Intelligent Ground Vehicle Competition 2024

Manipal Academy of Higher Education
Project Manas

Design Report

NOVA

I certify that the development of the vehicle, Nova, described in this report is equivalent to the work in a senior design course. The students of Project MANAS have prepared this report under my guidance.

Ashalatha Nayak
Professor
Department of Computer Science & Engineering
Date: 15.5.24

DEPT. OF COMPUTER SCIENCE & ENGINEERING
MANIPAL - 576 104
MIT

Team Members

Artificial Intelligence

Aaryan Panigrahi
Aditi Joshi
Anirudh Agarwal
Chinmayi Bhat
David Jijo
Subham Patra

Sensing and Automation

Animesh Patil
Naman Kumar
Natasha Gonsalves
Siddharth Matkar
Sparsh Srivastava
Vansh Yadav (Team Captain)

Mechanical

Anuraag Raut
Gautham Nair
Jaival Shah
Manjyot Saini
Nirbhay Aher
Zain Hussain

1. DESIGN PROCESS AND TEAM ORGANIZATION

1.1. *Introduction*

Project Manas, the official AI and robotics student team of *Manipal Academy of Higher Education (MAHE)*, proudly presents *Nova*, our fourth entry into the *Intelligent Ground Vehicle Competition (IGVC)*. With guidance from our seniors, support from our university and sponsors, we have spent the past year developing *Nova*. Building on the insights gained from last year's autonomous bot, Otto, and integrating new advancements, *Nova* represents a comprehensive upgrade in every aspect.

1.2. *Organization*

Project Manas is a team comprising 42 undergraduate students from various engineering disciplines, organized into four subsystems:

- ***Artificial Intelligence:*** Designs and develops the software stack.
- ***Sensing and Automation:*** Manages power distribution, motor controllers, and the sensor suite.
- ***Mechanical:*** Handles mechanical design and manufacturing.
- ***Management:*** Oversees financials, logistics, PR, communications with the college and sponsors.

Each subsystem is led by a senior board member who guides and supervises its operations. The team is managed by a board comprising the Subsystem heads, the Team Manager, Technical Head, R& D Head. Integration of all these subsystems is ensured through regular team meetings, facilitating collaboration and communication.

1.3. *Design Process and Assumptions*

Our design process follows a modified version of the Rapid Application Development (RAD) model, characterized by continuous testing, analysis, and redesign of the bot's architecture. Initially, we establish design assumptions about the environment in accordance with IGVC rules and define the problem statement based on competition requirements.

Following extensive research, we divide the design process into subsystems, integrating these individual designs into an overall architectural framework. This prototype undergoes rigorous testing in both simulations and real-world conditions to identify and address failure points. We repeat this process until the bot passes acceptance tests and meets our satisfaction criteria.

We completed the course of design and implementation over a span of nine months, with every member averaging roughly 540 hours of work on the bot. A total expenditure of 8400 USD was incurred, purely for the acquisition, maintenance, and replacement of components.

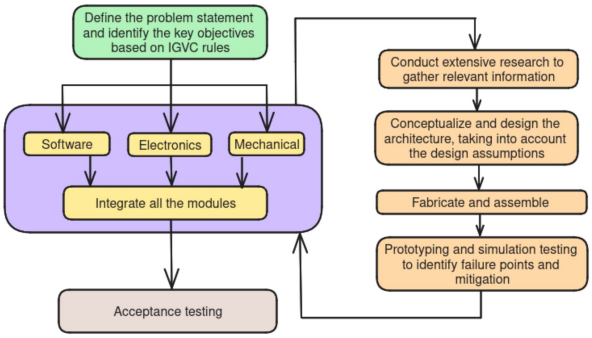

Figure 1. Design Process

2. SYSTEM ARCHITECTURE

2.1. *Significant mechanical, power, and electronic components*

Building on last year's design, we have modeled the power architecture to ensure an efficient mapping of need to consumption. Other mechanisms, such as current and voltage draw analysis provide additional data, helping the bot get a better idea of its surroundings. Vibration damping systems which include carbon fiber support rods and grommets for the electronics along with a heightened sensor pole to provide better visibility for the sensor suite, these components allow for a better traversal of the course.
The electronics module consists of the sensor suite, power architecture, drivetrain accessories, supportive wiring and organization of the system.

Computer and Controller Boards	**Sensor Suite**	**Power Architecture**	**Communications**	**Safety System**
Asus TUF F15	ZED2i Stereo Camera	Battery	CANine v1.4 CAN adapter	Serial Data monitor (watchdog)
STM32F446RE MCU	BNO055 IMU	Power Distribution board	USB hub	Battery Power Monitoring System
STM32F401	Mile incremental encoders	300W Inverter		Fuses
TinyMovr R5.2	Ouster LIDAR			MCB
	Current and Voltage sense			Estop
	NEO-M9N GPS module			

2.2. *Safety Devices*

Nova prioritizes safety by utilizing devices and algorithms for system failures. Learning from last year's design, we have factored in current draw safety, protecting motor controllers and sensors. The Power Distribution Board (PDB) has a 10A and 5A fuse for sensor power rails and accessory components respectively. Also, motor controllers have current limits to prevent overcurrent damage.

Nova includes an onboard Emergency-Stop (E-Stop) with a wireless trigger mechanism, following guidelines. When triggered, the wireless E-Stop deactivates the Solid State Relay (SSR) and cuts off power to the motors.

Nova also features LED strips which indicate the state of its autonomy and motion. They are switched by a relay, which is controlled by the sensor board.
The STM32 Nucleo interfaces via ROSSerial to receive the state, and switches between the following modes:

- ***Solid:*** When *Nova* is in manual mode.
- ***Blinking slow:*** When *Nova* is in autonomous mode.
- ***Blinking fast:*** When *Nova* is reversing.

A system crash or disconnection of a serial port, which results in stoppage of data flow, will trigger a watchdog to cut power from the motor controllers.

2.3. *Significant Software Modules*

Our software stack has been divided into various modules based on their functionality and purpose.

1. ***Perception:*** Responsible for detection of lanes, potholes, and obstacles using specific algorithms for each task.
2. ***Localization:*** Aggregate odometry data from multiple sources to generate an accurate pose estimate, immune to noise.
3. ***Mapping:*** Creates an occupancy grid by merging inputs from perception and localization modules for a dynamic depiction of the robot's surroundings.
4. ***Goal Selection:*** Selects intermediate goals for the bot based on the map, adjusting the method according to the bot's position on the course. A control node switches between two methods.
5. ***Navigation:*** Employs path and motion planning algorithms, based on goals received, to produce an optimal trajectory for the bot to efficiently reach the target.

2.4. *Integration of significant components and safety devices*

Sensor data collection is coordinated through an onboard monitoring system, with each sensor having its own ROS package for integration, utilizing signal processing techniques for noise mitigation. A significant upgrade is the adoption of ROSSerial for streamlined sensor interfacing into the stack.

The software stack processes sensor inputs received via ROS topics to plan the bot's trajectory. The command velocities are fed to the motor controllers, whose algorithm maps them to RPM's in ticks per second. A transformation, taking into account axle length and wheel radius, is applied to the velocity commands,which are used to reach the target setpoint.

The safety systems mentioned earlier have been rigorously stress tested to address various failure modes. The new PDB includes automotive fuses and Miniature Circuit breakers (MCBs) paired with the motors to monitor current draw and ensure safe operation. A watchdog mechanism is implemented to immediately stop *Nova* in case of system failure.

3. EFFECTIVE INNOVATIONS

3.1. *Mechanical*

3.1.1. *Vibration dampening plate for electronics*

The caster wheels produce high frequency vibrations on uneven terrain which can damage electronics. To prevent this, our design team has explored the use of 3D printed dampeners crafted from Thermoplastic Polyurethane (TPU) filament, known for its elastomeric properties. We used TPU rated at 95 A on the Shore A hardness scale, which resulted to be sub-optimal. Consequently, the design approach shifted towards using commercially available vibration damping grommets engineered to isolate sensitive equipment from vibrations and shocks.

3.1.2. *New outer covering*

The materials evaluated for the body panels were Carbon Fiber, Polyvinyl Chloride (PVC), High Density Polyethylene (HDPE) and Wood. Ultimately we opted for PVC as the primary material due to a combination of factors. While other materials offered potential benefits, they ultimately fell short in some key areas.

Material	**Carbon Fibre**	**PVC**	**Wood**	**HDPE**
Water Resistance	✓	✓		✓
Temperature, Flame Resistance	✓	✓		
UV Resistance	✓	✓		✓
Chemical Resistance	✓	✓		✓
Ease of Fabrication		✓	✓	✓
Availability		✓	✓	
Availability		✓	✓	✓
Cost Effective		✓	✓	✓
Recyclable		✓	✓	✓

Table 1. Material comparision

3.1.3. *Configurable Wheel Options*

Nova's design is adaptable to both indoor and outdoor applications. Large wheels can be mounted outside the frame for higher ground clearance and stability in outdoor conditions. This also keeps internal components away from dust and water. Alternatively, the internal wheel wells can accommodate up to 12 inch wheels in a compact form for indoor applications.

Figure 2. BDC Motor configuration

Figure 3. Maxon Motor configuration

3.2. *Electronics*

3.2.1. *ROSSerial*

The incorporation of ROSserial into our communication architecture served as a major improvement compared to our previous iteration. It serves as an interface between our sensor suite and the software stack, allowing for full duplex communication. This implementation enables sensor data communication in the required formats with the software stack, which is built on ROS, and ensures reduced latency.

3.2.2. *FreeRTOS Multithreading for Concurrent Topic Communication*

Integrating FreeRTOS into our system allows for efficient multithreading, enabling simultaneous publishing and subscribing to multiple topics within our ROS-based architecture. Using FreeRTOS, we achieve an efficient framework that ensures timely data exchange among different modules, enhancing overall system responsiveness.

3.3. ***Software***

3.3.1. *Attention Res-UNet*

A custom CNN was created to segment lanes, utilizing attention and residual blocks in the architecture which allows the environmental context to be encoded into the model's decision making. Testing in scenarios with inconsistent lighting exposed the shortcomings of color filtering, while conventional deep learning methods proved ineffective on highly curved lanes, making our approach the most robust and adaptable solution for most situations.

3.3.2. *Goal Generation*

We've designed a U-Net architecture to generate goal points for navigation. The inputs to the model include a map of its immediate surroundings and a long-term waypoint that the intermediate points should trend toward . Skipped connections are used to focus on lanes across the model. Since an ideal goal is one that takes into account both the subtleties of local environment with a long-term idea of the final goal at hand, a deep learning approach, with its learnable parameters works best.

3.3.3. *Remote Operability*

We had designed a web application to publish cmd_vel commands to *Nova* and stream live from *Nova* via connection over a VPN server. This adds to ease of use in terms of tele operating capabilities of the bot. Live video stream from the bot is displayed on the app, based on which users can teleoperate the bot.

4. DESCRIPTION OF MECHANICAL DESIGN

4.1. ***Overview***

The Mechanical subsystem creates a functional bot model for testing by other subsystems. Fusion360 and Ansys are employed for component design and simulations, ensuring structural integrity under various stresses. Manufacturing tools include waterjet cutting, radial drilling machines, FDM 3D printing, and a mitre saw. The physical dimensions, weight and torque capabilities makes sure that *Nova* is able to complete the obstacle course with ease.

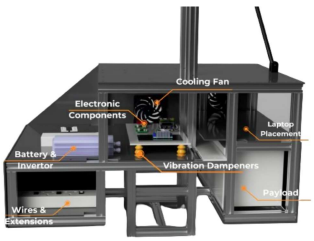

Figure 4. *Nova's* Structure

Vehicle Mass	70 Kg
Vehicle Length	950 mm
Vehicle Width	630 mm
Vehicle Height	1550 mm
Max Torque	14 N-m

Figure 5. *Nova's* specifications

4.2. ***Structure***

T-slots are engineered channels used in machinery to mount components securely and adjustably, enhancing operational flexibility. *Nova* uses aluminum as the T-slot material. With cross-sectional dimensions of 30x30 mm, they use aluminum corner brackets for strength without adding much weight. Their availability, cost, and modularity make them an excellent choice for industrial applications.

4.2.1. *Minimum Footprint*

Nova's compact footprint (3ft x 2ft) enhances agility and maneuverability for obstacle avoidance, providing a less cluttered environment for flexible navigation. This design also broadens application potential in confined spaces such as hospitals and warehouses.

4.2.2. *3D Printed Motor Supports*

To prevent strain on motor mounts from longer-bodied motors, 3D-printed Polyethylene Terephthalate Glycol (PETG) brackets are used. These brackets distribute weight across the chassis and act as shock absorbers on uneven terrain, reducing impact transfer to the motor shaft and minimising damage to internal components.

4.2.3. *Vibration Modulation*

During testing we found high frequency low amplitude vibrations. To counter this, the chassis uses lightweight 3K 200GSM 2×2 twill carbon fibre rods for stable sensor mounting, tube rubber tires with symmetrical block treads for better traction and reduced noise, and leaf spring hammer nuts instead of traditional ones to mitigate vibrations.

4.3. ***Drivetrain***

Nova's drivetrain features centrally mounted wheels, utilizing a shaft supported by bearings for power transmission. The motor interfaces directly with the shaft, resulting in a compact drivetrain assembly, with two caster wheels, one in the front and one at the back for enhanced stability and turning.

Figure 6. Framework of *Nova*

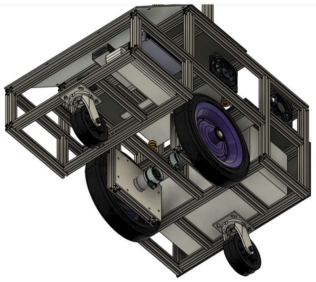

Figure 7. Drivetrain

Nova's design incorporates a central axis of rotation, offering several advantages for improved navigation capabilities:

- ***Enhanced Maneuverability:*** Enables more efficient navigation during turning compared to traditional non-central axis of rotation by eliminating the need to account for body swing. This translates to smoother, more predictable movements and reduces potential errors in estimating *Nova's* position, and navigation through tighter spaces.
- ***Improved Sensor Accuracy:*** Minimized frame vibrations during rotation, enhances sensor data accuracy for precise obstacle detection, mapping, and localization.The design's lower moment of inertia enables quicker stops and reduced overshoot, improving precision in position estimation.

4.4. ***Suspension***

4.4.1. *Active Suspension using Linear actuator as a design prospect*

Since the course is built on an asphalt track, we noticed that a dynamic suspension is not relevant and hence this idea was abandoned.

However, when *Nova* engages with the ramp the powered wheels lose contact with the ground. To counter this we intend to use a linear actuator for height adjustment of the castor to ensure that the powered wheels stay on the ramp. Adjusting the front caster wheel's height with a linear actuator could provide two main benefits. The first one being dynamic stability adjustments and the second one being optimized traction during ramp navigation or incline traversal.

4.5. ***Significant Design Benefits***

4.5.1. *Motor Quick Swap*

The quick-swap motor system on *Nova* enhances adaptability by using motors mounted on aluminum plates with adjustable T-slots, allowing for rapid installation of various planetary gearbox motors. See Figure 8 and Figure 9

Figure 8. With Maxon Motor

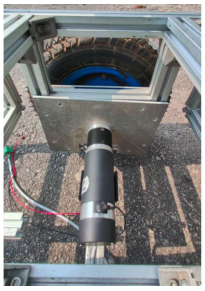

Figure 9. With Rhino Motor

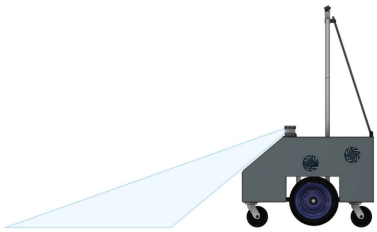

Figure 10. Lidar's unobstructed visibility

4.5.2. *LIDAR mount and Zed Cam mount*

Nova's design ensures unobstructed sensor visibility for accurate data, reducing blind spots and unexpected behavior. A modular setup allows for optimal ZED camera placement. The front panel's slope accommodates the LIDAR's ±22.5-degree and ZED Cam's 70-degree fields of view, minimizing post-processing and reducing noise and false positives. See Figure 10

4.5.3. *Coupling*

We employ wheelbarrow wheels for durability and easier maintenance. M8 bolts arranged perpendicular to each other are used for power transfer connecting the shaft to the wheel, effectively distributing forces across two shear planes to mitigate stress concentration. Two bearings per shaft are used to prevent excessive stress concentration on the shaft and keep it in place while turning.

4.6. ***Weatherproofing***

4.6.1. *Material selection*

We chose rigid PVC sheets for *Nova's* body panels for their excellent weather resistance, UV, moisture, and chemical resistance. They also provide durability and stability, essential for enduring harsh weather without structural compromise. Aluminum Components were selected for being lightweight, durable, high strength, stable and corrosion resistant to ensure durability in outdoor settings.

4.6.2. *Secured Fasteners*

Loctite 243 is a threadlocker adhesive designed to secure threaded fasteners like nuts and bolts. It creates a strong bond that prevents loosening from vibration and protects against corrosion caused by moisture, temperature changes, and other environmental factors.

5. DESCRIPTION OF ELECTRONICS AND POWER DESIGN

5.1. ***Overview***

Nova's electronics design emphasizes efficient utilization and output of onboard components while prioritizing safety against external disturbances. Key features include signal processing via Exponential Moving Average, failure mode exception handling (over current, modulation limitation), and improved sensors and connectors.

5.2. ***Significant power and electronic components***

The electronics stack includes control and sensor suites. Maxon EC-60 BLDCs provide primary propulsion, powered by the onboard PDB. The PDB houses buck converters for 12V and 5V connections, along with current and voltage sensors for analysis on the STM32 Nucleo-64. Integrated thermistors monitor thermal readings, safeguarding against thermal agitation.

The MILE encoders provide encoder feedback for odometry calculations. An onboard PCB interfaces multiple components to the STM32 Nucleo-64. The wireless E-stop NRF module is connected to a separate STM32F401 board via SPI, which wirelessly interfaces with the NRF on the remote.

5.3. *Power distribution system*

The power distribution comprises a 6S 30000 mAh Lithium-polymer battery for DC power. An off-the-shelf 300W inverter is also used for DC to AC conversion (12VDC to 220VAC). The total current draw is monitored via a Shunt Resistor and a Current Sense Amplifier that sends an analog signal to the ADC of the on-board STM32 Nucleo-64. The monitored current and voltage are displayed on an LCD display.

Conservative power consumption estimate indicates a nominal power consumption of around 600W and full-load power consumption of 1200W. This corresponds to a current draw of 25A and 50A at 24V respectively, giving a nominal runtime of around 72 minutes and a full load runtime of 36 minutes. The battery specifications have been summarized in Table 12.

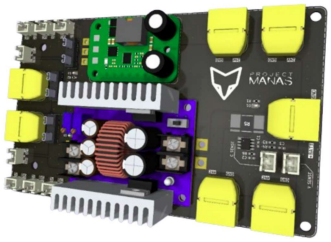

Figure 11. Power Distribution Board

Battery Type	Lithium Polymer
Voltage Rating	6S(22.2V to 25.2V)
Capacity	30000 mAh
Discharge Current Rating	25C
Nominal Runtime	72 minutes
Full-load Runtime	36 minutes

Figure 12. Battery specifications

Power to the motors is passed through a Solid State Relay (SSR), which can be deactivated anytime by the E-stop signal.

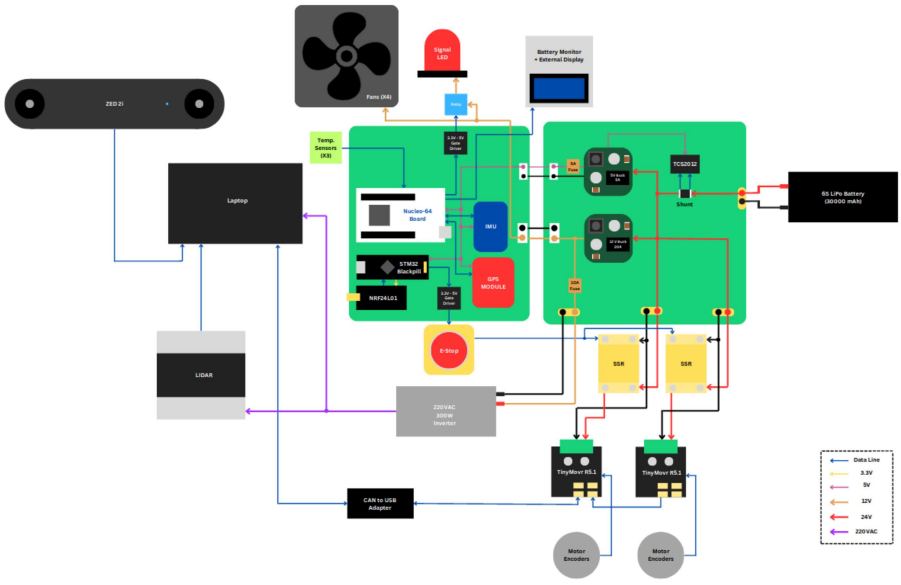

Figure 13. System Architecture

The power distribution board houses a 24-12V and a 24-5V Buck Converter. The 12V rail is primarily used to power a 300W (12VDC to 220VAC) Inverter, which in turn powers the Laptop and the LIDAR. A 10A automotive fuse has been used to protect the 12V rail from drawing too much power. Automotive fuses are known for being quick and easy to replace. The 5V rail is similarly protected by a 5A fuse.

5.4. *Electronics suite*

5.4.1. *Control Suite*

1. ***Asus TUF F15:*** The F15 houses all control systems featuring an Intel Core i7-11800H (11th Gen) and NVIDIA GeForce RTX 3050.
2. ***STM32F446RE MCU:*** The STM32 Nucleo-64 provides a flexible solution with optimum processing power to handle the accessory needs of the bot which include cooling systems, safety lights, peripherals E-stop package, and ROSserial.
3. ***TinyMovr R5.2:*** It is a compact and powerful brushless motor controller featuring an integrated absolute angle sensor for position, velocity, and torque modes via Field Oriented Control (FOC). It offers regenerative braking and flux braking for efficient power utilization.
4. ***STM32F401CCU6 MCU:*** The wireless e-stop implementation utilizes the STM32 blackpill with an NRF24L01p radio, communicating via the Serial Peripheral Interface protocol with an in air frequency of 2.4 GHz for swift and constant communication.

5.4.2. *Sensor Suite*

1. ***Ouster OS1 32-channel:*** The OusterOS1 32-channel Lidar sensor has a maximum range of 120 meters and a vertical field of view of 45°. The sensor offers a configurable horizontal resolution of up to 2048 points, enabling a detailed representation of the environment.
2. ***ZED2i Stereo Camera:*** The Zed2i Stereo Camera has a maximum depth range of 15 meters and a wide field of view of 110° horizontally and 70° vertically,capturing stereo images at a resolution of up to 1920x1080 pixels and generating dense point clouds.
3. ***BNO055 IMU:*** The BNO055 is a System in Package (SiP) solution that integrates a triaxial 14-bit accelerometer, an accurate closed-loop triaxial 16-bit gyroscope, a triaxial geomagnetic sensor and a 32-bit microcontroller running the BSX3.0 FusionLib software, used for sensor fusion and active suspension system.
4. ***NEO-M9N GPS module:*** The NEO-M9N GPS module is our GPS for *Nova.* With a horizontal accuracy of up to 0.5 meters and a velocity accuracy of 0.05 meters per second, it ensures precise positioning and navigation, whose compactness and power efficiency make it suitable for use.
5. ***MILE encoder:*** The MILE encoder uses an inductive angle measurement system to generate incremental quadrature output signals. It detects change and state of the Maxon motors to provide accurate velocity and position estimates for odometry and localisation.

5.4.3. *Motor Control*

The drive train for *Nova* accounts for our past inability to complete the course on time and low RPM output. To counteract this, the motor control uses two Maxon EC-60 brushless DC motors having a, 2510 mNm stall torque, and a 200W rating, 100 RPM which translates into the linear velocity of the bot at 2 meters per second.

Two TinyMovr R5.2s manage the Maxons' motor control. Using ROS and Python, velocity setpoints are established for each motor to follow the planner's path, transmitted through the 'cmd_vel' topic. MILE Encoders deliver precise feedback for odometry, ensuring accuracy with minimal noise.

An Exponential Moving Average (EMA) is used to filter out electromagnetic noise in the velocity and position estimates.

5.5. ***Mechanical and Wireless ESTOP Systems***

Nova has an Emergency-Stop (E-Stop) system that immediately halts the drive system by cutting power to the motor controller through a Solid State Relay (SSR). This relay is controlled by the STM32F4 controller present on the E-Stop board. The E-Stop can be triggered in two ways:

- ***Mechanically:*** A button placed on *Nova* pressed on malfunction.
- ***Wirelessly:*** A wireless E-stop for remote stoppage of the bot communicating using an NRF24L01 radio pair upto 800 meters.

A gate driver was used to step the 3.3V signal up to 5V , in order to drive the Solid State Relay(SSR), while working below ideal temperatures.

6. DESCRIPTION OF SOFTWARE SYSTEM

6.1. *Overview*

Our modular software stack, built on ROS (Robotic Operating System), comprises independent packages handling specific functionalities. It utilizes the sensor suite inputs to perceive the environment and generate a map. Based on this map, goals are set, and an optimal path and trajectory are calculated.

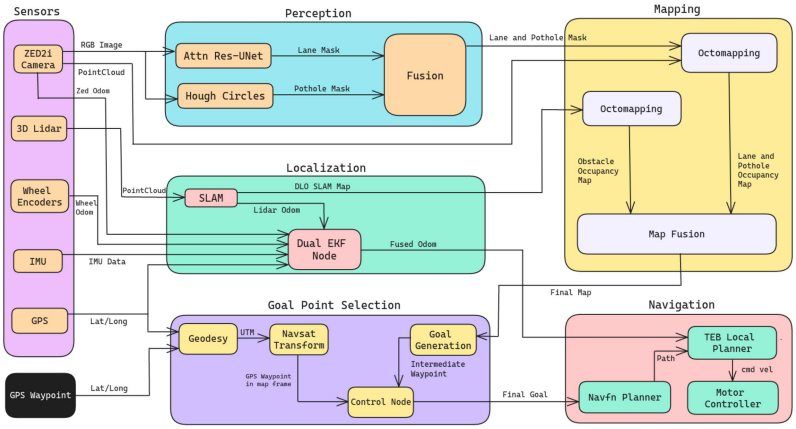

Figure 14. Software overview

6.2. *Perception*

6.2.1. *Lane and pothole detection*

Our lane detection module takes an RGB image from the Zed2i stereo camera and segments it to create a binary mask of the lines, using a U-Net-based model (see Section 3.3.1).

Leveraging the circular nature of the potholes, which appear as ellipses due to distortion of the ZED2i's perspective, we find contours in the image to which we fit ellipses of the pothole's dimensions.
We finally fuse the two masks received to obtain a combined binary mask containing lanes and potholes.

6.2.2. *Obstacle detection*

We use an OS-1 32 channel Ouster Lidar which constructs a 3D point cloud of the surrounding environment.

6.3. *Sensor Fusion*

6.3.1. *Direct Lidar Odometry*

We employ the Direct Lidar Odometry approach to perform simultaneous localization and mapping (SLAM) on the dense 3D point cloud obtained from the Lidar. This method allows us to estimate the robot's odometry reliably but at a slower pace. The resultant stable odometry is then integrated with the faster but less stable odometry obtained from other sources, creating a combined odometry that is both faster and more stable overall.

6.3.2. *Dual EKF*

For improved state estimation, we found two separate EKF filters can be used in tandem for a more robust state estimation.

The first is the local EKF node which integrates data from high frequency inputs including wheel encoders, visual odom and IMU. This node maintains a continuous estimate of the robot's pose and velocity. The second one is the global EKF node, which fuses less frequent but globally accurate data such as the GPS coordinates. We also include the DLO ODOM from the LIDAR in this EKF node, because we found it to be globally accurate, given its long range. This node is used to correct for the accumulated drift in the readings from the first node.

6.3.3. *Localization and Sensor Fusion*

	GPS	**LIDAR**	**IMU Sensor**		**Wheel encoders**	**Stereo Camera**
Datapoints	x, y	x, y, yaw, v_x, v_y, v_{yaw}	yaw	v_x, v_y, a_x, a_y	x, y, v_x, v_y	x, y, yaw, v_x, v_y, v_{yaw}
Type of data	lat, lon	odometry	imu	imu	odometry	odometry
Scope	global	global	global	local	local	local
Covariance	Positional Covariance Matrix calculated based on the specifications of our GPS module	Two 6x6 covariance matrices for Pose and Twist published DLO Odom	Three separate 3x3 Covariance matrices for orientation, linear and angular velocities respectively, were derived from the specifications of our IMU		6x6 covariance matrices for Pose and Twist calculated from the results of loop closure tests	6x6 covariance matrices for Pose and Twist published by Zed using visual odom

Table 2. Dual EFK configuration

6.3.4. *Localization on a Pre-saved map*

It is possible to localize on pre-saved maps using Adaptive Monte Carlo Localization (AMCL). This frees up computational requirements and allows for a more confident navigation at higher speeds, with necessary safety mechanisms in place.

6.4. ***Mapping***

We use OctoMapping to map a point cloud onto a 3D occupancy grid. Using an Octree data structure, which breaks down space into smaller sections. Each parent node recursively divides into eight smaller nodes, until we reach the smallest units, which store the probability that each small volume, or voxel, is occupied. To save memory, we remove nodes whose occupancy probabilities exceed a certain threshold. This keeps our mapping process accurate and resource-efficient.

6.4.1. *Adding lanes and potholes to map*

After detecting lanes and potholes, we create a mask to map pixels to points in the Zed point cloud, resulting in a masked point cloud. We then use OctoMapping to convert this masked point cloud into an Octree, which stores spatial occupancy information. Finally, we project the 3D occupancy grid onto a 2D occupancy map.

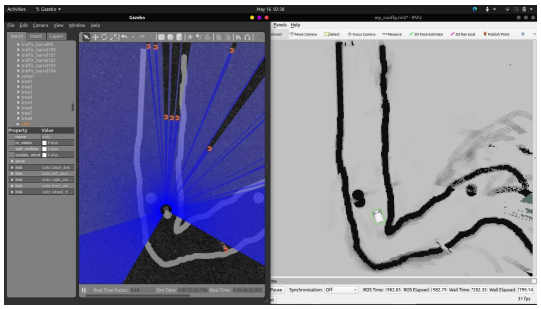

Figure 15. Lanes and potholes mapping

6.4.2. *Adding obstacles to map*

To address obstacles, we use the stitched point-cloud map from DLO SLAM. We filter out ground points by setting a height threshold, keeping only the relevant points for obstacle detection. This filtered point cloud is then mapped out,in the same way lanes and potholes are added to the map.

By combining the lane and the obstacle occupancy maps, we create a final fused map. In no man's land, while lanes on the ramp are accurately mapped, the ramp itself is not marked as an obstacle. This is because the height threshold is set to accommodate the ramp's maximum height.

6.5. *Goal Selection*

We employ two distinct modes for goal calculation, dependent upon the bot's location either within the lanes region or in no man's land. The selection between these modes occurs upon reaching the first GPS waypoint and the final GPS waypoint.

6.5.1. *Lane following*

This mode is enabled in the regions with lanes, before and after the no man's land. The camera's field of view is obstructed by obstacles, which leave gaps in the mapping of lanes. This could potentially lead the bot to navigate away from the lanes. To mitigate these issues, we implement our custom deep learning framework (refer to Section 3.3.2), involving the generation of intermediate goal points. This approach enables the vehicle to navigate effectively within the lanes.

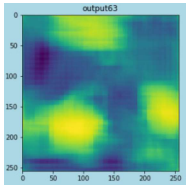

Figure 16. Raw probability output

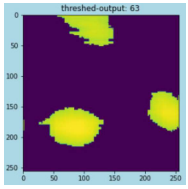

Figure 17. Filtered probability

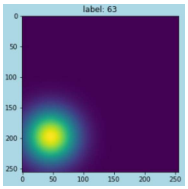

Figure 18. Gaussian label

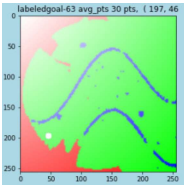

Figure 19. Labelled goal

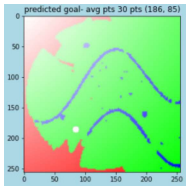

Figure 20. Goal prediction

6.5.2. *GPS following*

On entering the no man's land, the control node transitions to use the GPS waypoints to navigate. The latitude and longitude of GPS waypoints are transformed to the Universal Transverse Mercator (UTM) frame using the geodesy ROS package. Subsequently, these UTM coordinates are transformed into the map frame with the help of Navsat transforms. Finally, the calculated map coordinate is published as the designated goal, facilitating precise navigation within the robot's environment.

6.6. *Path and Motion planning*

6.6.1. *Path planning*

Nova employs the NavFN planner to find the least-cost path from the bot's current position to the next goal point on the map. This is done by running an algorithm similar to Djikstra's algorithm on a custom generated costmap. The planner continuously processes input data and updates the path at a frequency of 10Hz. The resulting path not only avoids obstacles but also maintains a significant distance from them, contributing to enhanced navigation safety.

6.6.2. *Motion planning*

We employ the Timed Elastic Band (TEB) algorithm to achieve smooth and efficient motion planning. TEB generates a locally optimal trajectory and velocity commands that don't violate the physical constraints of our robot. Extensive tuning of parameters associated with relevant heuristics was done to optimize TEB for our use case.

6.7. *Additional creative concepts*

- ***Deep Reinforcement Learning for Navigation:*** During research, we used end-to-end Policy Gradient RL for navigation, mapping velocities to state with Lidar range and goal info, successfully training a bot to navigate.
- ***Semantic Segmentation of 3D Point Clouds:*** We utilized the PointNet model architecture for semantic segmentation on 3D point clouds from Lidar to identify points belonging to dynamic obstacles like people and cars and remove them.
- ***Potential field based goal calculation:*** We explored a potential field based algorithm for goal calculation during lane following. Gaussian distributions of energy values were assigned to obstacles, lanes, and the bot, so as to guide the goal to reduce potential energy based on the gradients.

7. CYBERSERCURITY ANALYSIS USING RMF

7.1. *NIST Risk Management Framework(RMF)*

The NIST RMF is a 7-step procedure that can be applied by an organization flexibly to cost effectively apply and continuously monitor their security policies. The 7 steps are:

1. ***Prepare:*** This step involves preparing the organization for RMF by assigning roles and devising a strategy.
2. ***Categorize:*** The system undergoes categorization based on controls, and impact levels are determined by evaluating availability, confidentiality, and integrity, followed by a final impact level review.
3. ***Select:*** Based on the impact level and defined risk tolerance, we choose suitable controls and devise a strategy for implementation and monitoring.
4. ***Implement:*** Implement the identified security controls within the information system.
5. ***Assess:*** A third party assesses the controls to check if they meet the requirements.
6. ***Authorize:*** A senior official, often called the Authorizing Official(AO), evaluates the identified risks from the security control assessment to determine organizational acceptability and grants final authorization for the system or common controls.
7. ***Monitor:*** The system is continuously monitored to look for security risks.

7.2. *Model the threats and analyze their impact*

System	**Threat Description**	**Confedential**	**Availability**	**Integrity**	**Overall**
GPS Waypoints	If the GPS is altered the bot may go in wrong direction	low	moderate	high	high
Odom	If the odometry is altered bot may not know its relative position	low	high	high	high
ROS Commands	If velocity commands are intercepted the bot can be controlled	low	high	high	high
Version control credentials	Getting access to version control credentials can be used to change bot code	high	low	high	high
Wireless E-stop command	Rival team may send a wireless estop command to sabotage the run	high	moderate	low	moderate
Bag files	Bag files are only used for debugging, unlikely to be a threat	low	low	low	low
Onboard Computer	The onboard computer acts like the central node which can be used to control everything	high	high	high	high
Power Distribution Board	The main board responsible for relaying the commands the parts	high	high	high	high
Printed Circuit Board	The board which receives sensor board if modified can be used to send wrong sensor data	high	high	high	high

7.3. *Cyber controls and description of their implementations*

Goal	**Implementation**	**Efficacy**
AC-1 (Access Control)		
Protecting code from being modified by a rival team	Gitlab is used to manage version control for better security and access control	High: Delegating version control to gitlab reduces risks the team has to take

AC-4 (Information Flow Enforcement)		
Protecting bot from receving commands from rival team	Only the onboard computer may be connected to the internet and local parts are part of a local network which is not accessible from the internet so only the parts on bot can communicate with each other	High: Stops most wireless attacks targeted at onboard parts
Receiving wireless ESTOP signal from rival team	The ESTOP signal will be encrypted using RSA with a timestamp to stop man-in-middle or someone else sending an estop command	Medium: Prevents "man in the middle" attacks
AC-10 (Concurrent Session Control)		
Stopping rival team from connecting to central node	The central node only allows a a set number of devices to connect to it	High: Stops rival team from connecting to central node
SC-41 (Port and I/O Device Access)		
Stopping rival team from adding devices	Extra ports are blocked, and a custom circuit board is employed to exclusively permit essential devices.	High: Stops rival team from adding malicious devices
IA-5 (Identification and Authentication)		
To prevent unauthorized users from gaining access to the onboard computer	Each user has to enter a root password and a seperate login password, only accepted if there is a minimum password length, at least 1 digit, at-least 1 uppercase character	Moderate: Bears the brunt of the most rudimentary entry level attacks
IA-7 (Cryptographic Module Authentication)		
In the eventuality, that a password is lost, we need a safe and secure way to gain access to it	The assigned security supervisor has access to an encrypted Bitwarden account, which allows him to access credentials discreetly and securely in case a password is lost	High: BitWarden Premium has been tried and tested on the field and is a safe and secure password manager

8. ANALYSIS OF COMPLETE VEHICLE

8.1. *Lessons learned during construction and system integration*

Through multiple design iterations, we achieved optimal vibration damping and structural integrity for the T-Slots despite initial concerns about their weight. Initially, placing the payload above the motors for a centered gravity axis proved unstable during descents. Relocating the payload to the rear and lower position significantly enhanced the bot's stability by lowering the center of gravity.

8.2. *Top hardware failures that would prevent competition success and your mitigations*

Failure Points	**Causes**	**Solutions**
SSR failure at lower temperatures	Operation at suboptimal temperatures	Introduce a gate driver to drive the SSR
Motor overheating	Inconsistent tunes, physical hindrance to movement	Fuses on the PDB, current limiting in motor controllers
Suspension Failure	Misalignment of motorized and castor wheels during obstacle maneuvers leading up to the ramp	Employing linear actuator for dynamic control over terrains
Connection Integrity	High speed turns and vibrations	soldering, clip-on connectors, and fastening wires
Over Voltage	Excessive current draw	Installing fuses and setting upper bounds in the motor controllers
Motor Shaft Breakage	Excessive tipping, falls, or collisions	Double bearings for optimal stress distribution and secure fastening to minimize forward force

8.3. *Software Testing, Bug Tracking and Version Control Process*

Our software testing strategy includes three phases: tests for module functionality and component interoperability, and performance tests for system responsiveness. We utilized tools like ROS Logging through RQT Console, GDB Debugger, Roswtf, and rosbags for logging, debugging, and data replay, providing insights into robotic system behavior. For version control, we used Gitlab with modular ROS packages and managed dependencies through virtual environments,The following failure points were encountered:

Failure Points	**Causes**	**Solutions**
Sensor Data Inaccuracy	Accumulated error from continuous data streams and gps drift	Dual EKF (See Section 6.3.2)
Timestamp Mismatches	Delays in transforms being published, possibly due to computational overloads	Maintain a buffer along with interpolation techniques
No possible path to goal	Cluttered costmap possibly due to dynamic obstacles mapped incorrectly	Movebase Recovery Mode: performs a 360-degree scan to remap the local surroundings and clear any incorrectly mapped obstacles
Gaps in lanes	Due to restricted field of view and blind spots gaps in lanes while mapping arose causing the path to sometimes be planned through these gaps	Goal generation (See Section 3.3.2) generates intermediate waypoints
Critical system failures	System Crash or malfunction	Watchdog timers that trigger recovery upon critical failure

8.4. *SIL Virtual Environment Simulation Testing Process*

Figure 21. Screenshot of Gazebo simulation

We utilize Gazebo as our simulation platform, replicating real-world physics while constructing an environment that mirrors the competition course. This environment encompasses asphalt ground, obstacles, lane markings, and a ramp. Within this Gazebo world, we simulate an accurate model of our robot, complete with all its sensors, reflecting the specifications of our actual robot including mass, dimensions, inertia, and sensor positions. To ensure comprehensive testing, the lanes and obstacles are randomized each time the simulation is loaded. This is achieved by employing control points being randomly chosen to define the Bezier curves for the lanes, while obstacles and ramp are dynamically spawned within the environment.

8.5. *Physical testing to date*

- ***Mechanical:*** Mechanical testing of *Nova* focused on chassis stability and integrity. An issue with the Zed cam's low height led to extending the sensor pole, but this caused excessive vibrations affecting mapping accuracy. To resolve this, a carbon fiber rod was added to decrease deflection and increase structural rigidity.
- ***Electronics:*** Some issues we predicted were an offset in the control variable and encoder noise, leading to increased instability. Analysis of odometry and cmd_vel data indicated lag and clamping due to tuning inconsistencies. Prototyping revealed inconsistent tunes which were analysed using an oscilloscope and rectified using Ziegler-Nichols.
- ***Software:*** For testing, we created a course similar to the competition environment using white corrugated fiberboards for lanes, obstacles, and a ramp. The two main differences observed were sensor noise issues like unstable ZED points and GPS drift which affected mapping and variations in lighting on the lanes such as shadows causing lane detection issues.

9. INITIAL PERFORMANCE ASSESSMENT

The software was rigorously tested on the aforementioned course with an average run time of 5 minutes per attempt. We successfully tested complex and dynamic obstacles. Our bot was also able to climb a ramp with 15% inclination successfully.

Total Runs	189
Average Run Time	4 min 38 seconds
Average Speed	0.5 m/s
Max Speed	1 m/s
Max Acceleration	$1.5m/s^2$
Battery Life	36 mins
Waypoint accuracy	0.1 meters tolerance of the goal point
Distance at which obstacles is detected	120 meter Lidar range for obstacles, 20 meter Zed range for lanes
Reaction times	10 Hz path update rate, Reactive TEB motion