must be at least one mile-per-hour within the first forty-four feet (or first thirty seconds from the start). The robot must traverse a "No Man's Land," where there are no lane lines for navigation. The area is denoted only by barrels and contains a ramp with an incline no steeper than 15-grade (or 8.5 degrees from level). Waypoints are given for both the entrance and exit of No Man's Land as well as for the start and end of the ramp. The course is on asphalt in an empty parking lot.

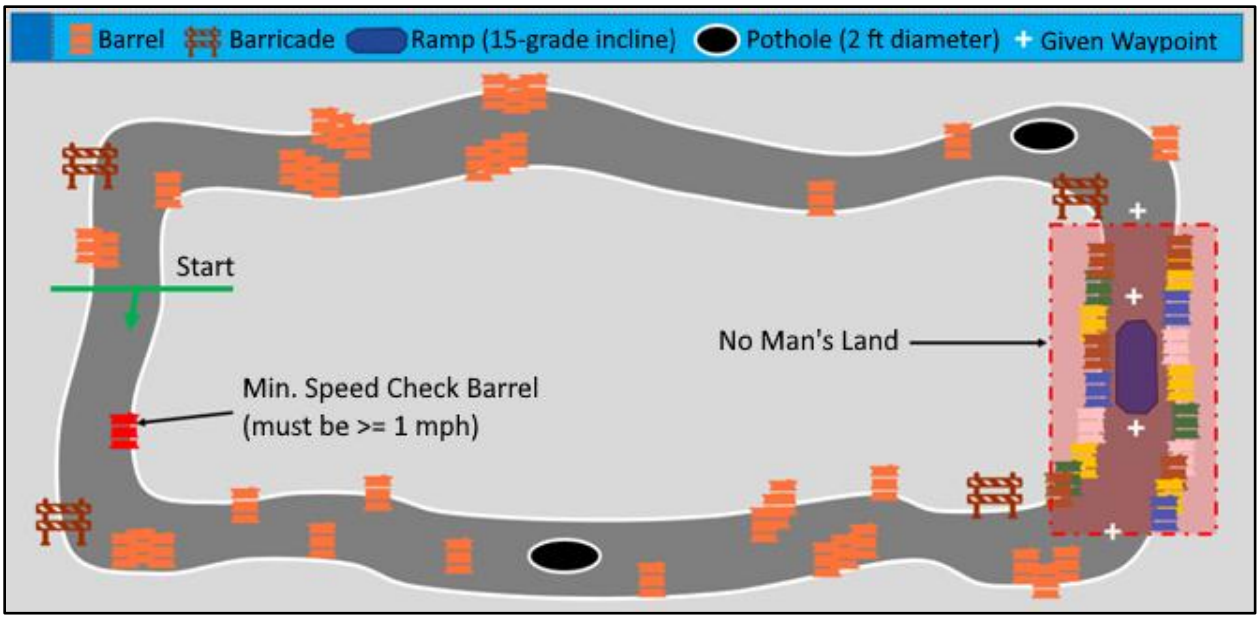

Figure 1 Course Overview Created from Competition Description [1]

IGVC provides additional events, including a roadway-like self-driving competition. This platform is capable of supporting development to participate in more advanced programs like this with some work and minimal integration.

III. System Requirements

Prior to constructing the robot, three different types of requirements were identified to aid in the robot's development. The requirement categories are *course requirements*, which describe things expected of the robot during the competition; *hardware requirements*, which describe physical limitations placed upon the robot based on the design and course; and *software requirements*, which outline the software architecture and are also sourced from design and course. These represent the minimal solution that must be achieved to have a working product. Additional aspects of the project ultimately derive from these, including modularity of the software system and flexibility of the hardware power distribution. Many requirements (like the safety system and payload) are directly translatable to other autonomous navigation competitions. There may be overlap across categories. These have been grouped and outlined below:

A. Course Requirements

1) *AutoNav Course:* Six minutes will be allowed for course navigation. Vehicle must complete one lap around the course. Vehicle must reach 1 mph by the speed check barrel. After the speed check barrel, the vehicle must average more than 1 mph for the entire course and cannot exceed 5 mph.
2) *Traffic Flow:* The vehicle cannot stop on the course for over 1 minute. Course boundary lines cannot be crossed by the vehicle. If more than 60 seconds have passed when the robot reaches the 85 feet speed check barrel, the vehicle cannot keep up with traffic and is not viable.
3) *Payload:* The vehicle must carry a 20 lb. payload that is 8x8x16 in. This is comparable to a delivery parcel or additional components needed for an advanced design.

B. Hardware Requirements

1) *Vehicle Configuration:* The vehicle must be propelled by direct mechanical contact with the ground. Vehicle power must be generated on-board.
2) *Safety System:* Both a local and remote hardware-based emergency-stop system must be created that is not tied to the obstacle avoidance software system. The local button must be visible, accessible, and red. The remote button must work within 100 ft. The system must shut down if the remote E-stop is not active or is out of range. A safety light will blink patterns to inform of status of robot (autonomous, on, off, etc.)

C. Software Requirements

1) *Navigation:* The vehicle must demonstrate it can follow waypoints. Vehicle must demonstrate that it can follow parallel lane lines and stay between them. Lane lines may be solid or dashed and may fluctuate between 10 ft and 20 ft apart.

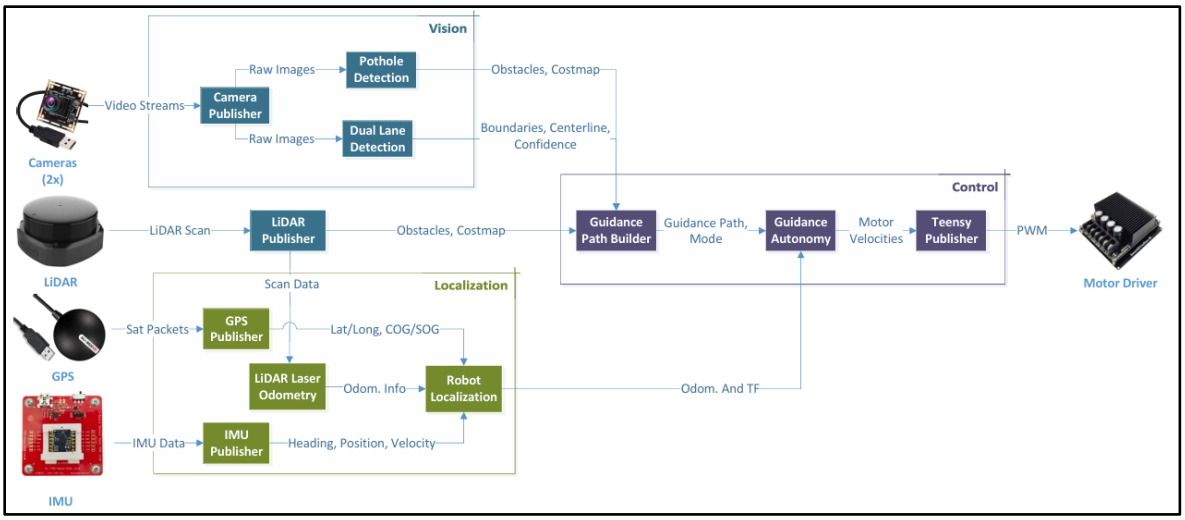

Figure 2 Diagram of Software System

2) *Obstacle Clearance:* The vehicle must demonstrate it can detect and avoid obstacles. Vehicle must prove it can find a path to a single two-meter navigation waypoint around an obstacle. Obstacles the vehicle must avoid include potholes, lane lines, and barrels. Obstacle barrels will not be shorter than 40 inches.
3) *Method:* The vehicle cannot map out the course for future navigation paths. No prior knowledge of the specific course structure (other than the waypoints by the ramp) shall be programmed into the robot. Obstacle avoidance decisions must be made on-the-fly.

IV. System Design

A. Software System

The obstacle avoidance and navigation processing occurs on a Jetson Orin Nano Super (or "Jetson Nano") [2]. The Jetson Nano best aligns with the project requirements for IGVC and has the support for the computer vision computation needed for autonomous vehicle navigation. This software platform acts as the main computing unit for the robot: it has sufficient computational power that meets the perception requirements outline in the IGVC 2026 ruleset. The Jetson Orin Nano has superior hardware specifications for the CPU, GPU, and memory. It is also designed specifically for AI capability and has multiple inherent resources such as JetPack SDK and neural network support through TensorRT CUDA, cdDNN, DeepStream, and multimodal AI acceleration. These strengths make it ideal for more advanced use cases in the future.

The Jetson Nano runs an installation of Robot Operation Software, or ROS, which is a middleware framework that defines software architecture, communication patterns, and runtime environment for the robot's software system [3]. The robot runs on the ROS 2 Humble distribution [4] due to its high support and widespread usage among many autonomous robotics projects.

The robot's software and control stack is comprised of ROS nodes each covering an operation that the robot will use for autonomous navigation. Each ROS node operates on a publisher-subscriber framework where data is organized, sent, and received in channels called ROS topics [5]. ROS nodes can be compiled and added to the system at-will due to the nature of the system architecture. Figure 2 the software architecture diagram shows the design of the software system. The components pictured in the diagram are described in the hardware section. Coloured blocks represent ROS nodes and labelled arrows represent topics. Sensors used in data collection and outputs of the system are shown in the diagram as well.

Each sensor in Fig. 2 has an associated publisher. These publisher nodes collect information from the sensor, format it for use by the system, and publish it as a topic. Then, data-processing nodes subscribe to these topics. Note that the camera publishers interpret data for pothole and lane detection and pass results (position of obstacles, anticipated centerline, etc.) to a guidance node. The GPS and IMU are grouped into a localization process, which informs the robot on its position. This is used with the information from the guidance node to form a path forward, which is passed to the motor controller and drivers. The LiDAR accomplishes two purposes: it provides horizontal scan data to identify vertical obstacles and gives the robot odometry information based on the change in position of scanned objects. Additional nodes can be inserted into the system easily by subscribing to the relevant data topics and performing whatever analysis is necessary on the data.

B. Electronics System

The vehicle includes several electronic components for control, processing, and sensing. To reduce noise in the control

and software systems and prevent accidental shutdown, the motor circuit (discussed in the mechanical section below) is kept electronically isolated from the processing and control circuit (discussed here). The components of the software and control systems are summarized in Table 1. Components that have voltages denoted by a dash are powered from the Jetson Orin Nano Super as peripherals. Also, note that the Cytron MDDS30 listed in the table is used in motor driving and uses a separate battery supply than the other components.

Table 1. Components by Purpose and Supply Voltage.

Component	Purpose	Supply
Jetson Orin Nano Super [2]	Run Software	19 V
KLT-USB-0362 V2 [6]	Comp. Vision	-
Slamtec S2L [7]	LiDAR	5 V
BU-353-S4 [8]	GPS	-
Yost 3-Space v1.0 Nano [9]	IMU	-
Teensy 4.1 [10]	Motor Control	-
Cytron MDDS30 [11]	Motor Drivers	24 V
Arduino MKR WAN 1310 [12]	E-Stop TX/RX	3.7 V

As these components have a wide range of operating voltages, an array of step-down converters is used to power them. The power source used to step-down the voltage is isolated from the motor source. These regulators are adjustable and can be configured in future designs to accommodate other components with differing power requirements.

C. Mechanical System

The robot's structure is built around a motorized wheelchair base. The low-speed, high torque design and weight capacity considerations of the construction translate directly to a competition use case where speed is low and weight is important. Further, many motorized wheelchair bases have a central mounting pole in which a post is inserted to seat a chair: this allows for flexible alterations by changing out the inserted construction placed into the mounting pole. See Fig. 3 for an example of a pole-mount motorized wheelchair. Aluminium is a good candidate for this frame extension as it is inexpensive, light, and easily formed into sheets, corners, and extrusions. A competition-specific design is then built atop the aluminium frame extension and can be tiered and supported about the central post for proper weight distribution.

Since unique competition designs can vary widely depending on the specifications put forth by the organizers, this structure allows the width and height of the system to be changed at-will to accommodate a large family of use cases. Coupled with the modularity of the ROS-based software system and step-down supplies for electronics, the platform can be readily altered to support a different solution. Other than the motors, every aspect of the system is easily modified.

Figure 3. Example Motorized Wheelchair Base [13].

V. Implementation

A. Software System

A Linux environment for software development was chosen for the autonomous vehicle. Ubuntu version 22.04 was selected due to its optimal support for the Humble distribution of ROS2 [4], which acts as the middleware framework for the modular software system. NAV2 [16] was chosen for the main navigation, localization, and control stack for the robot. NAV2's vast library of diverse controllers and planners provides a rich environment to swap in different navigation systems that can be altered for chosen use case. Gazebo Harmonic [15] was chosen for simulation and virtual testing of the software control stack. Multiple packages could be created and tested in the same Gazebo world for direct performance comparison. Different worlds were also created for testing separate portions of the robot's software.

The three pillars of our autonomous system design are vision, localization, and control. The robot's vision includes lane-following and obstacle detection, which are handled by a camera and LiDAR sensor system. Modular software packages for both single and dual camera setups were created to satisfy the lane-following and pothole detection requirements. Gazebo testing was conducted to weigh the costs and benefits of a single versus dual camera setup. The dual camera setup was ultimately chosen for the larger field-of-view obtained by designating a left and right camera to search for each lane. The requirements for our robot's course had the lanes' width vary significantly between 10-20 feet, so a wider field of view was necessary to establish the boundaries and centerline the robot uses for navigation. However, other use cases may not have varying lane widths, and a single camera could be the optimal choice, directly cutting down on computation cost and complexity of the lane detection algorithm. The robot's localization and odometry is governed by sensor fusion between the GPS, IMU, and LiDAR due to the current platform not having wheel encoders. However, the modularity of the autonomous robot has the capability to add on wheel encoders to improve the accuracy of the robot's odometry. Different methods for localization can be tested in the Gazebo environment to find the optimal method of sensor fusion and what components are

necessary to achieve the level of accuracy needed for a particular competition's requirements.

B. Electronics System

The electronics were assembled around the Jetson Orin Nano Super, which performs all software computational tasks. The two cameras used in pothole and lane detection, the Teensy 4.1, the IMU, and the GPS all receive power from the Jetson Nano and communicate data directly to it via USB connections. The other components were deemed to be too large of a drain on the Jetson Nano's power system and are instead supplied via the step-down voltage regulator. The Teensy 4.1 board handles real-time motor control. The Teensy 4.1 receives PWM and direction signals calculated from command velocities from the Jetson Nano via serial communication. The PWM and direction signals are then sent to the Cytron MDDS30 motor driver. An overview of this electrical system is given in Fig. 4.

Electrical isolation was implemented between the motor power system and the control electronics to minimize noise and ensure system stability. The 24 V motor supply powers only the drive motors and is isolated from sensitive electronics. The remaining components are powered by a separate supply derived from three 7.2 V batteries connected in series and regulated through step-down converters.

To mitigate the effects of voltage transients caused by high motor current draw, the power distribution is structured such that disturbances on the motor line do not propagate to the control electronics. This prevents momentary voltage drops from affecting the Jetson Nano and associated decision-making processes.

To prevent damage due to motor stall conditions, circuit breakers are incorporated to interrupt power during sustained overcurrent events. Additional fast-acting protection is used to safeguard sensitive electronics.

As an additional safety layer, an emergency-stop (E-stop) system provides both local and remote shutdown capability. In the local configuration, activation of the E-stop directly interrupts the motor power line, de-energizing the drive system. This ensures a fail-safe condition where any loss of control or system fault results in immediate cessation of motion. The remote system, which uses the wireless communication protocol LoRa via the Arduino MKR WAN boards, breaks that same control line by removing power to a different engaged relay. In this way, any failure of the system (receiver does not activate, control line is cut) will stop the motors. Additionally, the remote node communicates with the receiver on the robot to shut the system down once it is out of range.

The electronic components are supplied by a 15 Ah battery array (passed through the step-down system) and the motors are supplied by a pair of 50 Ah batteries (for a total of 100 Ah). Repeated testing of the system over consecutive days for several hours showed minimal power loss. As robotics competitions usually require the robot to run for less than ten minutes over a course and allow for recharging of components, these specifications are well within what is expected for this use case. It is worth noting that a more capable solution that carries or hauls a heavier payload will pull more current from the motor batteries, and a more robust navigation system that performs more analysis will pull more current from the electronics batteries. In either case, the batteries currently in use will still be powerful enough to supply the system, based on the aforementioned course testing.

C. Mechanical System

The mechanical platform, as shown in Figure 5, was constructed using a Pride Jazzy motorized wheelchair base, providing an integrated drivetrain with sufficient torque and stability for the competition environment [13]. This is the same structure pictured in Fig. 3. This eliminated the need to design a drivetrain from scratch and allowed the team to focus on sensor integration and control, acting as the foundation to house motors and motor batteries. It also has a central mounting pole.

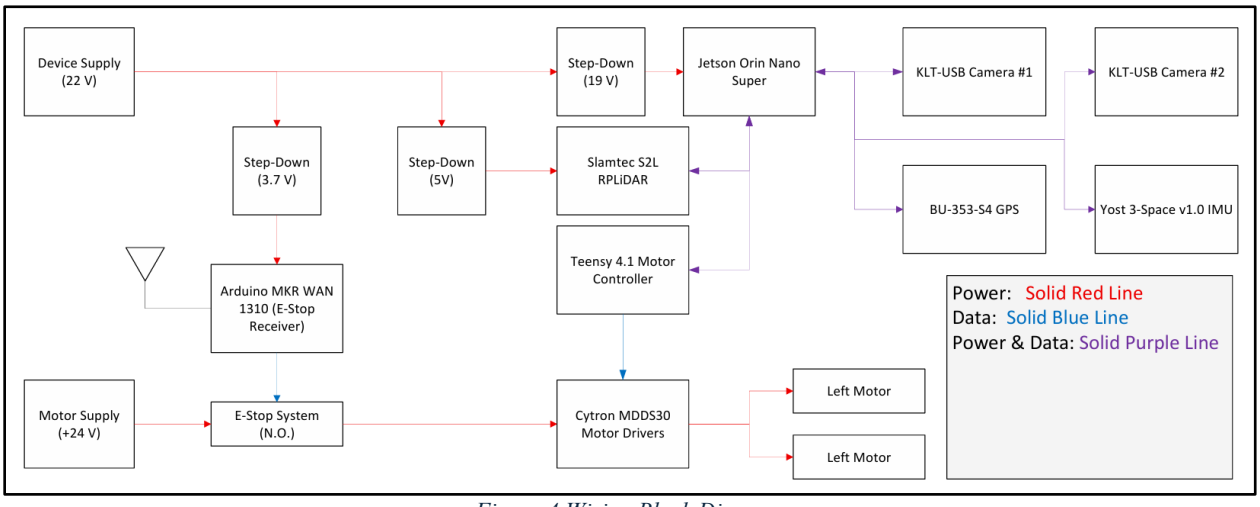

Figure 4 Wiring Block Diagram

An aluminium frame was mounted to the base using that central mounting pole. This avoids major modifications to the motorized base, preserving structural integrity and increase ease of disassembly. An aluminium frame was constructed to support the Jetson Nano, sensors, power system, and wiring located on the second layer of the chassis. Aluminium was used because it is lightweight and easy to modify during development. An extrusion-based frame using 6063-T6 aluminium was modelled and purchased, further lending itself to quick changes and modifications [14].

To match simulation-based height specifications for the cameras and LiDAR, a third layer for the chassis was developed. A steel tube was fabricated to hold a platform for the IMU, GPS cameras and LiDAR as seen in Figure 5. This extends directly above the base pole, meaning bending stress on the frame is minimized and weight is pushed down into the base as intended in a wheelchair design.

Figure 5. Implementation.

The design was kept modular so that parts could be removed or repositioned during testing. For example, the cameras have an adjustable mount which can change pitch and yaw depending on the needs of testing. This allowed quick adjustments when issues were found while wiring, placing sensors, or critical balance of the system.

VI. Results

The implementation described above underwent a series of tests to assess its viability as a flexible but effective solution.

A. Navigation Capabilities

The vehicle's ability to accurately move in response to motor commands from the navigation system on different terrains and during different times of day was tested. As the base is that of a professional motorized wheelchair, it supports many terrain types without difficulty. While some speed is gained on downhill slopes, a more advanced control system can be integrated into the software if encoders are added or odometry is more developed. Nodes can be added in the ROS 2 environment to process this change with minimal interference in the existing software solution.

The success of the completed system is summarized in Table 2. All tests described in the table were performed on light concrete with white lanes. In each, the robot started at the same position. A successful run involves no collision with any obstacles and no crossing of any lines, while also reaching the end of the course. The course was approximately two-hundred feet in length and included four obstacles, a corner, a bend, and a dotted line segment.

Table 2. Results of Robot Testing by Lighting

Lighting	**Successful Runs**
Early Morning (Low, Even Light)	7/10
Morning (Inconsistent Light)	7/10
Midday (Bright, Even Light)	9/10
Mid-Afternoon (Even Light)	10/10
Cloudy (Low, Inconsistent Light)	6/10

The reasons for failure of a run were almost exclusively lane-based. During test, no obstacle collisions occurred. The most common lane-related failures were:

1) *Lane Inversion:* If the vehicle had to turn a large degree to avoid an obstacle, it sometimes picked up the left lane as the right lane or vice versa and oriented itself outside the course. This is more common in poorer visibility scenarios with low or inconsistent light.
2) *Noisy Lanes:* During certain times of day, gravel and splotchy sections of concrete created blobs of white in the cameras that biased the centerline created from the lanes, pulling the robot outside the course. Camera exposure settings were adjusted to help mitigate this, but it was still a contributing factor.
3) *Fully Lost Lanes:* Occasionally, the robot would lose both lanes. In the event that this occurs, a failsafe activates that has the vehicle navigate towards an anchor point (as identified by LiDAR scans) and look for lanes as it travels towards the obstacle. Depending on which obstacle the robot orients to (potentially, some off of the course) this would cause a run failure.

The most successful runs occurred in consistent moderate-to-bright mid-afternoon light. Lanes were clear and overall image brightness did not shift much, meaning that the cameras were experiencing minimal noise and minimal exposure interference. Additional obstacle-only tests were conducted to ensure detection and avoidance of vertical obstacles functioned as intended. These results are summarized in Table 3. Obstacle

identification was consistent and almost always correct. Obstacles that were identified were always avoided.

Table 3. Robot Obstacle Avoidance Success

Test	Identified Obstacles	Misidentified Obstacles
1	4/4	0
2	4/4	0
3	4/4	1
4	4/4	0
5	4/4	0

The identified obstacles described in the table represent obstacles that appeared in the course and are supposed to be seen and avoided. Misidentified obstacles are artifacts or mis-scans that appeared and interfered with robot movement. Overall, based on the table data, vertical obstacle avoidance and detection was very successful. No expected obstacles were missed and all were avoided. Only one misidentified obstacle appeared, but the refresh rate of the extrapolated obstacle map removed it from view quickly and it did not interfere with long-term pathing.

B. Safety Features

During testing, ample time was given and conditions were introduced to strain the system (uphill, gravel, higher speeds, etc.) At no point did components overheat or work in unexpected manners. The E-stop system also functioned properly and stopped the robot promptly upon activation. It is worth noting that, since the braking system on the motors is not engaged upon stop, some risk still remains, even if the motors are de-energized. The risk of the vehicle sliding or rolling down steep inclines is still present and cannot be fully mitigated in the current design.

C. Modularity and Flexibility

The modular architecture proved critical to the successful development of the system. During navigation testing, it became evident that the robot experienced drift due to incomplete odometry information. To address this, a LiDAR-based odometry node was introduced to estimate motion using LiDAR data. Although this represented a significant functional addition, the ROS 2 architecture allowed it to be integrated by simply compiling a new node and subscribing to existing topics. The new node handled all required processing and published its output without requiring modifications to the rest of the system.

Similarly, when a discrepancy in the Jetson Nano's required supply voltage was identified due to a datasheet misinterpretation, the issue was resolved by adjusting the step-down converter without impacting other subsystems. In another instance, poor GPS performance caused by low mounting height and environmental interference was corrected by extending the sensor mast, enabled by the flexibility of the wheelchair-based mechanical structure.

These examples highlight how modularity in software, power distribution, and mechanical design reduced integration complexity and avoided significant redesign effort. Without this flexibility, such changes could have introduced substantial delays. By leveraging a modular approach, development time and cost are minimized, making the platform well-suited for iterative research and experimentation in autonomous systems.

VII. Limitations and Improvements

While the proposed solution proved to be quite effective for the constructed course, several improvements were identified based on the results and experiences during test to improve the design.

First and foremost, the inclusion of motor encoders as a main source of odometry information is critical in improving the design. While LiDAR-based odometry was beneficial, it was inconsistent and self-sabotaging: if LiDAR odometry drops out for a significant period of time, the robot cannot trust its navigation and must stop. Unfortunately, this makes the LiDAR map static, which stops LiDAR odometry as well. It is also not good enough for dead-reckoning if other sensors fail. With proper odometry, camera blindness is mitigated, since the odometry and past map of the centerline can be used to navigate without proper vision.

The second change that would improve the success rates of the vehicle would be replacing (or informing) the dual-camera lane system with a single stereo-vision camera in the front of the robot. Not only does this help inform odometry and obstacle identification by allowing for depth perception, but it also mitigates lane inversion issues. Having a front-facing camera also reduces the need to fine-tune the positions of the peripheral cameras, as the image is already stitched together and only a single processing node is needed for the data. This allows for more critical filtering and image processing to reduce noise and make a more accurate and longer, predictive, centerline.

Some issues in weatherproofing appeared as the design developed, as certain components (like the step-down voltage converters and IMU) required separate and inconvenient rain covers. Future designs can subscribe to the modular focus of the design and develop a fully enclosed step-down solution or a proper IMU mount.

VIII. Conclusion

The resulting autonomous robotics development platform combines modularity and flexibility in the software system, electronics distribution techniques, and chassis design to create an effective system for competitions and research relating to self-driving vehicles. Users familiar with the ROS 2 development environment can create and compile new nodes and insert them into the system with minimal interference to integrate new types of sensors or format new output modes like

driving lights or control arms. With four step-down voltage regulators attached to the system, up to four different supply voltages can be distributed across the vehicle, servicing many potential subsystems. Additionally, the supported frame can be removed from the driving base, meaning that the structure of the vehicle can be altered quickly to adapt to different use cases. With autonomous navigation continuing to be a key research topic, systems like the one developed here have the potential to improve the speed and quality of student research in the field.

ACKNOWLEDGMENT

The project's team would like to thank Dr. Luis Felipe Zapata-Rivera and Dr. Joel Schipper for their oversight and support. Their feedback and insight throughout the design and development process was instrumental in the success of the final system. We would also like to thank the Computer, Electrical, and Software Department at the Prescott, Arizona campus of Embry-Riddle Aeronautical University for providing lab spaces and materials to construct and test the design. Lastly, we would like to thank the Undergraduate Research Institute of the same university for sponsoring the project and providing funding for the purchase of components.

REFERENCES

[1] Intelligent Ground Vehicle Competition, "31st Intelligent Ground Vehicle Competition," Accessed: Apr. 2026. [Online] Available: www.igvc.org. http://www.igvc.org/index.htm

[2] NVIDIA, "Jetson Orin Nano Super Developer Kit." Accessed: Nov. 2025. [Online]. Available: https://nvdam.widen.net/s/zkfqjmtds2/jetson-orin-datasheet-nano-developer-kit-3575392-r2

[3] Open Robotics, "ROS: Home." Accessed: Nov. 2025. [Online]. Available: https://www.ros.org/

[4] Open Robotics, "Distributions — ROS 2 Documentation: Humble." Accessed: Feb. 2025. [Online]. Available: https://docs.ros.org/en/humble/Releases.html docs.ros.org

[5] A. Kumar, "Understanding Publisher and Subscriber in ROS2." *Medium*, 15 Oct. 2025. Accessed: Feb. 2025. [Online]. Available: https://medium.com/@aadhilkumar989/understanding-publisher-and-subscriber-in-ros2-d8a1ee546cd3 Medium

[6] Kai Lap Technologies Group Ltd., "KLT-USB-0362 V2 2.12 MP 0362 Sony IMX323 M12 Fixed Focus USB 2.0 Camera Module." Accessed: Feb. 2025. [Online]. Available: https://www.kailaptech.net/KLT/EN/PDF/KLT-USB-0362%20V2%202.12MP%200362%20Sony%20IMX323%20M12%20Fixed%20Focus%20USB%202.0%20Camera%20Module.pdf

[7] Slamtec, "RPLidar S2L Low Cost 360 Degree Laser Range Scanner Introduction and Datasheet Model: S2M1-R2L", Nov. 2022. Available: https://wiki.slamtec.com/download/attachments/83066883/SLAMTEC_rplidar_datasheet_S2L_v1.2_en.pdf?version=1&modificationDate=1677815477544&api=v2

[8] Globalsat, *BU-353S4 USB GPS Receiver (SiRF Star IV Chipset) — Technical Datasheet*. Accessed: Feb. 2025. [Online]. Available: https://www.nettec.no/shop/productpdfs/BU353S4.pdf

[9] Yost Labs, "3-SpaceTM Nano Evaluation Kit," *Yost Labs*, 2018. Accessed Apr. 06, 2026. Available: https://yostlabs.com/product/3-space-nano-evaluation-kit/?srsltid=AfmBOoqHvI8b8-wEgucjDM64iGTjUv9JSP_PX2AzEWhHz_-W-eF4_jCW

[10] PJRC, "Teensy® 4.1," *www.pjrc.com*. Accessed Apr. 2026. Available: https://www.pjrc.com/store/teensy41.html

[11] "PN00218-CYT14 Motor Driver 2 Channels 30Amp 7V-35V DC SmartDriveDou MDDS30," *Makermotor*, 2026. Accessed Apr. 2026. Available: https://makermotor.com/pn00218-cyt14-motor-driver-2-channels-30amp-7v-35v-dc-smartdrivedou-mdds30/?srsltid=AfmBOop-y_y3Z260glHI025P0x1nl248fjSTte3wDcfFDCkTVOp0ITJE

[12] *Arduino.cc*, 2024. Accessed Apr. 2026. Available: https://docs.arduino.cc/hardware/mkr-wan-1310/

[13] "DME of America Inc," *DME of America Inc*, 2023. Accessed Apr. 2026. Available: https://dmeofamericainc.com/products/pride-jazzy-evo-614-hd-power-chair?srsltid=AfmBOoqW5SwDfugq6KQvU3ecbowOI7CwprrS32-iQ-BxWe0E2Uw7B4lo

[14] "80/20 T-slot Aluminum Building System," 2026. Accessed Apr. 2026. Available: https://8020.net/

[15] "Gazebo Harmonic — Gazebo harmonic documentation," *Gazebosim.org*, 2023. https://gazebosim.org/docs/harmonic/install/ (accessed Apr. 06, 2026).

[16] "Nav2—Nav2 1.0.0 documentation," *docs.nav2.org*. https://docs.nav2.org/