Intelligent Ground Vehicle Competition Design Report

Clankers

University Of Detroit Mercy

Team Members:
Amna Ben Abdelkader, abenabd@udmercy.edu
Gianlorenzo Cavallo, cavallgi@udmercy.edu
Brenden Flinn, flinnbp@udmercy.edu
Erin Leahy, leahyen@udmercy.edu
Elias Madi, madier@udmercy.edu
Joseph Messina, messinjv1@udmercy.edu
Alharith Munassar, munassam@udmercy.edu
Aranza Ramirez, ramirean3@udmercy.edu
Andrew Rios, riosan@udmercy.edu
Ricardo Tapia Vargas, tapiari@udmercy.edu
Allen Zaina, zainaal@udmercy.edu

Advisors:
Dr. Vasilis Pentsos, Assistant Professor, Electrical Engineering, vpentso@udmercy.edu
Dr. Michael Santora, Associate Professor Electrical Engineering, santormj1@udmercy.edu

Challenge: AutoNav

April 30, 2026

Contents

1 System and Subsystem Requirements

The system architecture shown in Figure 1 is organized as a modular ROS 2 framework, where functionality is divided into key subsystems as an intentional engineering choice to improve scalability, reliability, and ease of integration. The perception layer processes raw sensor inputs (LiDAR, camera, IMU, GPS) into structured environmental data, which feeds into the localization subsystem. Localization uses sensor fusion (via an Extended Kalman Filter) to estimate the vehicle's state, providing accurate position and orientation information. The navigation subsystem builds on this by generating a costmap and computing feasible paths based on environmental constraints and goal inputs.

At a higher level, the control subsystem translates navigation outputs into actuator commands, interfacing with a Texas Instruments TM4C123GH6PM microcontroller (Tiva C) to generate motor PWM signals. Overarching these layers is the system integration framework, which coordinates communication between nodes using standardized ROS message types and ensures safe operation through mechanisms such as the emergency stop. This structured division of perception, localization, navigation, control, and integration reflects deliberate engineering decisions to decouple complex functionalities while maintaining a cohesive and responsive autonomous system, as illustrated in Figure 1.

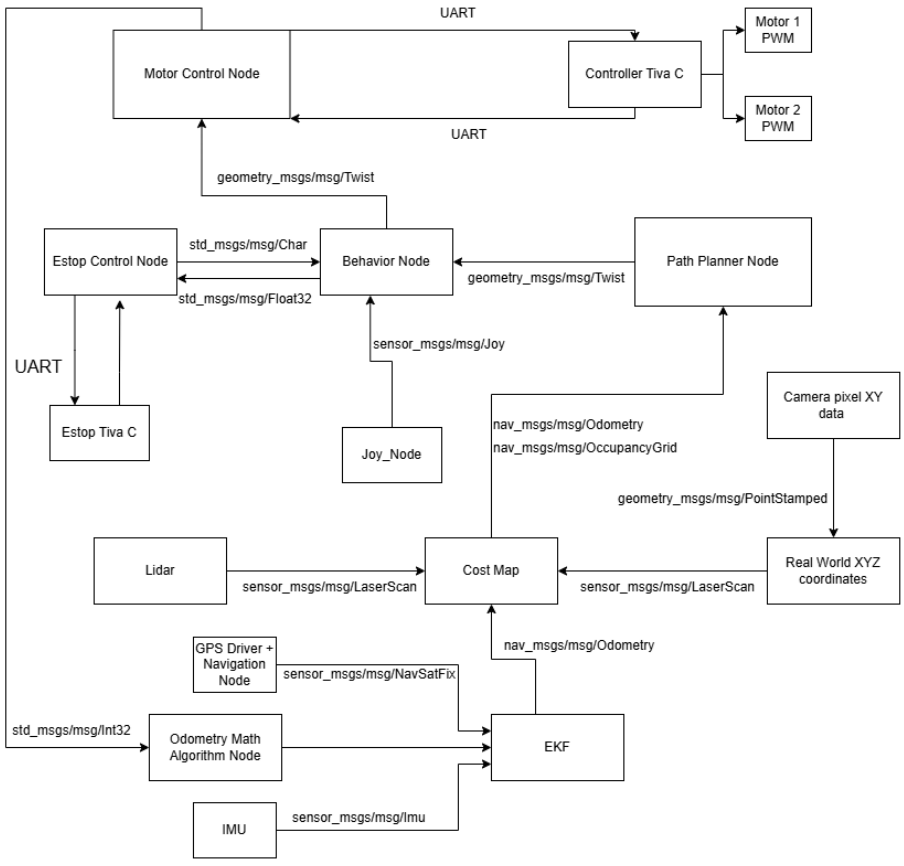


Figure 1: Pipeline

2 Mechanical Design

The vehicle mechanical design consists of a body frame, chassis, and a two-motor differential drive assembly. The platform measures 0.62 m in length and 0.62 m in width, with an overall height of 1.07 m including wheels and 0.97 m at the frame. These dimensions satisfy IGVC size constraints while providing sufficient deck space for onboard electronics and sensor payloads.

The chassis is constructed from aluminum, selected for its favorable strength-to-weight ratio, corrosion resistance, and ease of machining. An aluminum frame forms the main skeleton of the vehicle, and a flat metal base plate supports the motors and drivetrain. The upper frame mounts to the base chassis and supports electrical enclosures, LiDAR wiring, and the emergency stop system. This modular structure allows for straightforward integration and maintenance of mechanical and electrical subsystems.

The drivetrain configuration is illustrated in Figure 2, showing the direct coupling between the motor, gearbox, and wheel.

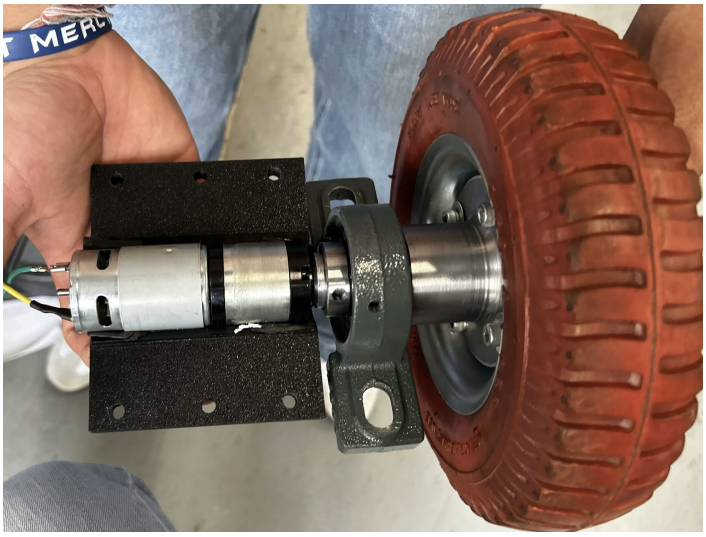


Figure 2: Drivetrain

Each drive wheel has a diameter of 8 inches, selected based on torque and speed requirements derived from motor specifications and verified by the calculations in Section 2.1. A 51:1 integrated gearbox provides sufficient torque output for propulsion under expected operating conditions. The gearbox output shaft is directly coupled to the wheel through a custom-designed mount, which is secured to the chassis using a SolidWorks-designed mounting block, ensuring a rigid and stable connection. The two drive wheels are positioned near the vehicle's center of mass to improve stability and traction.

To handle axial and radial loads without damaging drivetrain components, the wheel assembly incorporates a pillow hub. Early designs lacked this support, causing motor shaft damage from excessive axial stress during testing. The pillow hub reduces stress on the motor shaft and minimizes wheel oscillation, improving both reliability and encoder measurement accuracy. The wheel mount also includes a fender that supports the encoder on the outer wheel face.

The vehicle uses a three-point caster configuration to maintain stability. Two 5-inch casters are placed at the rear of the platform, where the majority of system mass, including the motors, batteries, and drivetrain, is concentrated, providing load distribution on the heavier end. A 4-inch caster is positioned at the front to prevent forward tipping. This three-point arrangement ensures

consistent ground contact and reduces wobbling during operation, including when ascending the 15-degree ramp specified by IGVC.

The vehicle's center of mass is located approximately 11 inches from the front reference point. It is intentionally shifted slightly forward of center to improve maneuverability and control responsiveness while maintaining overall system stability.

The housing uses a modular architecture that allows motor and sensor removal and replacement. Custom mounts house sensitive electronic components on ABS plastic plates within sealed enclosures, providing environmental protection, organized cable routing, and convenient access for maintenance. Externally mounted sensors use a combination of 3D-printed PLA and aluminum brackets selected based on each sensor's IP rating, maintaining a lightweight and mechanically stable structure. Waterproof electrical enclosures with sealed cable inserts and outdoor-rated connectors protect onboard electronics from moisture. Critical components are positioned higher on the vehicle frame to reduce exposure to water and debris, while custom 3D-printed covers provide additional protection. Together, these design choices produce a modular, weather-resistant system suitable for reliable outdoor operation.

To supply power to the differential-drive motors, two DC-DC buck converters step the 36 V battery voltage down to 24 V. Each converter drives a Cytron MD20A motor driver, which controls the gear motor for each drive wheel.A Tiva C mounted on a custom PCB regulates angular velocity using the onboard Quadrature Encoder Interface (QEI) peripheral [1]. Two PI controllers running at 1 kHz adjust the PWM duty cycle based on measured angular velocity and the target velocity commanded by the APF path planner [2].

The drivetrain was sized to satisfy two performance requirements: achieving a target operating speed within the IGVC maximum of 2.235 m/s (5 mph), and generating sufficient torque to propel the fully loaded vehicle up the 15° ramp specified by IGVC. The vehicle has an unloaded mass of 18.14 kg (40 lb) and carries a sensor and electronics payload of at least 9.07 kg (20 lb), giving a total operating mass of 27.22 kg (60 lb). The selected gearmotor operates at 24 V DC with an integrated 51:1 reduction stage, producing a no-load output speed of 230 RPM, a rated output torque of 35 kg·cm (3.43 N·m), and a stall torque of 50 kg·cm (4.90 N·m). The governing design case is ramp climbing at full payload, which imposes a larger torque demand than flat-ground rolling resistance.

Figure 3 shows the free-body diagram of the vehicle climbing the 15° ramp. The normal force N acts perpendicular to the ramp surface, the vehicle weight mg acts vertically downward, and the drive force F_{drive} acts along the ramp in the uphill direction.

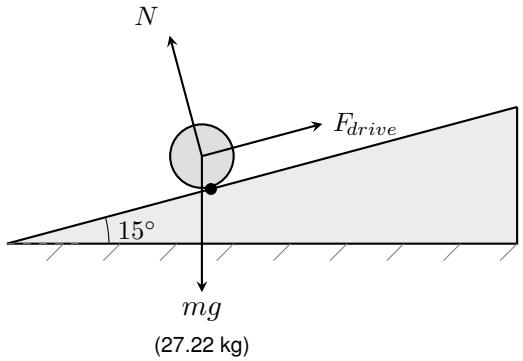


Figure 3: Free-body diagram of the fully loaded vehicle climbing the 15° IGVC ramp. N acts perpendicular to the ramp surface; mg acts vertically downward at the full operating mass of 27.22 kg (60 lb); F_{drive} acts along the ramp in the uphill direction.

2.1 Drivetrain Sizing

The torque required at each drive wheel is:

$$\tau_{ramp} = \frac{1}{2} \cdot m \cdot g \cdot \sin(15) \cdot r_w \tag{1}$$

where $m = 27.22$ kg (60 lb) is the total operating mass including payload, $g = 9.81$ m/s^2, and $r_w = 0.1016$ m is the wheel radius (8-inch diameter). The factor of $\frac{1}{2}$ distributes the load equally across the two drive wheels. Substituting:

$$\tau_{ramp} = \frac{1}{2} \times 27.22 \times 9.81 \times 0.2588 \times 0.1016 \approx 3.51 \text{ N·m per wheel} \tag{2}$$

The selected gearmotor provides a rated continuous output torque of 3.43 N·m and a stall torque of 4.90 N·m at 24 V. Comparing against the per-wheel ramp requirement:

$$\frac{\tau_{stall}}{\tau_{ramp}} = \frac{4.90}{3.51} \approx 1.40 \tag{3}$$

$$\frac{\tau_{rated}}{\tau_{ramp}} = \frac{3.43}{3.51} \approx 0.98 \tag{4}$$

The rated continuous torque of 3.43 N·m falls marginally below the calculated per-wheel ramp requirement of 3.51 N·m by approximately 2.3%. However, ramp traversal is a brief transient event rather than a sustained continuous load, and motors tolerate short-duration peak currents beyond their rated continuous capacity without thermal damage. The stall torque of 4.90 N·m exceeds the ramp requirement by a factor of 1.40, confirming the motor generates sufficient torque to complete ramp traversal. During normal flat-ground operation, the motor runs well within its rated continuous envelope.

At rated load, the gearmotor output shaft turns at 190 RPM. The corresponding vehicle speed is:

$$v_{rated} = \omega_{rated} \cdot r_w = \frac{190 \times 2\pi}{60} \times 0.1016 \approx 2.02 \text{ m/s } (4.52 \text{ mph}) \tag{5}$$

The theoretical no-load speed of 230 RPM corresponds to 2.45 m/s (5.48 mph), which exceeds the IGVC limit of 2.235 m/s (5 mph). In practice, the PI velocity controllers running at 1 kHz on the microcontroller regulate wheel speed to a commanded target, preventing the vehicle from reaching no-load speed during operation. The rated operating speed of 2.02 m/s satisfies the competition constraint with a 9.6% margin.

The 8-inch wheel diameter balances three constraints. A larger diameter increases vehicle speed at a given output RPM per Equation 5 but raises the per-wheel torque requirement in Equation 2 due to the larger moment arm. A smaller diameter reduces torque demand but lowers achievable speed and reduces ground clearance over course terrain. At 8 inches, the gearmotor stall torque exceeds the ramp requirement by a factor of 1.40 while the vehicle achieves a rated operating speed of 2.02 m/s, satisfying both constraints.

The selected 24 V gearmotor with 51:1 reduction satisfies the drivetrain requirements for the fully loaded vehicle across both performance constraints. At the total operating mass of 27.22 kg (60 lb), the stall torque of 4.90 N·m exceeds the per-wheel ramp climbing requirement of 3.51 N·m by a factor of 1.40, providing adequate margin for transient loading during ramp traversal. The rated operating speed of 2.02 m/s (4.52 mph) complies with the IGVC 5 mph speed limit, and active PI velocity control prevents the motor from reaching its unloaded speed of 2.45 m/s under

any commanded operating condition. The motor's rated continuous torque is sufficient for all flat-ground operation, and the available stall torque margin ensures ramp traversal can be completed without stalling even under full payload. Together, these characteristics confirm the gearmotor selection is appropriate for the vehicle's operating envelope and competition requirements.

3 Safety

In transport, the aluminum frame protects components from impact. Fasteners secure major components and prevent movement during transport. When parked, the vehicle maintains stability through a wide base and balanced mass distribution. The emergency stop button shuts down the vehicle during a safety event. During battery charge, protective housing covers the battery and prevents contact with exposed wiring or short circuits.

The robot employs mechanical and wireless emergency stop systems to provide shutdown during emergency conditions. The mechanical E-Stop mounts on the vehicle, while the wireless E-Stop uses an onboard transmitter and remote module. Both systems interrupt power to the drive system when activated. A controlled-stop mode slows the robot before power cutoff and reduces inertial effects during shutdown.

The microcontroller monitors the emergency stop states in real time through the emergency stop circuit. An optical isolator separates high-power safety circuitry from low-voltage microcontroller inputs. The microcontroller controls a relay and MOSFET switching circuit that selects immediate power cutoff or controlled stop through an onboard selector switch.

The wireless emergency stop triggers at 600 ft and exceeds the required 100 ft range. The emergency stop circuit actuates a relay that disconnects motor power when either the wireless or mechanical stop signal activates. The E-stop circuit also responds to a software selector switch that initiates a controlled stop. Megatron draws power from a 36 V battery pack consisting of three 12 V batteries connected in series. Individual batteries and the full pack charge through 110 V three-prong grounded chargers without a voltage converter.

4 Electrical & Electronic Design

The electrical and electronic design of the autonomous ground vehicle enables reliable sensing, computation, communication, and actuation required for fully autonomous navigation. The system is designed to support real-time perception, precise motor control, and robust safety mechanisms while operating within strict power and performance constraints.

To achieve this, the architecture distributes functionality across dedicated subsystems. High-level autonomy and path planning are handled by an onboard computing platform, while real-time perception is performed using GPU-accelerated processing. Low-level control and safety-critical operations are isolated on microcontrollers to ensure deterministic response and fault tolerance.

The vehicle integrates multiple sensing modalities, including depth imaging, LiDAR, and inertial and satellite-based localization, to provide comprehensive environmental awareness. Actuation is achieved through DC motors with closed-loop feedback, enabling precise motion control. A regulated power distribution system supports all subsystems, and onboard networking enables communication between processing units.

The system architecture separates hard real-time and soft real-time tasks within a distributed computing framework to reduce resource contention and improve system performance. Latency-aware designs demonstrate 94% deadline compliance versus 67% for combined approaches, a 40% improvement that supports the decision to separate tasks [3].

Hard real-time functions, such as motor control and emergency stop logic, execute on dedicated microcontrollers to guarantee deterministic timing and immediate response to safety-critical events. In contrast, the NVIDIA Jetson Orin Nano and Intel NUC9 manage perception, mapping, and high-level planning. This distributed approach upholds strict control timing while enabling computationally intensive algorithms, as validated by the performance gains in layered real-time AI systems.

Table 1. Electrical and Electronic Components Supporting Vehicle System Requirements

System Requirement	Electrical / Electronic Components	Driven Electrical Aspects
High-level processing and autonomy algorithms	Intel NUC9	CPU throughput, system memory bandwidth, communication interfaces
Real-time perception processing	NVIDIA Jetson Orin Nano	GPU processing capability, sensor data bandwidth, processing latency
Lane detection and obstacle avoidance	Intel RealSense D455 depth camera, Ouster OS1 LiDAR	Sensor resolution, frame rate, data interface bandwidth
Localization and navigation	GNSS multi-band helical antenna, IMU	Position accuracy, update rate, signal processing
Motor control and vehicle actuation	TM4C123GH6PM microcontrollers, Cytron MD20A motor drivers, RS550 brushed DC motors with 50:1 planetary gearboxes	PWM control signals, motor current capacity, voltage regulation
Motion feedback	Quadrature optical encoders	Pulse frequency, timing accuracy
Power distribution and system operation	12.8V 30Ah $LiFePO_4$ batteries, DC–DC buck converters	Voltage levels, current delivery, power efficiency
System communication	TP-Link AXE5400 Archer AXE75 router	Wireless bandwidth, latency
Safety and emergency stop	Safety microcontroller, E-stop circuitry	Low latency response, fail-safe electrical logic

Component selection balances cost, performance, and system requirements for autonomous vehicle operation. The NVIDIA Jetson Orin Nano delivers 67 TOPS GPU acceleration for real-time perception at enabling 30-90 FPS obstacle detection, critical for dynamic environments.

The Intel NUC9 provides 6-core CPU processing (2.4 GHz base) with 64GB DDR4 support for high-level planning and system coordination in a compact 238mm x 216mm x96mm form factor, ideal for vehicle integration.

TM4C123GH6PM microcontrollers (80 MHz ARM Cortex-M4) ensure deterministic motor control and safety functions, while Cytron MD20A drivers provide up to 20 A bidirectional current for RS550 gearmotors. The drivetrain delivers approximately 3.43 N·m of continuous torque and up to 4.90 N·m peak torque per wheel. As shown in (3) and (4), the rated torque is slightly below the ramp requirement, while the available peak torque provides a sufficient margin for transient ramp traversal.

The Intel RealSense D455 provides short-range depth sensing with up to 1280×720 resolution at 90 Hz and an effective operating range of approximately 0.4–6 m, enabling reliable perception

for lane detection and near-field obstacle avoidance. This is complemented by the Ouster OS1 LiDAR, which delivers 45° environmental mapping with a range of up to 200 m at 20 Hz, supporting long-range obstacle detection and global situational awareness.

For motion feedback, 600 PPR quadrature encoders provide approximately 0.6° angular resolution, enabling precise closed-loop control of wheel position and velocity. The system is powered by 30 Ah $LiFePO_4$ batteries, supporting over 2 hours of operation at an average load of 150 W.

Megatron draws power from a 36 V $LiFePO_4$ battery pack consisting of three 12 V, 30 Ah batteries connected in series, providing 384 Wh of energy capacity. The system operates for up to 3 hours under nominal loads, with quick battery swaps enabling extended missions. Individual batteries and the full pack charge directly from standard 110 V AC three-prong grounded chargers.

The 36 V battery powers motor controls and mechanical systems. The 36V is bridged to 2 Variable DC-DC converters, allowing for mechanical governing of the DC motors by limiting supply voltage, one 24V DC-DC converter powering the Ouster LiDAR, and one 12V DC-DC converter, powering all other features such as the 2 onboard computers, saftey systems, and status LEDs. This segmented architecture isolates subsystem loads, improves reliability, and simplifies maintenance

5 Perception

The perception pipeline detects cones, barrels, and other obstacles in the vehicle's local environment. LiDAR data passes through filters and clustering algorithms to convert raw point clouds into obstacle information for the planner. A camera-based pipeline complements this system with YOLOv11 object detection on red-green-blue (RGB) images [4]. Depth data from the Intel RealSense D455 camera provide 3D position estimates relative to the vehicle. Together, these outputs give the planner a spatial representation of obstacles and free space for safe navigation through the AutoNav course.

LiDAR point cloud processing begins with the collection of a 3D point cloud around the vehicle. The algorithm takes in a ROS 2 PointCloud2 stream from the Ouster LiDAR and transforms the cloud into the working frame, where obstacle detection is performed. A crop box filter, removes points outside a predefined region of interest to restrict processed to that of which is relevant. Since the lane width is approximately 20 feet wide, the active crop box covers about 6 meters in front of the vehicle, 1 meter behind, and roughly 3.5 meters to both the left and right sides, allowing the system to monitor the driving corridor while excluding unnecessary distant data. The bounds are defined by minimum and maximum x, y, and z coordinates in the working frame. To reduce point cloud size and improve clustering efficiency, voxel grid downsampling is applied after cropping, consistent with prior work reporting a 91.89% simplification rate[5]. Ground segmentation is then applied to remove floor points so the driving surface is not treated as an obstacle. The remaining obstacle points are converted into a synthetic ROS 2 LaserScan message instead of being processed into bounding boxes. For each scan angle, the algorithm records the nearest occupied point within the configured scan range, creating a 2D obstacle representation around the robot. This synthetic LaserScan topic, published as /fake_scan, allows the LiDAR data to be used directly by navigation and costmap packages that expect LaserScan input.

The filtered cloud is then processed using Euclidean cluster extraction, which groups neighboring points into individual obstacle clusters based on distance tolerance. For each cluster, a 3D axis-aligned bounding box is generated by finding the minimum and maximum x, y, and z coordinates of the points in the cluster, creating a box that fully encloses the detected object. These 3D boxes are then projected into 2D outlines and published as a standard ROS 2 LaserScan message

whose range values are generated from detected obstacle edges rather than measured directly by a planar laser scanner. This allows compatibility with navigation and costmap packages that expect LaserScan input.The main parameters affecting performance are voxel leaf size, crop box bounds, ground segmentation threshold, scan angle range, and scan range limits. These values were tuned through repeated testing in RViz and live runs by observing scan density, obstacle visibility, false positives, missed detections, and processing stability. Validation was performed by placing obstacles at known measured positions and comparing those distances to the reported LaserScan ranges. Repeated trials were also used to verify consistency, meaning the same obstacle produced similar scan distances across multiple runs with only minor variation.

LiDAR supports obstacle detection and spatial awareness but provides limited lane information due to weak geometric contrast between painted lines and ground surfaces. Optimal lane detection performance is achieved when the camera is mounted below approximately 0.8 m. At greater heights, perspective mismatch with the pretrained YOLOPv2 model, which is optimized for wider roadway scenarios, degrades accuracy on the narrower lanes typical of the AutoNav course. To compensate for the limited lane tracking abilities of the LiDAR the camera pipeline detects lane markings, grass boundaries, and other visual features that define the drivable region.

The camera-based perception system processes synchronized RGB and depth data from the Intel RealSense sensor to extract structured environmental features. The YOLOPv2 model introduced by Han et al, was slightly modified to use the lane-line segmentation head to produce binary masks representing lane boundaries in the image space [6]. Rather than using the full mask, a set of representative points is sampled along each detected lane (e.g., five points per boundary) to reduce noise and computational load while preserving geometric structure.

Each sampled pixel is paired with depth data and projected into 3D space using camera intrinsics, producing the coordinates (x, y, z) relative to the camera frame. These points are then mapped onto the ground plane by constraining the vertical component, enabling a consistent 2D representation of the lane boundaries in the vehicle's local frame. This approach allows the system to estimate lane geometry and width while maintaining stability in the presence of segmentation noise.

YOLOv11 is applied to RGB images to detect and classify objects. Instead of full bounding boxes, key points such as the bottom-center and bottom edges of each detection are selected to better approximate ground contact locations. These points are then converted to 3D coordinates using depth data to provide accurate spatial positioning of obstacles relative to the vehicle.

Metric	**Value**	**Notes**
Lane Detection Rate	16 Hz	Stable publishing rate
Object Detection Rate	30 Hz	Average over runtime
Max Detection Range	13 m	Sunny conditions
Reliable Width Range	$\leq$ 10 m	Beyond this, width less stable
Distance Error	0.5%	Within 10 m
Width Error	3% (max 6.7%)	Worst-case deviation
Lane Deviation	6.25%	Across test runs

Table 2: Camera-Based Perception Performance Metrics

Object Class	mAP@50	mAP@50-95
Barrel	0.934	0.638
Orange Cone	0.845	0.597
Pothole	0.759	0.492

Table 3: YOLOv11 Object Detection Performance

The LiDAR and camera pipelines combine to transfer data into a LaserScan representation for costmap integration. Figure 4a shows the lane detection output, while Fig. 4b shows object detections from the YOLOv11 perception pipeline.

(a) Lane detection output

(b) Object detection output

Figure 4: Perception outputs used for navigation

Empirical evaluation of the camera-based system confirmed stable and repeatable object detection across diverse lighting conditions, including indoor, outdoor, shaded, and direct sunlight environments. The fine-tuned YOLOv11 model reliably identified cones and barrels within a defined operating range. At distances greater than approximately 1 m, the camera provides high-confidence detections; below this threshold, LiDAR serves as the primary detection modality. This handoff applies mainly as an initial condition: if a course object is closer than 1 m at the start of a run, LiDAR covers detection until the object moves into the camera's reliable range, after which both modalities track it with high confidence. The camera system detects objects up to 13 m in sunny conditions, with reliable width estimation up to 10 m. Within that range, distance estimation deviation is approximately 0.5% and width estimation deviation is approximately 3%, with a maximum observed deviation of 6.7%. The LiDAR-based detection module exhibits a maximum positional deviation of approximately 5% in distance and 6.78% in object width, confirming geometric consistency in near-field conditions.

In cases where adjacent objects such as closely spaced cones are grouped within a single bounding box, the system still correctly represents occupied space. This reduces instance-level separation but remains acceptable for navigation, as the vehicle is still prevented from entering the detected region. Occasional false positives were observed but do not significantly degrade performance due to the navigation strategy's conservative obstacle avoidance behavior.

The evaluation of the perception subsystem compares target performance with measured performance. This comparison assesses perception capability and navigation support.

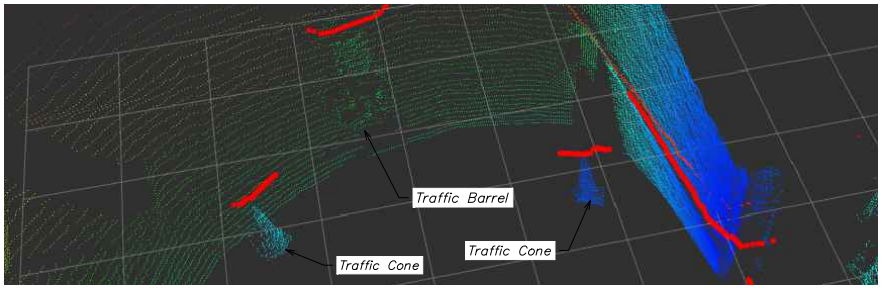


Figure 5: LiDAR LaserScan Output

The LiDAR validation tests demonstrated consistent and repeatable distance measurements from 2 m to 6 m under controlled indoor conditions. Measured distances remained close to the ground truth values verified by tape measure, with accuracy ranging from 94.75% to 97.97%. Standard deviation remained low across all trials, indicating stable sensor performance with minimal variation between measurements. Summarized in Table 4

Distance (m)	Mean Range (m)	Std Dev (m)	Error (m)	Percent Error (%)
2	1.895	0.066	-0.105	5.25
3	2.869	0.051	-0.131	4.37
4	3.813	0.118	-0.187	4.68
5	4.898	0.074	-0.102	2.03
6	5.857	0.080	-0.143	2.38

Table 4: LiDAR distance accuracy measurements

These results confirm that the LiDAR provides reliable ranging performance within this distance range in a controlled environment.

The results indicate that the perception subsystem achieves real-time operation and produces planner-compatible outputs. Obstacle detection and classification performance are generally reliable. The camera model produces high-confidence detections, and the LiDAR system consistently detects obstacles within the defined region. However, variability in clustering stability and minor inconsistencies in obstacle boundaries suggest that further parameter tuning and filtering are required. Additionally, challenges with frame alignment during development highlight the importance of robust transform management.

6 Driving Logic

The system defines GPS-based waypoints in global latitude and longitude coordinates and transforms them into the planner's local Cartesian frame. GPS coordinates are first converted into a local Cartesian reference using `navsat_transform`, allowing the planner to operate in meters rather than WGS84 coordinates. Dual GPS is used to estimate heading by computing the relative east/north offset between the front and rear antennas, which provides a yaw estimate for the vehicle. An Extended Kalman Filter (EKF) fuses GPS measurements with wheel encoder and IMU data to produce a stable pose estimate [7]. A second EKF combines wheel odometry and IMU data to form a smooth local estimate in the odom frame, which is then corrected using GPS-based position in the map frame to produce a globally aligned pose.

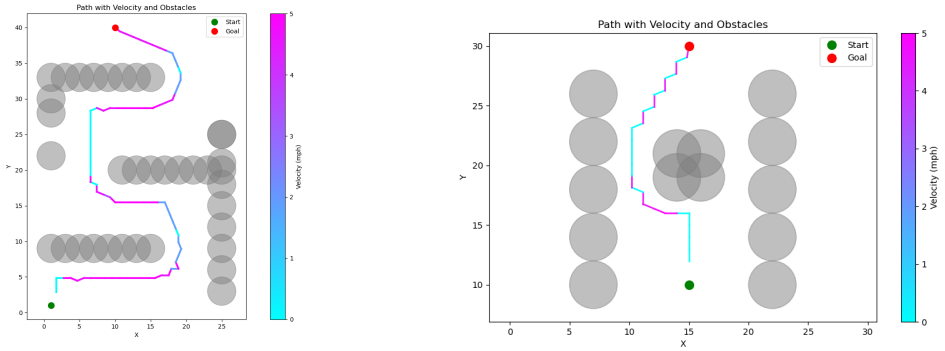
The global and local costmaps use obstacle and inflation layers to represent nearby obstacles and maintain safe clearance. LiDAR and camera obstacle detections are converted into LaserScan inputs, which the Nav2 obstacle layer uses to mark occupied cells and clear outdated free space through raytracing. Obstacle marking occurs when detections fall within 10.0 m of the robot, while clearing is applied out to 12.0 m. The inflation layer assigns higher costs to cells near obstacles, with costs decreasing exponentially as distance increases. As a result, the costmap provides a continuously updated representation of the environment that supports obstacle avoidance and path planning.

SLAM Toolbox provides the map-to-odom transform, allowing the costmap to remain aligned with the global frame. As the robot traverses the course, the local costmap updates as new obstacles enter the sensor field of view. The APF planner uses this map representation to compute paths around obstacles and along course boundaries.

The vehicle navigation system is implemented as a ROS2 C++ node operating at 10 Hz, taking an occupancy grid from perception and GPS-derived waypoints in a local Cartesian frame to generate velocity commands. At each planning cycle, the planner constructs a local potential field from two components. Occupied grid cells are converted into obstacle structures that generate repulsive potentials. The active waypoint generates an attractive potential that pulls the vehicle toward the goal. Gradient descent over this combined field produces a candidate path, and the planner selects a look-ahead point as the steering target.

Two complementary strategies address the local minima problem inherent to potential field planners. A circular sampling strategy [8] evaluates candidate motion directions around the vehicle during each planning cycle, reducing the likelihood of entering trap regions by exploring alternative descent directions. This operates as a preventative measure during normal planning. When the planner detects that progress toward the goal has diminished, a simulated annealing–based perturbation method [9] temporarily accepts suboptimal motion directions to escape the trap region before resuming gradient-based descent.

Figure 6a shows the APF planner escaping a trap region using simulated annealing perturbation. Figure 6b shows the planner navigating a center island obstacle.



(a) APF planner escaping a switchback trap region. (b) APF planner navigating a center island obstacle.

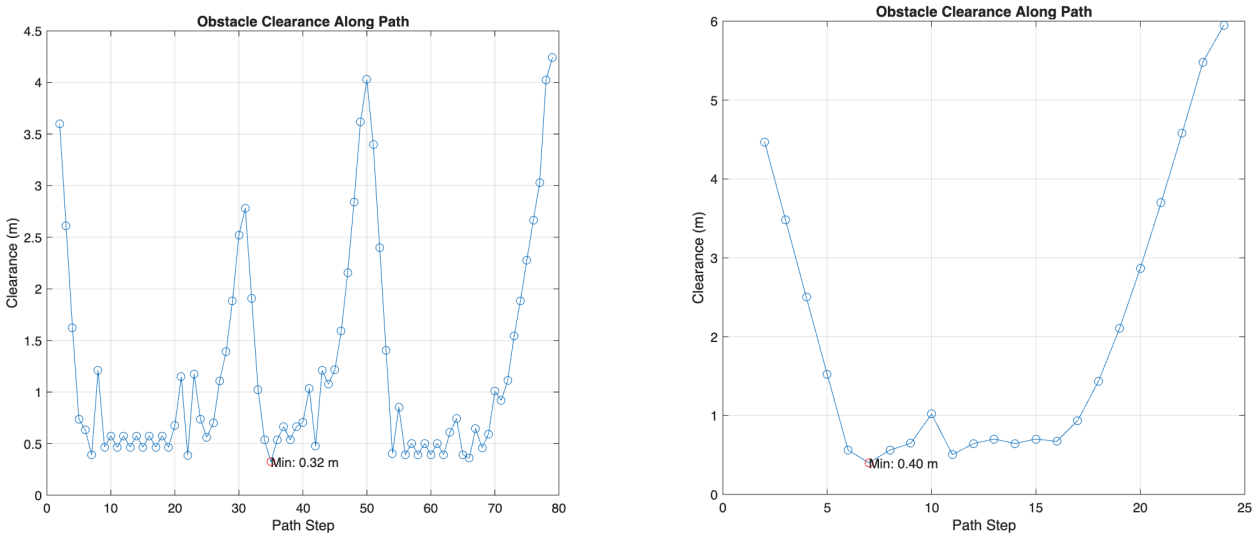
Figure 6: Path planning outputs on various tracks

The path planner was tested in ROS2 by commanding the vehicle to navigate to several predefined goal positions that represented GPS waypoints. The system recorded the final vehicle pose and computed the Euclidean distance from each target to evaluate navigation accuracy. Results are summarized in Table 5.

Target Position (m)	Final Position (m)	Distance Error (m)	Result
(2.50, 8.00)	(2.54, 7.79)	0.22	Reached
(5.50, 2.50)	(5.57, 2.76)	0.27	Reached
(9.00, 8.00)	(8.85, 7.86)	0.20	Reached
(8.00, 2.00)	(7.95, 2.29)	0.30	Reached

Table 5: ROS 2 Path Planner Navigation Accuracy

The planner satisfies the minimum obstacle avoidance distance of 0.3 m. Figure 7 illustrates obstacle clearance as a function planning steps. For the switchback and center island trials, the planner exceeded the minimum clearance by 0.02 m and 0.10 m, respectively.



(a) Switchback clearance vs time

(b) Center island clearance vs time

Figure 7: Obstacle avoidance performance and planned path

Velocity commands published by the planner generate vehicle motion through geometry_msgs/Twist messages sent to the cmd_vel topic. These commands specify desired linear and angular velocities, which the motor controller converts into wheel speeds that drive the robot through the course. Odometry data published on the odom topic monitors vehicle position by integrating wheel encoder measurements to estimate pose and velocity over time. A square trajectory validation test recorded 1390 odometry messages over an expected travel distance of 12.0 m. The measured path length reached 11.51 m, corresponding to a 4.08% error. Final position drift of 0.228 m (1.90%) indicates reliable motion monitoring with minimal accumulated error.

The planner uses an internal waypoint tolerance of 0.5 m, which is stricter than the IGVC competition threshold of 2.0 m. Testing produced position errors between 0.20 m and 0.30 m, with the maximum error representing 15% of the competition tolerance. These results confirm the system exceeds the competition requirement while remaining well within the planner's internal threshold.

The target planning rate of 10 Hz was met. The mean cycle execution time of 0.63 ms, maximum observed 4.68 ms over 445 cycles. The 10 Hz timer period allocates 100 ms per cycle; the maximum observed execution time consumed 4.7% of that budget, confirming the real-time re-

planning requirement is met with a 95% margin. The C++ implementation was chosen specifically to satisfy this requirement, and the timing results validate that decision.

The results of the driving logic validation tests are summarized in Table 6.

Metric	Target	Measured (avg)
Planning rate	10 Hz	1.6 kHz
Waypoint error	0.5 m	0.25 m
Navigation accuracy	1.0 m	0.25 m
Odometry distance	12.0 m	11.51 m
Trajectory error	5%	4.08%

Table 6: Driving logic validation results

7 Key Performance Indicators

The robot's performance is evaluated base on key performance indicators (KPIs) such as navigation accuracy, obstacle avoidance, lane following, waypoint progression, speed, safety, and system reliability. Quantitative metrics measure each KPI to confirm autonomous operation on the IGVC course.

To determine navigation accuracy, estimated vehicle position was subtracted from the euclidean distance between estimated vehicle position and known waypoint locations along the a course. An EKF and APF planner support position estimation and navigation control. The error between target waypoint and actual GPS reading was 0.24 m, still with in the maximum allowable error 0.5 m. GPS noise and sensor uncertainty account for the margin of error.

To gather obstacle avoidance performance metrics the percentage of detected obstacles relative to known obstacle positions are compared. Target detection accuracy equals 95%. Lane following performance uses lateral deviation from lane center. Target lateral deviation equals 0.3 m to maintain vehicle position within lane boundaries.

Completion time and vehicle speed use onboard logs and timed test runs. Target operating speed equals 2–4 mph to balance course completion and vehicle stability. The 2–4 mph speed target was selected to balance course completion time with vehicle stability and obstacle reaction time.

Safety and system reliability are evaluated using response and endurance metrics. Emergency stop response time remains below 0.5 s to allow rapid vehicle shutdown during unsafe conditions. Battery performance is evaluated using runtime and voltage stability metrics, with a target runtime of 2 hours. Battery voltage must remain within the specified operating range to ensure reliable power delivery to all onboard systems.

8 Analysis of Complete Vehicle

Project evaluation revealed several system design limitations. Multiple communication nodes increased system complexity and introduced additional failure points. Object detection nodes generated high data throughput that exceeded available RAM capacity. High bandwidth demand from perception modules overloaded system memory and reduced performance.

These results highlight the need for balance between modular architecture and computational efficiency. Excess node fragmentation increases latency and resource overhead. Future designs require optimization within hardware limits.

Preliminary hardware tests revealed an electrical failure in the original planetary gearmotors when the encoder A-channel waveform lost signal during validation. Inspection revealed the encoder came in contact with the metal motor case, which produced an electrical short. An external encoder mounted on the output shaft, isolated from the motor housing eliminated the short.

Mechanical testing also revealed motor damage from excessive axial load. The original motor mount lacked sufficient structural support to withstand the wheel forces. A revised mount design added stronger structural support and external load-bearing components such as a pillow block hub. The improved design transferred axial and radial loads away from the motor shaft to prevent repeat failures, improve encoder reliability, and drivetrain durability.

GitHub served as the version control platform for software development. Separate repositories organized programs developed by different subsystem groups. Code comments tracked known bugs, and GitHub branches separated experimental features from validated software versions.

In simulation, RViz used LiDAR sensor data to populate an occupancy grid used for costmap generation. A TurtleBot3 platform simulated vehicle motion during testing. This setup enabled visualization of how the navigation stack interpreted obstacles and free space within the simulated environment, improving debugging and system troubleshooting. The LiDAR data populated both local and global costmaps representing the vehicle's operating environment. Simulink simulated the angular velocity PI controller firmware for each motor and supported controller gain tuning. To simulate the path planner, ROS2 Turtlesim accepts linear and angular velocity commands from the path planner to simulate robot motion along the planned path. These commands publish as geometry_msgs/Twist messages to the cmd_vel topic. During physical testing, the robot publishes odometry information to the odom topic while the planner continues to send velocity commands through cmd_vel.

Major hardware components were tested individually before integration into the MEGATRON vehicle. These tests verified sensing accuracy, localization performance, and drivetrain functionality. Table 8 summarizes the component-level validation procedures and results.

Component	**Test Description**	**Result**
Encoders	Linear and square trajectory tests using a localization-on-wheels cart with MATLAB analysis	Consistent distance tracking and motion estimation
IMU	Static drift test followed by multi-axis motion validation	Stable readings with minimal drift across all axes
GPS (RTK)	Indoor and outdoor localization tests using rover and base station configuration	Indoor accuracy: 1.5–2 m; outdoor accuracy: 0.24 m
Camera	Perception subsystem validation for lane and obstacle detection	Results discussed in the perception subsystem section
Motors and drivetrain	System integration test of motors, gearboxes, and encoder coupling	Proper actuation and encoder feedback observed

Table 8: Component-Level Hardware Testing

9 Cyber Security Analysis

Autonomous vehicles rely on wireless and network communication to perform core functions. These systems introduce cybersecurity risks that affect robot operation and public safety.

One vulnerability involves wireless emergency stop communication. Jamming or spoofing of the wireless signal can block the stop command or trigger a false shutdown event. A mechanical E-stop provides a backup shutdown method during wireless disruption and ensures that operators retain physical control of the vehicle.

A second vulnerability involves denial-of-service (DoS) attacks on the vehicle network. Excessive network traffic overloads communication channels and interrupts message passing between onboard systems. This disruption delays perception updates or prevents the planner from receiving critical data required for navigation. Network traffic rate limits and firewall rules protect system communication and reduce the risk of malicious traffic overwhelming the system.

A third vulnerability involves unauthorized access to onboard control systems. Weak passwords, open SSH access, or unsecured remote desktop connections allow attackers to modify commands or access system files. Strong passwords, restricted SSH access, and protected directories with controlled permissions limit system access and prevent unauthorized modification of control software.

References

[1] T. Instruments, "Tiva tm4c123gh6pm microcontroller data sheet," *Texas Instruments Incorporated: Dallas, TX, USA*, 2014.

[2] N. S. Nise, *Control systems engineering*. John Wiley & Sons, 2019.

[3] "Latency-aware design of real-time artificial intelligence systems," *IJSRED*, vol. 9, no. 1, p. 22, 2023. [Online]. Available: https://ijsred.com/volume9/issue1/IJSRED-V9I1P22.pdf

[4] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024.

[5] W. Lyu, W. Ke, H. Sheng, X. Ma, and H. Zhang, "Dynamic downsampling algorithm for 3d point cloud map based on voxel filtering," *Applied Sciences*, vol. 14, no. 8, p. 3160, 2024.

[6] C. Han, Q. Zhao, S. Zhang, Y. Chen, Z. Zhang, and J. Yuan, "Yolopv2: Better, faster, stronger for panoptic driving perception," *arXiv preprint arXiv:2208.11434*, 2022.

[7] Navigation2 Project, "Navigation2 with gps," https://docs.nav2.org/tutorials/docs/navigation2_with_gps.html, 2026, gitHub Pages documentation, Accessed: Apr. 14, 2026.

[8] B. Peng, L. Zhang, and R. Xiong, "Smooth path planning with subharmonic artificial potential field," in *2024 International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2024, pp. 1027–1031.

[9] S. Alarabi, "Local path planning optimization for a mobile robot based on a potential field," Ph.D. dissertation, University of Detroit Mercy, Detroit, MI, USA, Oct. 2025.