



Intelligent Ground Vehicle Competition 2025
Indian Institute of Technology Madras
Team Abhiyaan



Vizhi
Design Report



I hereby certify that the development of the vehicle, Vizhi, as described in this report, is equivalent to the work involved in a senior design course. This report has been prepared by the students of Team Abhiyaan under my guidance.

Dr. Sathyan Subbiah
Faculty Advisor, Team Abhiyaan
Professor, Department of Mechanical Engineering
IIT Madras



SATHYAN SUBBIAH, Ph.D
Professor
Department of Mechanical Engineering
Indian Institute of Technology Madras
Chennai 600 036, India

Team Leads

Alan Royce Gabriel
Yash Purswani

Contingent Captain

Mahesh Gondi

Team Members

Mechatronics

Asher Abraham
Kairav Bharat Pathak
Krishna Murari Chivukula
Lalith Akash M
Nishanth Senthil Kumar
Raghav Y
Rohit Balaji
P.V.S Tushar
G.S.V.S Tejas
Vishal V
Vigit G.S
Sarvesh Mane
Ashwaat Tarun T S

Software

Bhanu Kiran Verma
Govind S Ashan
Krutarth Patel
Mahesh Gondi
Sahaj Shetty
Srinidhi V
Mahmood Sinan
Anoushka Kuriakos

Business and Design

Dikshansh Raipure
Tanmai Sonwane
Yuvraj Singh

Contents

1. Conduct of Design Process and Team Organization	1
1.1. Introduction	1
1.2. Organization	1
1.3. Design Assumptions and Process	1
2. System Architecture	1
2.1. Components	1
2.2. Safety Devices	1
2.3. Software Modules	2
2.4. System-System Interaction	2
3. Effective Innovations in our Vehicle Design	2
3.1. Mechanical	2
3.2. Electrical	2
3.3. Software	2
4. Description of Mechanical Design	3
4.1. Overview	3
4.2. Drive System	3
4.3. Suspension and Articulation	3
4.4. Chassis Design	4
5. Description of Electronics and Power Design	5
5.1. Overview	5
5.2. Power Distribution Architecture	5
5.3. Communication Systems	5
5.3.1. PID Control via Internal Encoders	6
5.3.2. Temperature Monitoring	6
5.3.3. Custom CAN Communication Stack and Benchmarking	6
5.3.4. Electromagnetic interference (EMI) Mitigation	6
5.4. Ambient Based Light Controller	6
6. Description of Software Design	7
6.1. Overview	7
6.2. Perception	7
6.2.1. Lane & Pothole Detection	7
6.2.2. Obstacle Detection	7
6.3. Waypoint Following	8
6.3.1. Lane Follower	8
6.3.2. GPS Following	8
6.3.3. Context Switching	9
6.4. Navigation	9
6.4.1. Costmap	9
6.4.2. Controller	10
7. Cyber Security Analysis	10
7.1. Modelling Threats and Analysing Impacts	10
7.2. Description of Cyber Protocol Implementation	11
8. Analysis of Complete Vehicle	11
8.1. Analysis of the Predicted Performance	11

8.2. Simulations	12
8.2.1. Computational Simulations using Ansys	12
8.2.2. Power Analysis	13
8.2.3. SIL Virtual Environment	13
8.3. Physical testing to date	13
9. Special Components of Autonav and Selfdrive	14
9.1. Self Drive	14
9.2. AutoNav	15
10. Initial Performance Assessments	15

1. CONDUCT OF DESIGN PROCESS AND TEAM ORGANIZATION

1.1. Introduction

Team Abhiyaan is a group of around 25 multidisciplinary undergraduate engineering students from the Indian Institute of Technology Madras (IIT Madras), united by a shared vision of advancing autonomous vehicle technology in India. Under the guidance of **Prof. Sathyan Subbiah** from the Department of Mechanical Engineering and with the support of the Centre for Innovation (CFI) — a premier student-led research facility at IIT Madras — we attempt to continuously push the boundaries of autonomous mobility. This year, we are proud to present Vizhi, an enhanced iteration of our previous autonomous vehicle - Vidyut. This report provides a comprehensive overview of the conceptualization, design, and development of Vizhi, detailing the innovations and engineering advancements that define this year’s vehicle.

1.2. Organization

The team is structured into three core modules: **Software**, **Mechatronics**, and **Business & Design**. Each module has members from all across the departments of IIT Madras. All the students are in their second year of study. This year, the Mechanical and Electronics modules have been consolidated into a unified Mechatronics module, fostering a more integrated approach to vehicle development. This restructuring enhances cross-domain collaboration, streamlines integration, and allows for an overall efficient degree of resource allocation, leading to a more cohesive development process. Over the past year, each member has dedicated 2–3 hours per day to the project, demonstrating a strong commitment to excellence in autonomy. Extensive use of **Notion** has been adopted for better organisational delegation.

1.3. Design Assumptions and Process

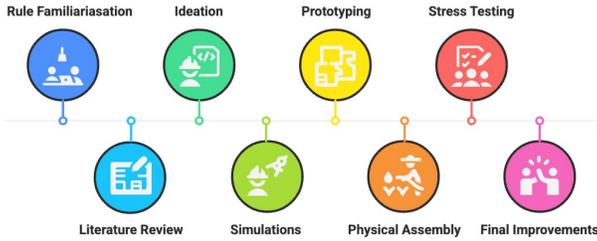


Figure 1. System Development Workflow of Team Abhiyaan

The design process follows a Linear Workflow, which begins with understanding IGVC rules and making assumptions about environmental conditions. A problem statement is defined, followed by a collaborative ideation phase addressing past failures. Feasibility is validated through simulations, components are selected based on performance and compliance, and subsystems are prototyped and tested. Final integration, optimization, and thorough documentation complete the flow.

2. SYSTEM ARCHITECTURE

2.1. Components

Mechanical	Electronics	Power	Sensor Track
Suspension & Gearbox	Motor Controller: BLVD-KBRD	24V Li-Ion Battery	RPLiDAR A2M12
Al-Extrusion Bars	MasterPCB	Power Distribution Submodule - Master PCB	ZED 2i Stereo Camera
3D-Printed Mounts and Joints	CANPCB	Inverter	Septentrio GPS Module
Oriental BLMR6400SK-GFV-B Motors	Oriental BLMR6400SK Encoders, TIVA TM4C123GH6PM MCU		ADT75ARZ (Temperature), MAX4081 (Current)

2.2. Safety Devices

The safety devices broadly comprise of an EStop System which Includes both a strategically positioned **wired E-Stop** for immediate local shutdown during anomalies and a **wireless E-Stop** using **LoRa** communication for safe remote intervention and a **Velocity Watchdog** which monitors velocity and sends an immediate kill command if safe speed limits are exceeded. A temperature-current cutoff auto-shuts down the system when the internal temperature of the MasterPCB exceeds a critical threshold of 100 °C or 25A. A **Heartbeat Monitor (CAN/RS-485)** maintains node connectivity across the comms network and issues disconnection warnings.

2.3. Software Modules

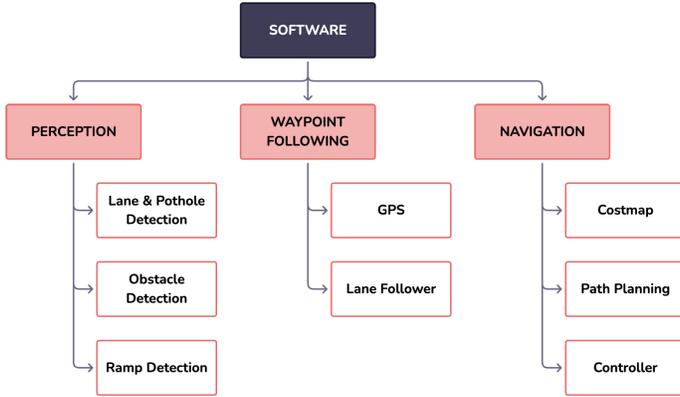


Figure 2. Software Organisation

The software stack on Vizhi comprises of the **Perception**, **Waypoint Following** and **Navigation** modules. These work parallelly to provide the most optimal goal point for the electrical stack to follow.

Perception uses thresholding techniques to obtain accurate information about the positions of the lanes, potholes and other obstacles. The waypoint following module uses this information to calculate an optimal goal for Vizhi to reach. A waypoint can be one of two things - a GPS coordinate, a goal calculated by the Lane Follower (6.3). We also use a simple algorithm to switch between GPS Following and Lane Following based on the proximity to a GPS waypoint. See 6.3.3

2.4. System-System Interaction

The robot features a lightweight yet robust aluminium extrusion chassis with trailing arm suspension for vibration damping. BLDC motors with gearboxes provide high torque, precise control, and weatherproofing. Custom 3D-printed joints are used to secure extrusion bar connections and sensor mounts hold sensors. A unified MasterPCB handles power distribution protection and communication using CAN across all subsystems. LiDAR, stereo camera, and GPS work together for mapping, lane detection, and navigation, with encoder feedback enabling accurate closed-loop control. A temperature sensor adds thermal safety, resulting in a compact, interconnected, and reliable robotic system.

3. EFFECTIVE INNOVATIONS IN OUR VEHICLE DESIGN

3.1. Mechanical

1. **Customised 3D Printed Parts:** Engineered custom 3D-printed clamps and precision-fit mounts for securing joints and components robustly. See 4.4
2. **Modular Chassis Redesign:** Reworked from a V-Nose to a Rectangular shape, incorporating drawers for greater modularity in storage. See 4.4
3. **Refined Drive System Architecture:** Transitioned from hub motors to conventional BLDC motors and adopted solid rubber wheels over tubed tyres. See 4.2
4. **Weatherproofing and LiDAR Cooling:** Laser-cut acrylic panels were used for durable weatherproof enclosures which were secured with screws for a tight seal. Integrated cooling fan in mount to regulate LiDAR temperature during extended outdoor operation. See 3

3.2. Electrical

1. **MasterPCB:** Designed a compact, centralized PCB combining power management and communication functionalities featuring overcurrent protection, ESD protection, and a more efficient coulomb counting method for accurate battery monitoring. See 5.2
2. **ProtoStack:** Implemented a modular in-house PCB setup developed for testing and validation including a Hotswap PCB and a Communication PCB. See 5.3
3. **Adaptive Gamma-Dimming Light Controller:** Designed an ambient light-controlled gamma dimming system using Solid State Relays to reduce power consumption. See 5.4
4. **Copper Shielding:** Constructed the GPS system with a copper mesh Faraday cage to reduce electromagnetic interference from external electronic noise and nearby devices. See 5.3.4

3.3. Software

1. **CUDA-Accelerated Lane Clustering:** Implemented DBSCAN clustering using CUDA-based GPU parallelization to leverage massive thread-level parallelism. This improved the speed of lane clustering by 10x. See 6.2.1

2. **Pointcloud Perception Stack:** Direct processing stereo-camera pointcloud to cluster potholes and lanes, bypassing intermediate image detection and perspective mapping steps. Using an IMU sensor to correct for ground plane inaccuracies. See 6.2.1
3. **Advanced Lane Follower** Fully mathematical lane-follower that dynamically computes safe, drift-resistant waypoints using polynomial lane fitting, queuing, and obstacle-aware goal correction—eliminating reliance on GPS. See 6.3.1
4. **Noise-Resilient GPS Estimation:** Developed a robust single-antenna GPS heading estimation method resistant to noise and interference, with a lightweight GPS-to-map frame transformation using heading. This approach improves the heading estimate as the vehicle moves further. See 6.3.2

4. DESCRIPTION OF MECHANICAL DESIGN

4.1. Overview

The mechanical subdivision of Team Abhiyaan’s Mechatronics module focuses on building a robust vehicle. With the merger of mechanical and electrical subsystems under Mechatronics, integration has become more seamless. This year’s design includes a new modular chassis, a transition from hub to drive motors, and improved cooling and weatherproofing techniques.

4.2. Drive System

Vizhi utilizes a **differential drive configuration**, similar to that of Vidyut. The chassis is equipped with two caster wheels—one at the front and one at the rear—and two centrally positioned drive wheels. Each drive wheel is independently powered by an **Oriental BLDC motor**, which comes integrated with an encoder and a gearbox. **Solid tires** are used to prevent issues associated with punctures and improve reliability.

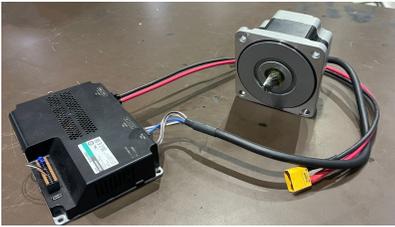


Figure 3. Oriental BLDC Motor



Figure 4. Drive Assembly



Figure 5. Overall Vizhi Drive

4.3. Suspension and Articulation

To support higher-performance motors, we **redesigned the suspension** system to suit Vizhi’s drivetrain. The caster wheel suspension was also rebuilt for compatibility. We used **Aluminium 6061** for its great strength-to-weight ratio, corrosion resistance, and ease of machining, making it ideal for strong, light parts.

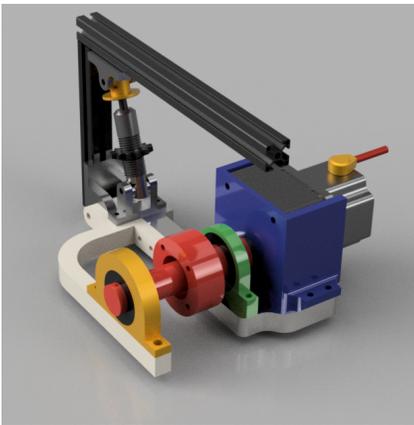


Figure 6. Trailing Arm Suspension

1. **Drive Wheel Suspension:** We chose a **trailing arm suspension** for Vizhi due to its proven performance, reliability, and space-efficient design. It keeps camber angle steady through motion, helping maintain tire contact and improving grip and stability on rough terrain or during sharp maneuvers.

We **fully redesigned** its geometry and structure to fit the larger conventional motors replacing the earlier hub motors. The new setup includes inner and outer trailing arms, a coupler, and dedicated mounting brackets to support the motors and bearings. The integration of standard motors did increase the overall footprint of the suspension system compared to the previous design. However, when benchmarked against alternative suspension types—our trailing arm configuration still remains more compact, mechanically simpler, and easier to manufacture.

2. **Caster Wheel Suspension:** To achieve uniform weight distribution and enhanced stability, caster wheels are placed at both the front and rear. The suspension rod was rebuilt for compatibility with Vizhi. These are **screw-in type swivel caster wheels**, each paired with a shock-absorbing spring that supports the robot's weight, cushions terrain impacts, and maintains wheel alignment.



Figure 7. Caster Suspension

4.4. Chassis Design

1. **General Structure:** This year, there has been a modification in the overall skeleton of the chassis of Vizhi. To incorporate ease of assembly, the V-Nose structure from last year's iteration Vidyut, has been discontinued and a **rectangular frame** has been adopted. **3-D Printed joints** were used for a more robust securing.

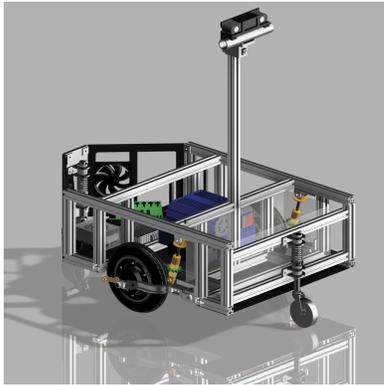


Figure 8. Vidyut-(2024)



Figure 9. Vizhi-(2025)

2. **Material Strategy and System Modularity:** Standardized **aluminium extrusion bars** of the chassis, engineered to be structurally robust, can support the entire system load while remaining lightweight to ensure enhanced maneuverability. Further efforts have been made to enhance the modularity of the system this year. A **drawer-based component housing** approach has been introduced, enabling organized placement and easy access to subsystems. This design modification not only streamlines integration and servicing but also improves the overall scalability and adaptability of the platform.



Figure 10. Weatherproofing

3. **Weatherproofing & Cooling:** A major drawback of last year's model, Vidyut, was inadequate weatherproofing, with hub motor failures mainly due to water ingress. To address this, **high-quality 5mm acrylic sheets** were used for better enclosure, and silicone sealant was applied to seal gaps, ensuring improved moisture protection. We also developed a **specialised 2-part mount** for the LiDAR to shield it from sunlight and prevent overheating, a known issue in Vidyut. To further tackle thermal challenges, a cooling system with **four high-volume 12cm axial fans** (max flow rate: 65.6CFM each) circulates air inside the central compartment. The PCBs are cooled using dedicated fans. The idea of using a ROS Stack-Integrated custom controller was dropped, as a constant power supply sufficiently reduced the fans' power consumption. Ideal fan positioning was determined through **CFD analysis on Vizhi**. See [8.2.1](#)

5. DESCRIPTION OF ELECTRONICS AND POWER DESIGN

5.1. Overview

Complementing the mechanical developments, Vizhi's electrical and electronic systems have undergone a redesign to support higher efficiency, improved integration, and enhanced reliability. This year's architecture involves some changes at the fundamental level.

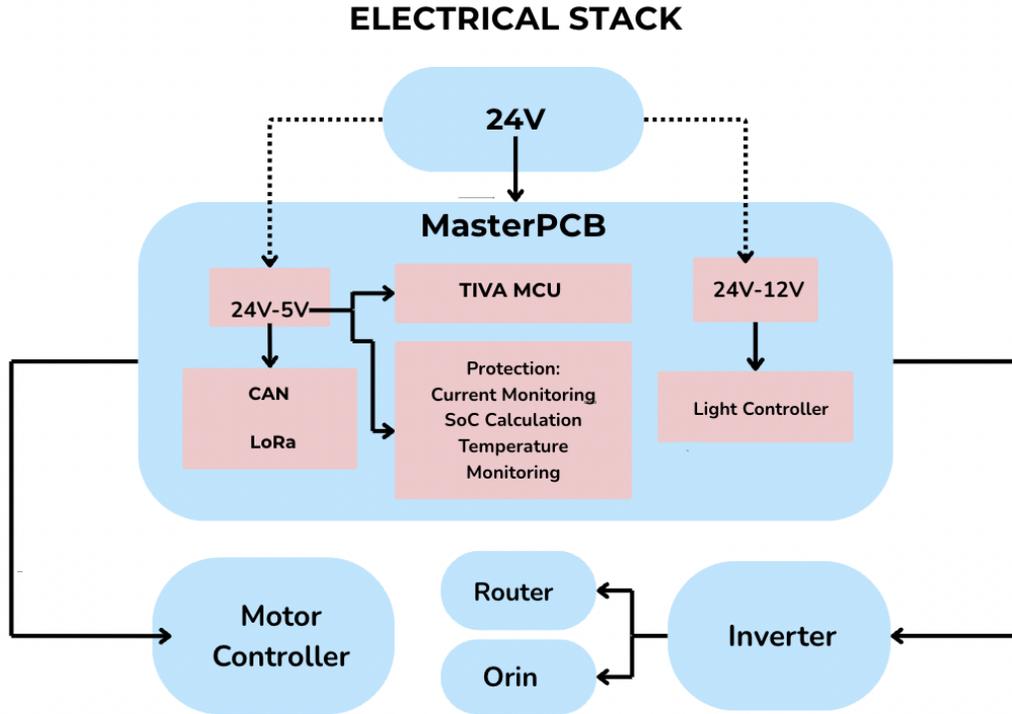


Figure 11. Overall Electrical Stack

5.2. Power Distribution Architecture

The power distribution system(PDS) uses a 24V Li-Ion battery coupled with a 24VDC-230VAC off-the-shelf inverter. The key driving component is the Master PCB, developed in-house. Primary features of the PDS include:

1. **Unified Drive + Battery Management PCB:** Built-in odometry of the Oriental BLDC motors eliminated the need for a separate Drive PCB. Its functionality was integrated into the MasterPCB, streamlining the architecture.
2. **Improved Coulomb Counting:** The MasterPCB uses precise op-amp-based current sensing instead of the Hall sensor used previously.
3. **Enhanced MasterPCB Features:** Includes reverse current and overcurrent protection, ESD protection via TVS diodes, and temperature monitoring.

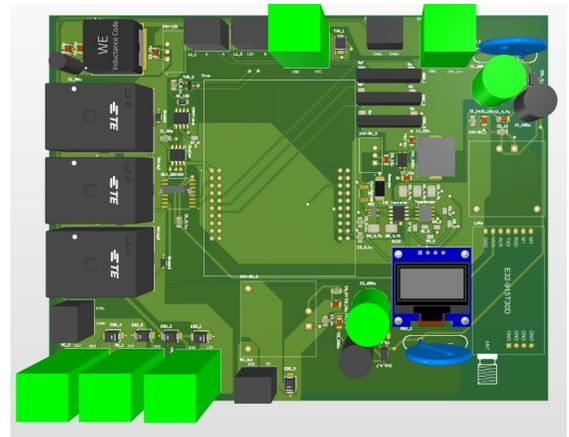


Figure 12. Master PCB

5.3. Communication Systems

A significant upgrade was made by replacing the serial communication system used in Vidhyut with a **Controller Area Network (CAN)** based communication system. This transition was enabled by the development of a dedicated **CAN PCB**, which is integrated with the Light Controller and later merged into the MasterPCB. CAN was chosen over

Serial communication due to its superior noise immunity from differential signaling, real-time performance with built-in Cyclic Redundancy Checking (CRC), and priority-based messaging ideal for critical safety signals. It also supports easy scalability with minimal wiring. Compared to USB Serial, CAN reduces CPU load on the Jetson Orin, prevents data drops at high baud rates while simultaneously lowering latency.

5.3.1. PID Control via Internal Encoders

The new Oriental motors use a **closed-loop internal PID** control system with built-in encoders for high accuracy. This eliminates the need for a dedicated external PCB for encoder data processing. Furthermore, it reduces CPU load on Orin as it only provides target speed as a setpoint to the algorithm. The internal encoders are directly mounted on the shaft, reducing misalignment and slippage. They also avoid latency, data loss, and USB buffer delays, resulting in smoother control with reduced overshoot and jerk during motion.

5.3.2. Temperature Monitoring

A temperature sensor communicates over **I²C** to continuously monitor the PCB temperature. If it exceeds a defined threshold, the system can automatically cut off power to prevent thermal damage. This real-time thermal protection is also visualized on an OLED display for onboard diagnostics. Using **I²C** minimizes wiring overhead while allowing both the sensor and display to share the same reliable communication bus.

5.3.3. Custom CAN Communication Stack and Benchmarking

A **Real-Time Operating System (RTOS)** has been implemented to manage communication between the drivetrain and Jetson Orin. **Transmit Process Data Objects (TPDOs)** and **Receive Process Data Objects (RPDOs)** have been used to exchange parameters such as motor velocity and position for efficient control and monitoring with minimal latency.

5.3.4. Electromagnetic interference (EMI) Mitigation

This setup addresses the challenge of electromagnetic interference (EMI), which significantly impacted GPS signal integrity in previous iterations. A copper cage has been implemented around the computational unit (Jetson Orin) to shield the GPS from it. Additionally, copper shielding has been applied to the ZED camera cables, which were identified as a major contributor to EMI.

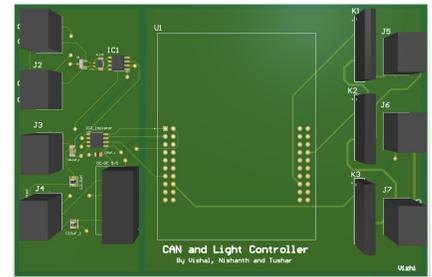


Figure 13. CAN PCB

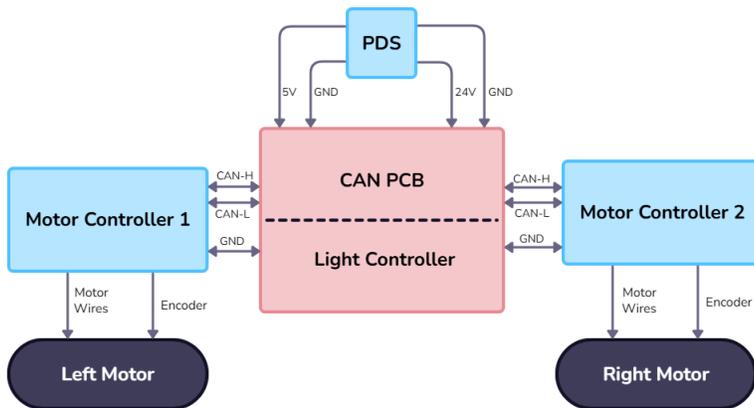


Figure 14. Communication Flow Chart

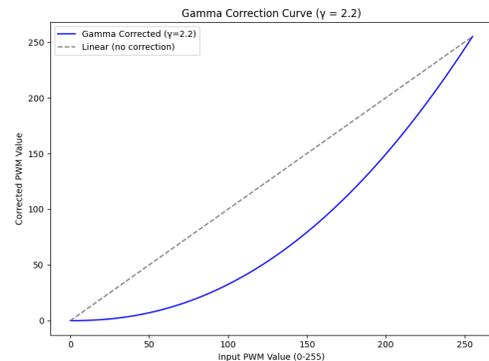


Figure 15. Gamma Correction Curve

5.4. Ambient Based Light Controller

The light controller system consists of a three-color ambient-light controlled LED strip regulated by **Solid State Relays** incorporated into the MasterPCB. The SSRs were chosen for their isolation. The brightness of the lights was adjusted using a gamma correction curve in accordance with the outside light, saving a significant amount of power. See 15

6. DESCRIPTION OF SOFTWARE DESIGN

6.1. Overview

The software stack is organized into a layered architecture. This allows us to update various parts of the stack independently of each other and gives us modularity while testing. The main parts of the pipeline include **Perception, Planning, Waypoint Following and Navigation.**

6.2. Perception

This module detects the lanes, potholes and obstacles around Vizhi to create a costmap of it's surroundings.

6.2.1. Lane & Pothole Detection

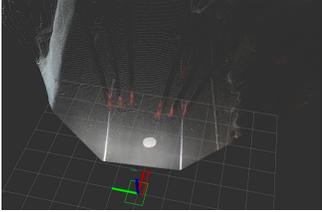


Figure 16. Pointcloud

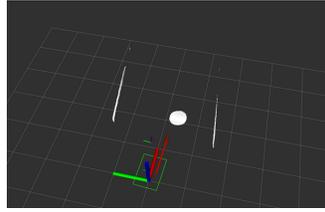


Figure 17. Filtering

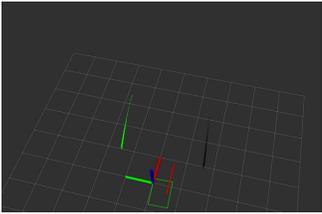


Figure 18. Classification

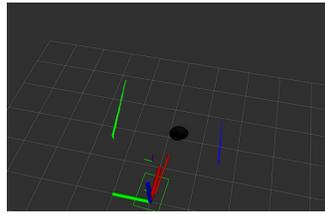


Figure 19. Clustering

This part of the perception stack comprises of the following modules running sequentially -

1. **Filtering** - The pointcloud received from the ZED2i stereo-camera is first filtered to obtain the white points on the ground plane. This is done using color and height thresholding directly on the 3D Pointcloud.
2. **Clustering** - Now these are passed to a custom implementation of the **DBSCAN** algorithm written with the **CUDA C++ library**. The clusters thus obtained now contain lanes and potholes in a mixed order.

Following are the techniques which were used to speed up our implementation:

- GPU memory buffer for pointclouds was fixed – **2x speedup.**
- Distance-aware density adjustment – smoother clusters.
- Warp-level synchronization with shared memory – **3x performance gain.**

The previous implementation had issues with **high latency**, hence this new and improved implementation bypasses the load of processing the points on the CPU, by leveraging the shared memory of the GPU.

3. **Shape-Based Classification:** Pointcloud clusters are projected onto a 2D image, and OpenCV contour detection is applied. Contours are classified as **potholes** or **lanes** based on:

(a) Aspect Ratio = $\frac{W}{H}$

(b) Area = A

(c) Extent = $\frac{A}{W \times H}$

A contour is classified as a **pothole** if:

$$1 < \text{Ratio} < 5, \quad 0.5 < \text{Extent} < 0.8, \quad \text{Area} > 200 \text{ px}$$

Otherwise, it is classified as a **lane**. Noisy clusters are filtered out based on point count. This heuristic shape-based method generalizes better than ellipse/circle fitting.

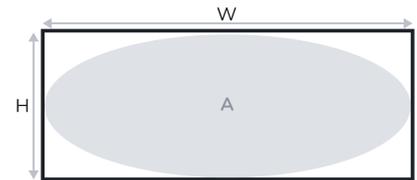


Figure 20. Pothole Classification

6.2.2. Obstacle Detection

Vizhi uses the 2D LiDAR sensor to detect obstacles around the bot. The detected obstacles are then added to a costmap, which is later used for path planning and navigation.

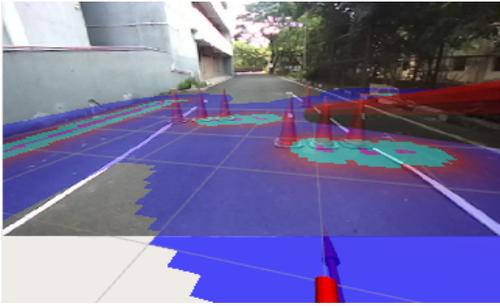


Figure 21. Obstacles

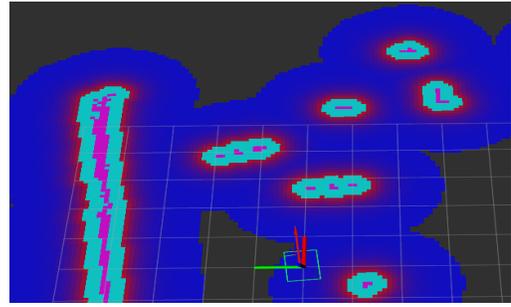


Figure 22. Costmap

6.3. Waypoint Following

6.3.1. Lane Follower

This module takes in the clusters from the perception module, and processes them to calculate a waypoint goal for the navigation stack to follow. The waypoint calculation is a purely mathematical process and it is ensured that wrongly calculated waypoints are corrected. It consists of the following submodules running sequentially:

Queuing —The two lane clusters are classified into two queues using their position relative to the bot. These queues form an occupancy grid of the lanes which ensures smooth and consistent polynomial fits for further use. Each queue has a fixed length, and old data is popped out of the queue once it is full. The queues are updated as regular intervals to prevent drift errors in the pointclouds without losing important information

Lane Fitting —A constrained polynomial is fit to the queues - $f(x)$. We constrain the curvature of the polynomial for an accurate extrapolation. The heading of the lanes is a linear fit on the queues.

Goal Calculation —We first find the intersection of the polynomial fit of the lanes with their headings calculated previously. Next, denote the farthest of these points from Vizhi as point P . A normal line is then constructed at point P to the first lane. The intersection of this normal with the second lane yields point Q . The calculated goal point is the midpoint of P and Q .

Goal Checker —This block checks for edge cases for the calculated waypoint. Using the costmap data, this first checks if the calculated waypoint lies on an obstacle. If so, it pushes the goal point along the direction of the lane heading until it reaches a low-cost region in the costmap. In the case that the calculated waypoint is either outside the local costmap, or too close to Vizhi, a "default" waypoint is published - this is a point along the lane heading at a fixed distance from the current position. Once Vizhi moves towards this waypoint, a new correct waypoint is calculated. Figure 23 shows our algorithm working on the data from Figure 34.

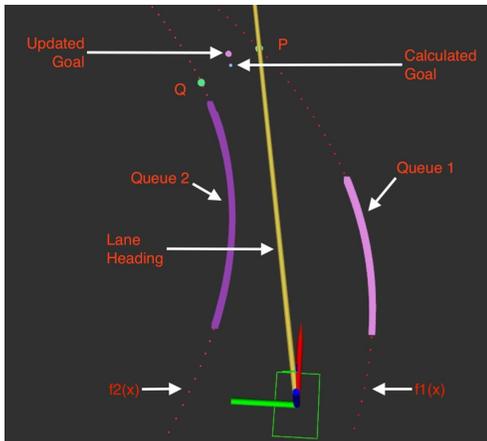


Figure 23. Waypoint Calculation

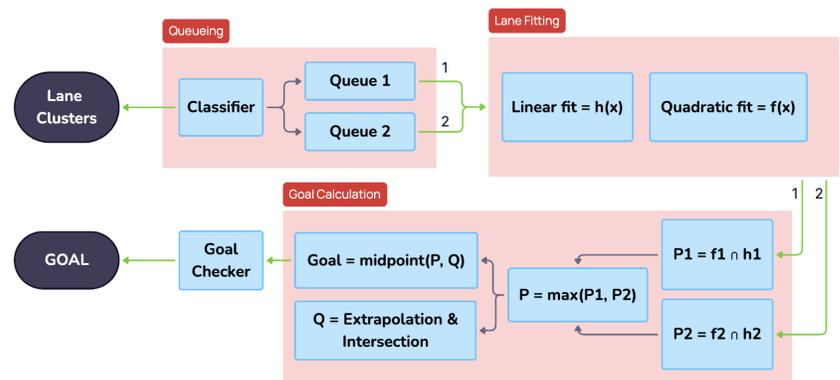


Figure 24. Lane Following Flow Diagram

6.3.2. GPS Following

The GPS localization process combines a single antenna GPS system with filtered odometry from the ZED2i camera.

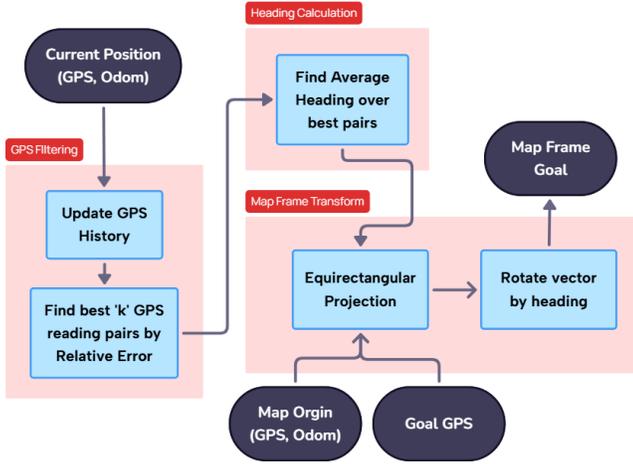


Figure 25. GPS Flow Diagram

GPS Filtering : To make our heading estimation robust to correlated errors and to account for sensor noise, a script records GPS readings at regular intervals of odometry displacement. For every reading, we store a tuple of (error, GPS, odom) and check this against all points in the GPS history.

To quantify the GPS horizontal positioning error relative to the odometry displacement, we define the following metric:

$$\epsilon_{\text{relative}} = \frac{\sigma_{\text{horz},1} + \sigma_{\text{horz},2}}{d_{\text{odom}}} \quad (1)$$

A pair with moderate sensor error is still acceptable if the odometry displacement is large. Whenever a new pair yields a lower relative GPS error, it is added to a sorted list of best pairs, which is then used for heading estimation.

Heading Calculation (H) : For each best pair, the heading difference between the GPS and odometry frames is calculated. The final heading is the average over all best pairs and is updated whenever a pair with lower relative error is found.

$$H = \frac{1}{N} \sum_{i=1}^N \left[\tan^{-1} \left(\frac{\Delta y_{\text{gps},i}}{\Delta x_{\text{gps},i}} \right) - \tan^{-1} \left(\frac{\Delta y_{\text{odom},i}}{\Delta x_{\text{odom},i}} \right) \right] \quad (2)$$

Map Frame Transformer : An **Equirectangular Projection** is used for GPS-to-map frame transformation as a computationally efficient alternative to the more complex Haversine formula. Given the short distances involved, the resulting approximation error is negligible.

$$x_{\text{goal}} = R \cdot (\lambda_{\text{goal}} - \lambda_{\text{map_origin}}) \cdot \cos \left(\frac{\phi_{\text{goal}} + \phi_{\text{map_origin}}}{2} \right) \quad (3)$$

$$y_{\text{goal}} = R \cdot (\phi_{\text{goal}} - \phi_{\text{map_origin}}) \quad (4)$$

where $R \approx 6,371,000$ m is the Earth's radius, λ denotes longitude, and ϕ denotes latitude. All angular measurements are in radians.

6.3.3. Context Switching

A simple algorithm is employed to switch between GPS following and Lane following. We have 4 GPS waypoints in the No Man's Land. Two goal points are calculated at all times - one from the Lane Follower (6.3.1) and one from the GPS waypoint publisher (6.3.2). The current position is checked for proximity to each of the 4 GPS waypoints provided. If Vizhi is closest to the 1st or 3rd waypoint, the waypoint following switches to GPS navigation. Else, it follows lanes.

6.4. Navigation

The heart of Vizhi's navigating stack is **Nav2** - a navigation framework built for ROS2. The various planners and controllers, along with the high level of customizability that Nav2 offers is leveraged to create a stack that complements our Perception stack well.

6.4.1. Costmap

The costmap comprises of three layers See 22 -

1. **Obstacle Layer (Light Blue):** The data from the 2D LiDAR creates this layer with the barrels and barricades. It interacts with the inflation layer to create a smooth roll off.
2. **Lanes and Potholes Layer :** The pointclouds containing the two lanes are added here. They are the outputs from the DBSCAN clustering algorithm.
3. **Inflation Layer (Dark Blue):** Both the above layers are inflated to create a safe region of variable radius around them. This is represented in the dark blue color.

6.4.2. Controller

The controller, being the most important part of the navigation stack, was stress tested extensively in both indoor and outdoor settings. It decides various parameters and limits for the velocities and accelerations, making sure the planned path is followed accurately and safely. After testing a variety of controllers, we settled on the **Graceful** and **DWB** controllers. This choice was made after careful assessment of smoothness and speed on various surfaces including indoor tiles and outdoor asphalt roads. Between these, we analysed the variations in linear and rotational velocity and chose to implement the **DWB** controller finally. Here is the comparison between the two controllers:

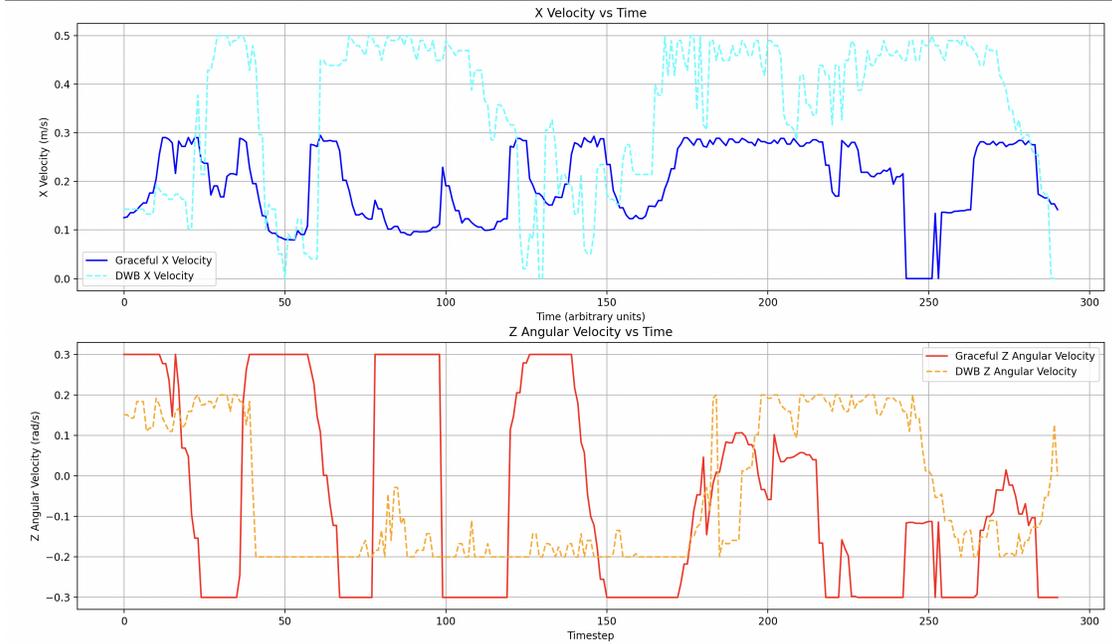


Figure 26. Controller Analysis

7. CYBER SECURITY ANALYSIS

For the Cyber-Security analysis a systematic process following the **NIST-RMF** framework was theorized. The amount of risk that the system can tolerate is analyzed and appropriate protections are enabled.

7.1. Modelling Threats and Analysing Impacts

Information	Threat Description	Integrity	Availability	Confidentiality
User Credential Data	Harvesting user identity for identity theft via malicious attacks.	High	Low	High
Debug Logs and Info	Allowance of debug information potentially leads to enough data for teleoperation of the bot by malicious individuals.	High	Moderate	Low
Wireless E-Stop	Potential to impair motion through wireless E-Stop commands.	High	High	Moderate
Odometry Readings	Erroneous odometry data affects localisation and hence potential relative positioning leading to inconsistent motion.	High	Low	Moderate
GPS Waypoint Data	Lack of GPS data for waypoints inhibit the motion of the bot in the right direction.	Moderate	High	Moderate

7.2. Description of Cyber Protocol Implementation

Implementation	Testing Strategy	Security Outcome:Protection and Mitigation
AC-1 (Access Control Policy)		
Only current team members are allowed to access the code on our system. Once someone leaves the team, their public keys are removed from the git repositories and computers used.	Github is used for version control with repositories containing only team members.	Ensures protection of user details, bot commands, and credentials. Unauthorized access from external actors is prevented, mitigating institute-external denial-of-service (DoS) threats.
SC-41 (Port and IO Access)		
Unused I/O ports are disabled. Usbguard for the Orin and bootloader authentication are enabled.	Any external device plugged into a port does not get detected.	Prevents rogue external devices (e.g., live USBs) from being booted or accessing system memory, thereby safeguarding bot credentials and debug data.
AC-4 (Information Flow Enforcement)		
The bot connects only over HTTPS. Unused ports are blocked on Orin using UFW. The www-data account is restricted to specific IPs, with all other traffic filtered and logged. HTTP is entirely disabled.	Encrypting vital information like estop and using rolling keys. A temporary switch to HTTP causes bot disconnection; switching back to HTTPS restores connection.	Secures debug information, credentials, code, and telemetry. Unauthorized payloads and remote shell attacks are blocked by restricting insecure and unsolicited network traffic.
SC-13 (Cryptographic Protection)		
The Orin's SSD, communication, and user data are encrypted to prevent key extraction by anyone with physical access.	Booting Orin requires a decryption key. Rolling keys are used for communication to prevent tracking.	Ensures user and usage data remain inaccessible without decryption keys, protecting against physical theft or unauthorized access to stored data.
IA-5 (Authenticator Management)		
Login is secured with strong passwords and fail2ban , which blocks brute-force attempts via entry limits, IP bans, and delays.	Audit ID and monitor SSH activity for unauthorized access attempts.	Protects bot credentials, user data, and source code by mitigating brute-force login attempts and preventing unauthorized decryption. Encryption keys are not stored in plaintext.

8. ANALYSIS OF COMPLETE VEHICLE

8.1. Analysis of the Predicted Performance

Vizhi was extensively tested to identify failure points and improve upon existing algorithms. The testing occurred in two stages -

- Electrical-Chassis Integration:** Tuning PID, testing and validating the electrical stack with manual control.
- Software Testing:** Emulating competition conditions to test various modules in the stack to improve algorithms for lane following and obstacle avoidance.

Feature	MAX4081(Shunt Resistor Based)	Hall Sensors
Accuracy	High($\pm 0.1mV$ offset) with Low Drift	Moderate($\pm 1-3$ % error)
Temperature Stability	Offset of < 10 ppm/ $^{\circ}C$	± 1.5 % full scale drift
Signal to Noise Ratio	High	Low
Environmental Stability	Not susceptible to magnetic interference	Highly susceptible to magnetic interference

Table 2. Comparison between MAX4081 and Hall Sensors for SoC-based Current Sensing

Parameter	Predicted Performance	Trial Data	Notes/Analysis
Speed	2.5 miles/hr	1.9 miles/hr	Speed was limited during indoor testing to prioritize tuning of navigation parameters.
Ramp Throttle	75 units	50 units	Control Parameters to be tuned
Obstacle Avoidance	Keeps sufficient distance to an obstacle based on the inflation layer of costmap.	Proximity to obstacles while navigating, changed depending on controller used.	The delays between the motor encoders and navigation stack caused turning to be in a haphazard motion.
Obstacle Detection Distance	The LiDAR sensor has a 12 metre range. Local cost map is a 30x30 grid.	Local cost map populates with obstacles correctly. LiDAR fails to detect some specific reflective obstacles at higher distances.	Local cost map populates with obstacles correctly. LiDAR fails to detect some specific reflective obstacles at higher distances.
Battery Life	1.75 hours	2 hours(nominal) and 1.53 hours(extreme)	Matches predictions under nominal load; drains faster during high compute tasks.
Waypoint Arrival Accuracy	Within 0.5 m	Avg. offset 0.45 m	Controller tolerance to be tuned
Complex Obstacle Handling	Recovery behaviour executes, resulting in a new trajectory being planned.	Recovery behaviour executes, resulting in a new trajectory being planned.	A simple goal checker and updater was written to reduce the necessity of recovery behaviours. See 6.3.1

8.2. Simulations

8.2.1. Computational Simulations using Ansys

A range of simulations, including Computational Fluid Dynamics (CFD) and Finite Element Analysis (FEA), were conducted to benchmark the performance of the mechanical components used in the design.

Structural Analysis of Suspension: Trailing Arms are integral structural components of the suspension system, susceptible to the highest loading. All structurally critical components have been developed with a minimum safety factor of 2 and have been optimized for overall rigidity. Ansys Static Structural has been used along with Fusion360 software to design the optimized parts.

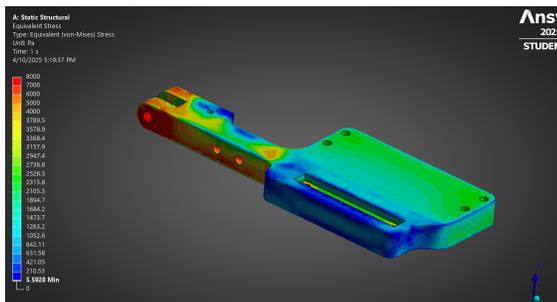


Figure 27. Inner Trailing Arm

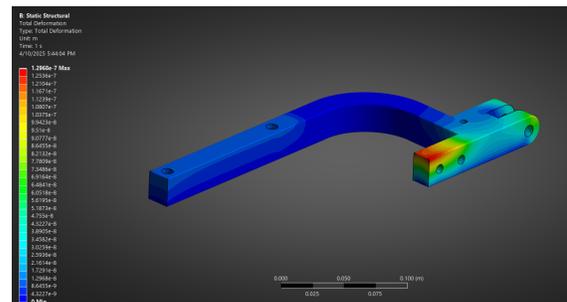


Figure 28. Outer Trailing Arm

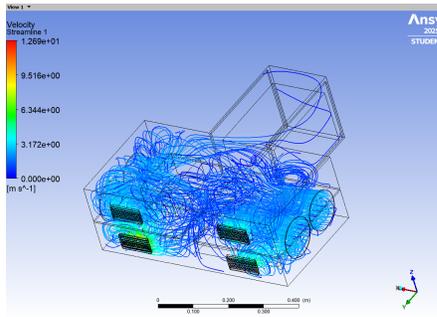


Figure 29. Cooling using Fans

Computational Fluid Dynamics(CFD) Analysis: As described in 4.4, 4 high-volume flow rate axial fans have been used to circulate air inside the main compartment. Airflow was modeled using Ansys Fluent and simulations were analyzed for fans having different speeds varied from angular velocity of **40 to 2050 RPM**, to visualize the airflow distribution in the vehicle. Airflow and components were placed accordingly, optimizing airflow requirements for each component and weight distribution of the vehicle.

8.2.2. Power Analysis

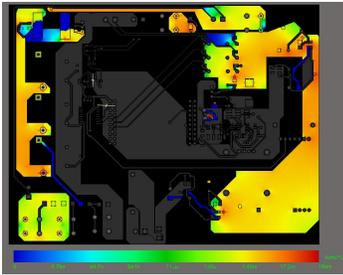


Figure 30. Current Density Heatmap

As a result of increasing copper weight from 1oz to 4oz on traces in the MasterPCB, the current density has reduced remarkably from an average of over $1A/mil^2$ to about $10mA/mil^2$. Fan cooling mechanisms have been employed further to reduce the temperature of the PCB. This **substantial decrease in current density** minimizes resistive losses and significantly improves the board's thermal performance. As a result, the system now operates more reliably under higher current loads with reduced risk of trace degradation or thermal failure.

8.2.3. SIL Virtual Environment

Software strategies and algorithms were extensively tested in simulation using Gazebo and **RViz** before deploying them on the physical robot. **Custom Gazebo** worlds were created to accurately replicate the Autonav and Self-Drive courses at IGVC, based on the provided sample course layout.

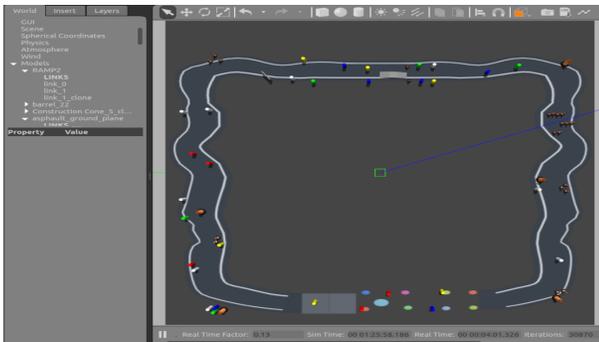


Figure 31. AutoNav Simulation

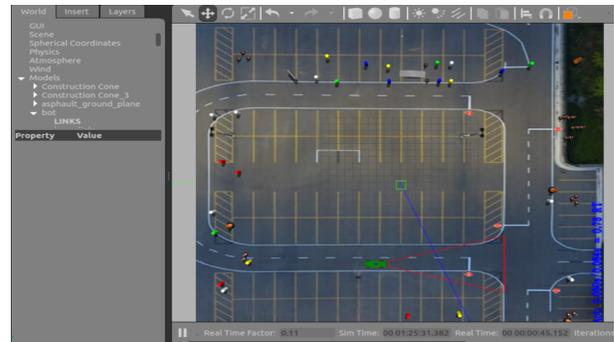


Figure 32. Self Drive simulation

8.3. Physical testing to date



Figure 33. Outdoor Testing

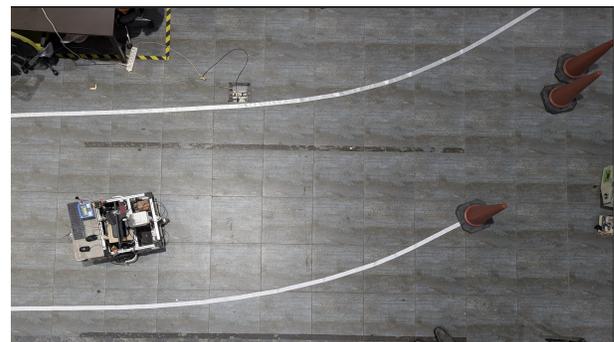


Figure 34. Indoor Testing

Failure Point	Cause	Resolution
Inaccurate GPS heading	Sensor noise and poor signal near buildings or under trees	Averaged over the best k pairs filtered by sensor error; enclosed electrical components, camera wire in copper meshes. See 6.3.2
Lane following failure	Minimally visible or obscured lanes due to obstacles or sharp turns	We maintain a running queue of lane points (cleared dynamically) to obtain a smoother curve fit. See 6.3.1
Wrong classification lanes in presence of potholes	The potholes were assumed to exist in lane - free regions	Clusters received from the DBSCAN output are classified into lanes and potholes before using them for following. See 6.2.1
Goal coordinates lie on obstacles	The mathematical approach to calculate waypoints from lanes did not consider the costmap	Dynamic goal checking function using values from Nav2's costmap to adjust goal co-ordinates to cost - free regions. See 6.3.1
Lane thresholding resulted in patchy detected lanes.	Camera tilt and changing light conditions (glare)	Corrected for tilt in ground plane and noise using an IMU sensor to detect camera motion. Implemented a GUI to dynamically set threshold ranges for ease of testing. See 9.2
Battery discharging unexpectedly	Missing battery level indicator	Implemented Coulomb Counting for battery level indication. See 5.2
Bending in chassis due to stress	L clamps loosen under vibration or stress	Introduced 3D printed mount at major load bearing joints. See 4.4
Overheating of LiDAR while testing outside	No protection against the sun	Designed and 3D printed a LiDAR roof to provide shade and cooling. See 3

9. SPECIAL COMPONENTS OF AUTONAV AND SELFDRIIVE

Parts of the software stack are unique for the Autonav and Selfdrive challenges. This includes Pothole Detection and Ramp Detection for Autonav, a modified Lane Detection stack and Object Detection for Selfdrive. The lane following stack is the same for both competitions.

9.1. Self Drive

Dual Camera Mount: Specific to the Self Drive Challenge, a custom dual camera mount was manufactured. It involves a variable angle setup, wherein the field of view can be adjusted, unlike the mount used for AutoNav which only faces one direction. New mounts were also designed and manufactured for the E-Stop and the new LiDAR. Required space is provided in both mounts for wiring purpose. See [35](#)



Figure 35. Dual Mount

Lane Detection: A new lane detection stack was created to consider the addition of the centre dashed lane. A Hough Line Transform is run on the filtered pointcloud containing lanes. The returned lines are clustered into 4 groups - Left Lane, Right Lane, Dashed Lane (extrapolated), Horizontal Line (at intersection). The clusters are then published on separate topics for further use. See [36](#)

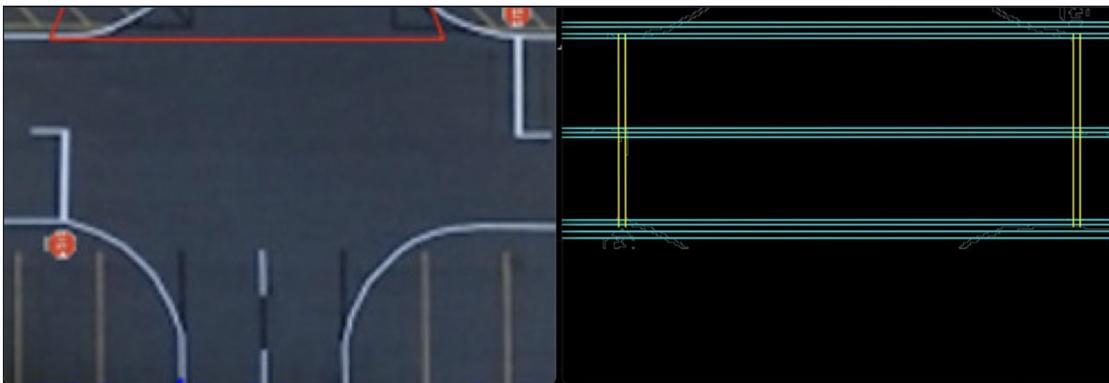


Figure 36. Detection of intersection

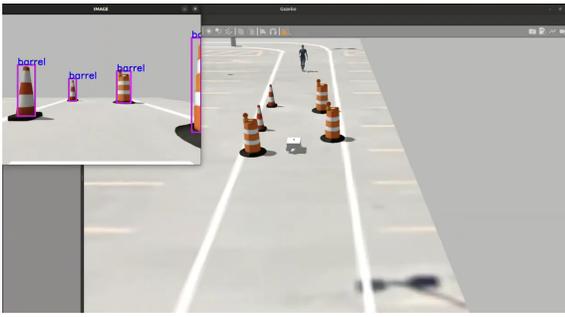


Figure 37. Object Detection with YOLOv10

Object Detection: A YOLOv10 model was trained in-house to detect cones. This model was trained on a custom dataset made by scraping data from around 30 publicly available datasets. The model is trained to detect cones, barrels, stop signs and people. YOLOv10 was specifically chosen for its high accuracy and fast inference speeds.

9.2. AutoNav

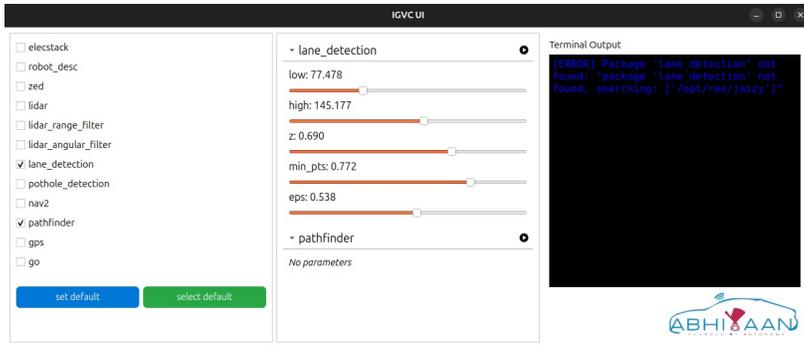


Figure 38. IGVC GUI

Custom GUI: We built a custom GUI for our stack using Qt. This lets us launch processes modularly, and makes independent testing of different parts of the stack more convenient. The various important parameters can also be modified dynamically using sliders. This helps Vizhi adapt faster to a newer testing environment requiring different parameters for Perception and Navigation. It also creates log files for each process launched, thus facilitating faster debugging.

Ramp Detection:

RANSAC (Random Sample Consensus) is a robust algorithm for model fitting in noisy data. For ramp detection, we used RANSAC to identify planar surfaces in point cloud data by iteratively fitting planes to randomly selected triplets of points and evaluating inliers within a threshold. The best-fitting plane with the most inliers is selected, and a minimum threshold on the plane's normal vector angle ensures that only inclined planes characteristic of ramps are retained, filtering out irrelevant surfaces.

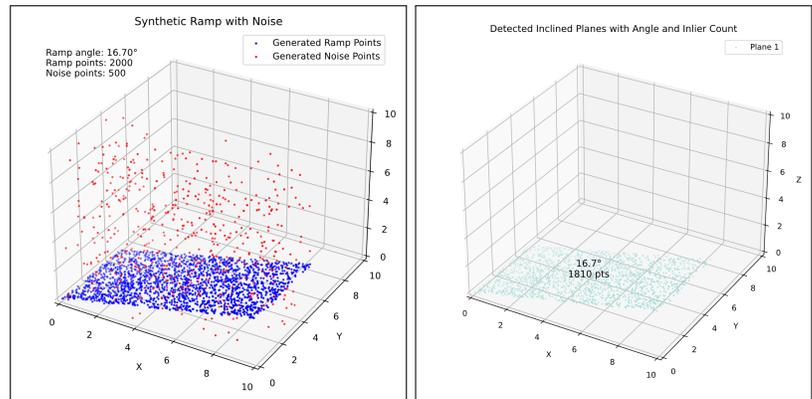


Figure 39. Ramp detection using RANSAC on point cloud data; Detected ramp surfaces

10. INITIAL PERFORMANCE ASSESSMENTS

To date, the vehicle has demonstrated reliable performance in controlled testing environments. Key systems, including localization, obstacle detection, and autonomous navigation, have operated within expected parameters. GPS waypoint tracking shows an average positional error of less than 1.5 meters, and lane-following algorithms maintain stability at speeds up to 2 m/s. Minor tuning remains for Nav2 and lane perception. Overall, initial testing validates the core design objectives and indicates strong readiness for full-course trials.