Intelligent Ground Vehicle Competition 2023

Indian Institute of Technology Madras
Team Abhiyaan
Vikram

Design Report

I hereby certify that the development of the vehicle, Vikram, 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 Members

Electronics

Adil Mohammed K
Arun Krishna AMS
Dhaksin Prabu K
Kesava Aruna Prakash R L
M.R.Kaushik
Niranjan A. Kartha
Rahul
Soumya Ranjan Behera
Vamsi Krishna Chilakamarri

Mechanical

Advait Abhijeet Kadam
Ananya Dhanvantri Guruprasad
Aneesh Bhandari
Arvind P
Hiran Neelakantan
Jayesh soni
Krishma Mehta
Mugdha Meda
Reitesh KV Raman
Saumya Mathur
Shobhith Vadlamudi
Sudharsan Raja

Software

Aahana Hegde
Aayush Agrawal
Amar Nath Singh
Keerthi Vasan M
Lalit Jayanti
Saiharan Rajakumar
Sukriti Shukla
Suneet Swamy
Suraj Rathi
U.K.Arvindan

*Names are hyperlinks for emails.

Contents

1. CONDUCT OF DESIGN PROCESS, TEAM IDENTIFICATION AND TEAM ORGANIZATION

1.1. *Introduction*

Team Abhiyaan is a group of 30 interdisciplinary students enrolled in undergraduate engineering programs at the Indian Institute of Technology (IIT) Madras. *Team Abhiyaan* is fueled by a common passion for autonomy, building driverless vehicles in India. The team is mentored by Prof. Sathyan Subbiah (Department of Mechanical Engineering) and supported by the Centre for Innovation (CFI), a student-run facility at IIT Madras. We are proud to present a robust and versatile *Vikram*, an upgrade on all fronts to the previous year's vehicle, *Vajra*. The design report is a comprehensive documentation of the thought and process involved in building *Vikram*.

1.2. *Organization*

The team is organized into 4 modules, namely Software, Electrical, Mechanical, and Business & Design. Each module is supervised by module leads who are accountable for module-specific tasks and functioning. The team heads ensure smooth handling of the team and make final decisions based on inputs from all team members. To facilitate effective knowledge transfer, the team encourages strong inter-module interaction. Moreover, the team embraces a flexible hierarchy, allowing members to contribute in multiple modules, promoting a free and expressive environment where members can learn, question, and implement their own ideas.

Experienced senior members play a crucial role in passing down their technical knowledge to new recruits allowing the team to continually build upon and capitalize on their experience. Frequent team meetings are conducted to ensure that all members are on the same page. Task management for the team is aided by productivity applications such as *Notion* with determined timelines and checklists, tracking work progress effectively.

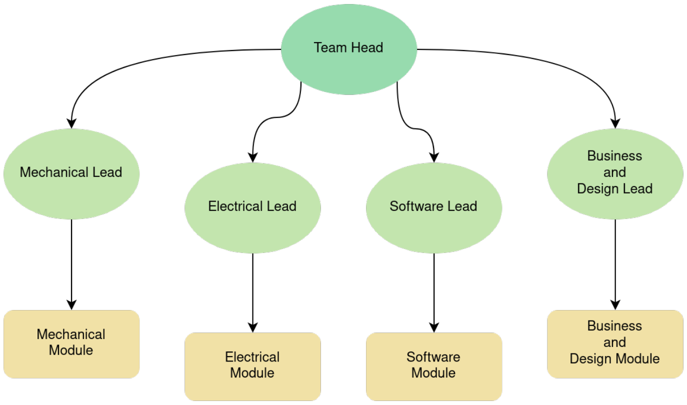


Figure 1: Team Structure

1.3. *Design assumptions and design process*

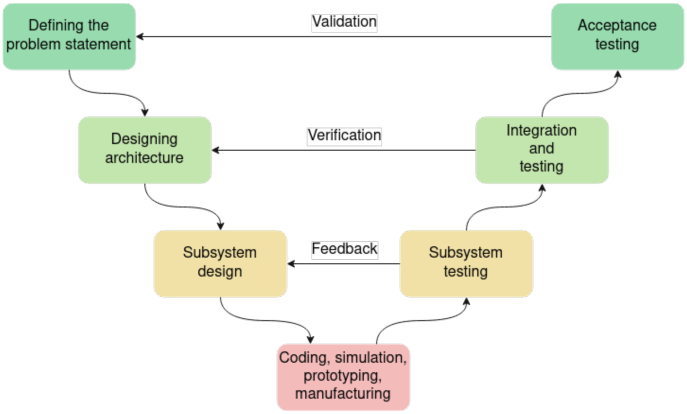


Figure 2: V - model

The design process begins with preliminary assumptions about the intended environment of the vehicle and the requirements to be fulfilled in accordance with the IGVC rules and regulations. The various criteria regarding physical dimensions, performance metrics, core functionalities and operating design domain (ODD) are decided based on these rules. The next step is to define the problem statement and devise solutions.

After a rigorous ideation phase, the overall architecture is devised and divided into multiple subsystems. Simulations, numerical and analytical methods are utilized to validate the subsystem requirements. The components are then chosen based on the requirements, resource constraints and compliance with industry standards. Subsystem prototypes are built and tested individually before integrating and finally testing them. Appropriate changes and tweaks are iteratively performed at each stage of design and development to optimize the performance. The results of the iterations are documented for inference and future reference.

2. EFFECTIVE INNOVATIONS IN OUR VEHICLE DESIGN

2.1. *Innovative technology applied to your vehicle*

2.1.1. *Mechanical*

1. **Independent Suspension:** A responsive, in-house produced independent suspension system for the wheels (including castor). See 3.3
2. **Cooling Systems:** A custom-built cooling system was designed to regulate the temperature within the component housing. The stock thermal system of the NUC did not provide sufficient heat dissipation and caused a drop in efficiency. A separate external cooling system was made to encase the NUC. 3.4 and 7
3. **Electronic Actuation for Hatch and Payload Access:** The actuated hatch provides easy access to the inner components and can potentially open and vent if the temperature inside reaches a pre-determined threshold. A separately actuated bonnet allows for convenient reach to the payload. See 3.5

2.1.2. *Electrical*

1. **BMS:** A Battery Monitoring System developed in-house to monitor battery health, acquire data for analysis and optimization, and turn off supply on detection of critical states . See 4.2
2. **CAN Bus:** A CAN bus introduced for secure, error-corrected inter-microcontroller communication using the CAN 2.0B framework. The design is highly modular and flexible to upgrades.
3. **SPARC Controller:** A Data-Driven Self-Evolving Parameter-Free Rule-Based Adaptive controller to eliminate the need to tune variables and achieve precise control despite non-linearities. See 4.4

2.1.3. *Software*

1. **Heartbeat:** Developed a program built on ROS to monitor the health of sensors, connectivity, and latency. This program issues warnings and takes safety actions if required. See 6.3
2. **Image Segmentation on a Custom Dataset:** Image Segmentation was used to detect the potholes and lanes from the image. Deeplabv3 is a Transfer Learning based model that was trained on a **custom dataset**. This was created using openCV. See 8
3. **Wheel Encoder Error Model:** Developed a program to estimate the covariance in the position of the robot based on the instantaneous displacement of the wheels. See 5.3.3
4. **Spatio Temporal Voxel Layers:** Utilized novel spatiotemporal voxel layers to keep track of obstacles in a computationally efficient manner. See 5.3.2
5. **Ramp Detection:** To detect the angle of elevation during ramp traversal a BNO05 9 DOF gyro was added to the PCB. The gyro sends data to the computation unit through a serial converter. This data is then taken into account for localization. See 5.2.1

3. DESCRIPTION OF MECHANICAL DESIGN

3.1. *Overview*

The key objective of the mechanical module of Team Abhiyaan is to build a robust vehicle that meets the software and electrical modules' requirements. The current vehicle is an upgrade on all fronts to our previous model, boasting a newly introduced suspension, cooling system, and electronic actuation for easy access, amongst other innovations. While designing, the vehicle's structural rigidity, weight optimization and distribution, modularity, aesthetics, and compliance with the rules were considered. The significant contrast between the two has been elaborated in the further sections.

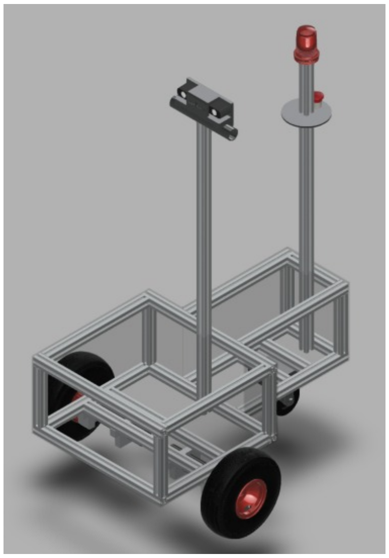


Figure 3: Vajra (2022)

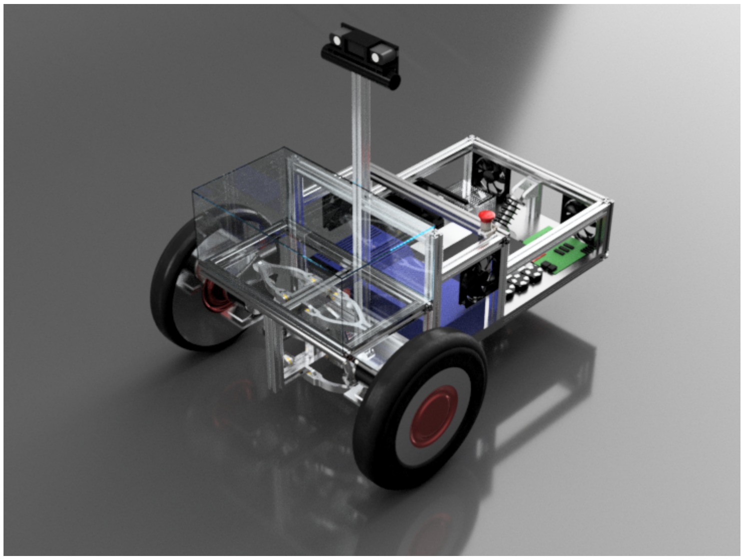


Figure 4: Vikram (2023)

3.2. ***Drive System Analysis***

Vikram employs a differential drive configuration, comprising a single castor wheel in the rear, and two drive wheels at the front, each powered by a separate 350W brushed DC motor, coupled to an encoder.

Motor and Encoder Assembly: The motors and encoders employed in *Vikram* are more modular and affordable than the previous year's. In contrast to the axial, direct coupling of the motor and encoder in the previous vehicles, the drive system of *Vikram* involves coupling via a Herringbone gear train. Herringbone gears enable smoother meshing than spur gears and negate axial forces present in helical gears.

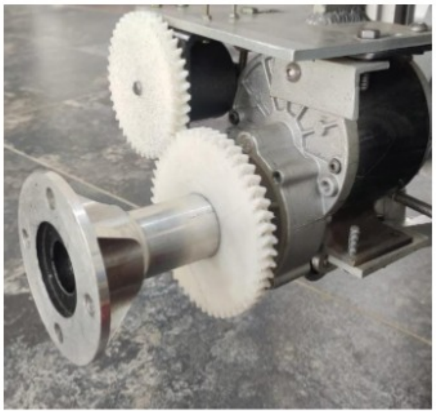


Figure 4a: Drive Assembly

Figure 4b: Drive wheel suspension

Figure 4c: Castor wheel suspension

3.3. ***Suspension and Articulation***

Camera feed received on our previous un-sprung vehicles was found to be unstable due to road modulations. In addition to this, insufficient articulation lead to erratic encoder readings. Hence an independent suspension system was developed to improve the vehicle dynamics.

1. **Drive Wheel Suspension:** A double wishbone setup is employed for the two front drive wheels. This configuration enables the vehicle to achieve a stable and balanced ride, isolating the vehicle from road vibrations as well as the motion of one wheel from the other.

2. **Castor Wheel Suspension:** The castor wheel was a major source of high amplitude shocks due to a single point of contact with the tarmac. The need to develop a suspension system for the castor wheel arose when off the shelf spring-loaded castors failed to provide the required damping and wheel travel. A trailing arm suspension setup enables a steady rear end of the vehicle, thus greatly improving the dynamic stability.

After rigorous testing, it was found that having a suspension system reduced acceleration deviations from 2.59g to 1.17g in repeated rumbler strip tests. Also, various components required for the suspension, such as wishbones and trailing arms, have been developed with a sufficient safety factor, close to 2. See 8.1

3.4. *Cooling Systems*

During testing, over-heating was observed, which could lead to permanent damage of electronic components. To combat the issue, a thermal cooling system was developed for the enclosure, consisting of high flow rate fans and heat dissipating fins. Airflow inside the compartment was modelled using CFD analysis tools. Components were placed accordingly, optimizing airflow requirements for each component and weight distribution of the vehicle.

The NUC produced large amounts of heat. To dissipate this, a custom cooling unit was made with radial flow fans to lower temperatures. The ventilation system for the main enclosure dissipates close to 200W of heat during normal operation and the external cooling system for NUC helps lower the peak temperature by 4°C. See 7

3.5. *Improved Accessibility with Electrical Actuation*

Electronically Actuated Hatch for Payload and Component Bay: The previous vehicle posed a considerable challenge to open up due to its firmly bolted acrylic enclosure. A hatch for the inner components and a separate compartment to carry the payload was introduced this year, both capable of automated access by a linear actuator.

3.6. *Chassis Design*

1. **Material Selection and Modularity:** The chassis needs to be strong to support the entire load, yet light, for better maneuverability. Hence, the frame is made with aluminum, using highly modular standard extrusion bars. They enable efficient assembly and disassembly as well as compact packaging to aid our logistics.
2. **Space Efficiency and Weight Distribution:** The vehicle was made more compact this year with space efficient organization of components. Proper weight distribution is crucial for the running stability of a vehicle. The payload, the heaviest component, is placed right above the suspension for better shock absorption and for more traction at the drive wheels.
3. **In house Manufacturing of Composites for the Vehicle Enclosure:** It was decided to use Fibre Reinforced Plastic(FRP) sheets for the body panels and base plates owing to their low weight and high strength, a clear upgrade from acrylic and metal sheets. They were fabricated with glass fiber mats and polyester resin by the team members themselves.
4. **Adjustable Positioning for Stereo Camera:** A simple yet innovative stereo camera mount that allows for variation in pitch and elevation(2 DOF) from ground if required.
5. **Weatherproofing:** Panel gaps and edges have been fitted with rubber seals and foam tape to prevent ingress of water. The actuated hatch applies a constant force on the seals when closed, ensuring a watertight enclosure. To prevent entry via the cooling fans, the body is made tapered towards the vents, causing water to flow off.

4. DESCRIPTION OF ELECTRONIC AND POWER DESIGN

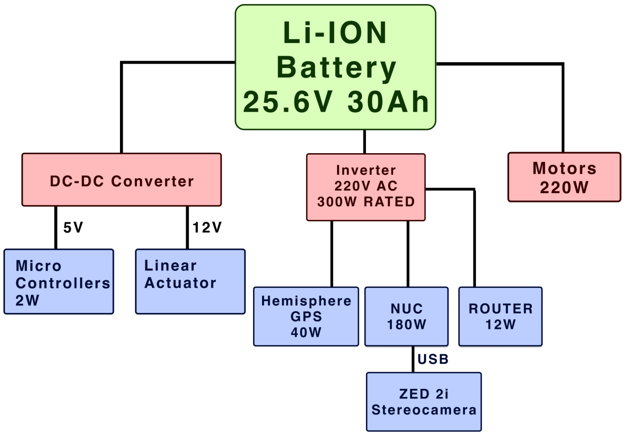
4.1. *Overview*

The Electrical Design this year was focused on mitigating past issues with more resilience to data and hardware errors, as well as making the whole system more modular and adaptable to future upgrades. Some prominent features are:

1. Active Battery Monitoring System .
2. CAN Bus for Reliable Communication between Microcontrollers and Flexibility for the Addition of more ECUs for Distributed work or Future Upgrades.
3. Parameter-Free Control System with Point Cloud analysis software.

4.2. *Power Distribution Systems*

The entire system is powered by a single 25.6V Lithium Ion Battery. Off-the-shelf DC-DC converters and DC-to-AC Inverter are used to power all components.



(a) Power Distribution Graph

1. **Capacity:** 25.6 V 30 Ah: 760 Wh
2. **Max Power Consumption:** 460 W
3. **Minimum Runtime on Full Charge:** 1.25 hours
4. **Recharge Time:** 2 hours
5. **Minimum Efficiency:** 75%

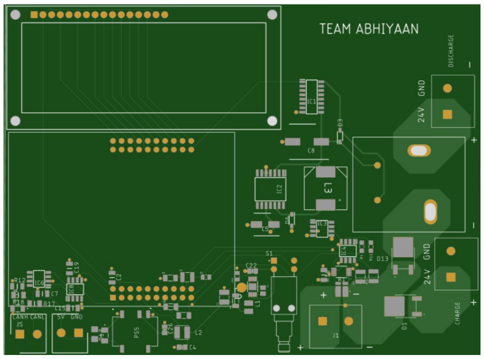
Wires are selected based on AWG standards to meet the necessary current requirements. In order to prevent the occurrence of accidental improper polarity connections, we employ the use of XT-60 connectors. These connectors are specifically designed with a polarized male and female interface, ensuring that the positive and negative terminals are correctly aligned during the connection process.

Battery Monitoring System:

A smart BMS has been developed as a separate independent node on the system to provide comprehensive data on battery life. The BMS includes undervoltage, overvoltage, short circuit, and reverse polarity protection features. LTC4151 current sensor measures the current drawn from the battery to find the State-Of-Charge of the battery using the Coulomb Counting method. The processing of all the data happens on a Texas Instruments TIVA microcontroller, and the battery health is displayed on an LCD display. To ensure data preservation during power loss, the microcontroller stores the information in its non-volatile memory.

The PCB is developed with several safety and convenience features:

1. Isolated communication between the current sensor and TIVA.
2. Isolated CAN Bus access to communicate with external nodes
3. Relays to stop the system's operation if a critical battery state(Overvoltage, under voltage, or short circuit) is detected.
4. Separate battery charging and discharging ports with reverse polarity protection on both.



(a) Battery Monitoring System

4.3. ***Motor Control***

Brushed DC motors that drive the vehicle are controlled by Roboteq SDC2160 2-channel High performance motor controller with closed-loop feedback using incremental encoders. The velocity feedback is obtained by a Texas Instruments TIVA microcontroller and sent to the NUC where the control algorithm is run. The controller output throttle is sent by the NUC to ROBOTEQ motor controller.

Roboteq motorcontroller is configured to detect and limit the max current drawn for each channel. The controller is powered off if an overvoltage, undervoltage and short-circuit protection is detected.

Quadrature incremental encoders are used for the velocity feedback of both wheels. The chosen encoders have 12V totem-pole output which provide bidirectional current flow and fast switching capabilities compared to last year's open-collector configuration(5V pulled-up). The larger voltage difference between the signal and the noise sources, offer high signal-to-noise ratio in our current encoders - making the signal less susceptible to interference. Digital signal processing is employed to implement a low-pass filter on the encoder data before the feedback is sent to the control algorithm.

4.4. ***Control Systems***

A classic PID controller faces limitations due to its lack of adaptability to load variation, non-linearity in the system, and high noise disturbances. The PID control gains must be appropriately tuned to ensure the functionality of the system. To avoid retuning of the parameters, SPARC - Self-Evolving Parameter-Free Rule-Based Controller, a Real-Time Adaptive controller is designed that can learn without requiring any static tuning.

Fuzzy rules are used to model relationships between inputs and outputs in a system. In SPARC, the fuzzy rules are based on the data density, which refers to how closely data points are distributed in the input space. The focal points derived from the data-density act as membership functions. These membership functions define the degree to which an input belongs to a specific fuzzy set.

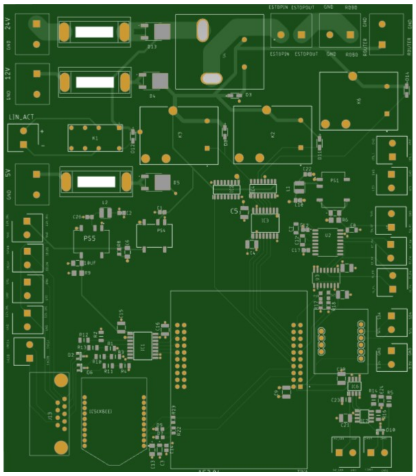
Focal points represent certain properties or characteristics of the current data. The new data is compared with the membership functions (focal points) & the degree of similarity or resemblance between the new data and the membership functions is determined. Focal points that share more properties with the current data contribute more to the output. Visualization tools are developed to understand how the focal points are updated over the time to get further insights about the system

The controller can function autonomously in a fully unsupervised manner with a parameter-free control structure. The controller's performance was similar to PID in ideal linear environments, but the difference was significant on increasing non-linear factors. Simulation and testing results are provided in the section: 4

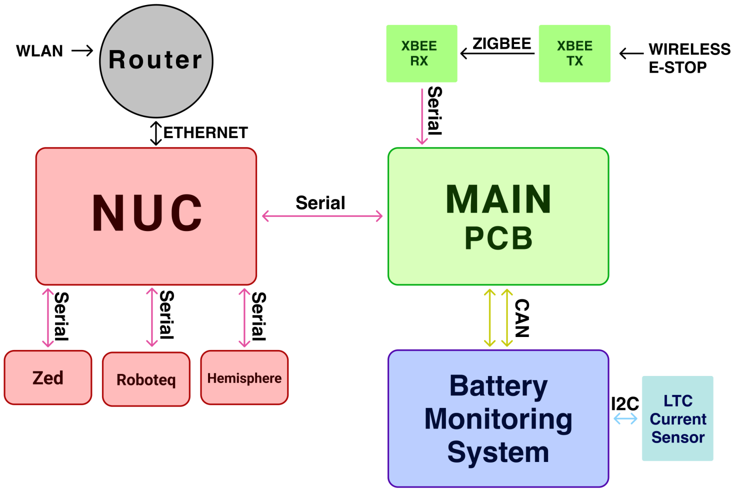
4.5. ***Main PCB***

The main PCB houses the entirety of electronic components: power supply control, microcontroller, encoder peripheral interface, safety and protection systems, and wireless communication modules. The PCB is been designed in Autodesk Eagle following the PCB layout guidelines to minimize noise and electromagnetic interference. Following features were added:

1. Tiva microcontroller - Encoder peripheral interface with power & signal isolation
2. Control of power supply to the components(motorcontroller, status lights) through electromechanical relays
3. Safety fuse & reverse polarity protection in power supply lines.
4. Wireless E-Stop receiver module using XBee.
5. Isolated CAN communication port to connect with external CAN nodes.



(a) Main PCB



(b) Communication Systems

4.6. *Safety & Protection Mechanisms*

1. **Power and Signal Isolation:** Power and signal isolation between encoder-TIVA interface, NUC-TIVA interface, NUC-Roboteq, TIVA-Current Sensor interface & CAN communication interface is designed to provide electrical isolation and to prevent power surges in one system affecting the other
2. **Emergency Stops:** A Bright Red Emergency Stop placed in an accessible location disconnects power to the motor controller. Wireless Emergency Stop(with range over 200m) is implemented using a Transceiver Remote built over an XBEE-Pro utilizing ZIGBEE protocol. The receiver XBEE-Pro is placed on the Main PCB on Vikram. A wireless heartbeat is implemented for the competition to ensure that Vikram is always in range of the E-Stop. Loss of communication stops the vehicle.
3. **Status Lights:** Status lights that indicate whether the vehicle is in autonomous or tele-op mode.
4. **Battery Protection:** BMS that provides undervoltage, overvoltage, short-circuit, low SoC protection measures. Fuses and Reverse polarity protection diodes are also added.

5. SOFTWARE STRATEGY AND MAPPING TECHNIQUES

5.1. *Introduction*

The software stack has been designed to be modular, reusable, and extensible. It is divided into independently functioning layers. Each layer is composed of one or more components. This allows us to build, test, and improve each layer independently.

There are three primary layers:

- **Perception Layer** detects (or 'perceives') the lanes, obstacles, and other features from the depth camera data.
- **Planner Layer** takes the costmap and waypoint goals. It uses the vehicle's current state and physical limits to generate a trajectory to follow.
- **Velocity Controller** To traverse the generated path we give velocity setpoints to the electrical systems.

5.2. *Obstacle detection and avoidance*

5.2.1. *Sensors*

- **Zed 2i Stereo Camera:** is used to perceive the environment. It provides a 50Hz RGB image and depth image stream. This combination of data allows us to perceive lanes, obstacles, potholes, and ramps while generating their point clouds in real-time, these point clouds are then converted to costmaps.
- **Septentrio's Mosaic-go GNSS evaluation kit:** is used as the GPS solution on *Vikram*. This model provides a dual antennae setup, enabling Vikram to precisely geolocalte itself. The GNSS has a dual antennas that provides us with global heading.

- **Adafruit BNO05 9 DOF Absolute Orientation IMU:** is used to get the Absolute Orientation(Roll, Pitch, and Yaw) of the vehicle. This data helps us get the orientation of the vehicle w.r.t the initial calibaration position. Data from this sensor is used to calculate ramp elevation.

5.2.2. *Ground Layer Perception*

There are 3 main components on the Ground Layer: Lanes, Ramps, and Potholes. A heuristic OpenCV pipeline was built to detect them. The steps are shown in the figure below.

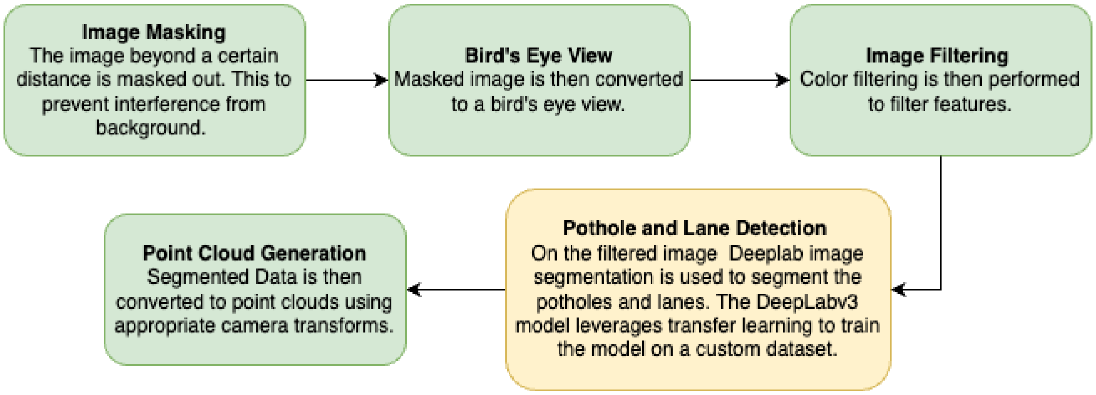


Image Masking
The image beyond a certain distance is masked out. This to prevent interference from background.

Bird's Eye View
Masked image is then converted to a bird's eye view.

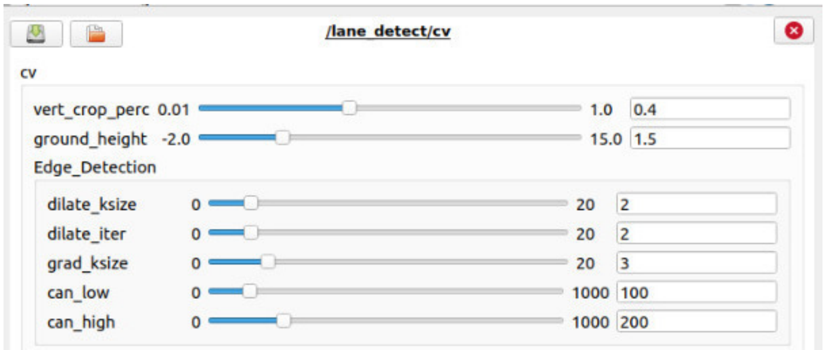
Image Filtering
Color filtering is then performed to filter features.

Pothole and Lane Detection
On the filtered image Deeplab image segmentation is used to segment the potholes and lanes. The DeepLabv3 model leverages transfer learning to train the model on a custom dataset.

Point Cloud Generation
Segmented Data is then converted to point clouds using appropriate camera transforms.

Figure 8: Ground Layer Perception

Dynamic Parameter Tuning: In IGVC 2022, we faced a large challenge due to variations in lighting conditions with the time of day and weather. This caused the stack to perform sub-optimally. The parameters could be tuned using immediate visual feedback.



/lane_detect/cv

cv

vert_crop_perc 0.01 … 1.0 0.4
ground_height -2.0 … 15.0 1.5

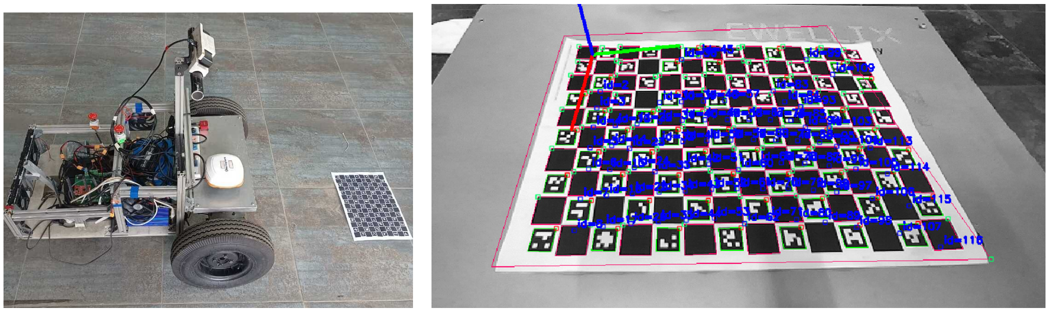
Edge_Detection

dilate_ksize 0 … 20 2
dilate_iter 0 … 20 2
grad_ksize 0 … 20 3
can_low 0 … 1000 100
can_high 0 … 1000 200

Figure 9: Dynamic Parameter Tuning

Camera Calibration: A camera calibration system is used to provide accurate data on the camera's parameters and position with respect to the robot's body. The ArUco markers can be used for calibration, which is versatile but may lack accuracy. To address this, a combination of ArUco markers' versatility with the precision of chessboard patterns was implemented, resulting in accurate calibration even with occlusions or partial views.

The ChArUco board is placed at a set distance from the base of the vehicle to get a fixed frame. This setup provides both the extrinsic and intrinsic parameters of the camera.



(a) Physical Setup

(b) Software Output

Figure 10: ChArUco Calibration

5.2.3. *Obstacle Detection*

On the raw 3D point cloud data, a coordinate filter and outlier filter are applied to remove irrelevant points. The remaining points are down-sampled onto the occupancy grid used by the path planner. These points are then fed to a Voxel Layer.

5.3. ***Software strategy and path planning***

Vikram uses a layered costmap architecture to keep track of the detected obstacles and plan the optimum path.

5.3.1. *Localization and Mapping*

Extended Kalman filtering is used to fuse sensor data for state estimation.

Pose estimate	**GPS**	**Encoders**	**Stereo camera**	**Gyroscope**
Global Pose estimate	✓	✓	✓	×
Local Pose estimate	×	✓	✓	✓

Covariance models were developed for the encoder-based position estimation. This is required for the optimal performance of the Kalman filter. The current covariance is estimated from the encoder readings in the last time-step and the previous covariance as seen in the following equation.

$$\Sigma_{p'} = \nabla_p f \cdot \Sigma_p \cdot \nabla_p f^T + \nabla_{\Delta_{rl}} f \cdot \Sigma_\Delta \cdot \nabla_{\Delta_{rl}} f^T$$

where, Σ_p is current covariance estimate, $\Sigma_{p'}$ is the updated covariance and $\nabla_p f$ and $\nabla_{\Delta_{rl}} f$ are the Jacobian matrices of the the current pose p and motion increments respectively .Further, for the motion increments of $\Delta s_r, \Delta s_l$ of the wheels the covariance matrix is

$$\Sigma_\Delta = \begin{bmatrix} k_r \left|\Delta s_r\right| & 0 \\ 0 & k_l \left|\Delta s_l\right| \end{bmatrix}$$

where k_r and k_l are experimentally determined constants representing non-deterministic parameters

The accuracy Vikram's localization was tested using a loop-closure test, where Vikram was driven over a closed path of $50m$ length twice and then measuring the difference in the estimate of starting and ending position for each loop. The errors for each loop were around $0.5m$ to $0.8m$. Similar tests were used to tune the nondeterministic parameters in the error model and the Kalman filter.

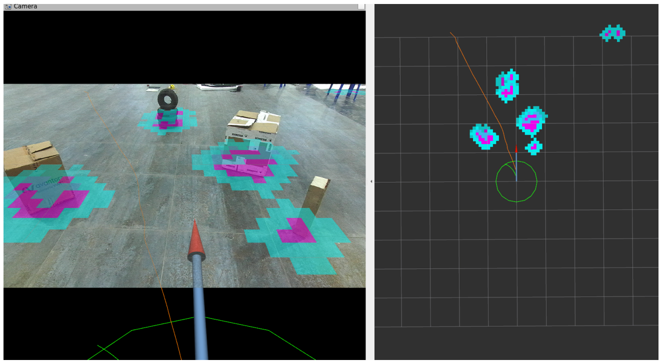


Figure 11: Costmaps and planning

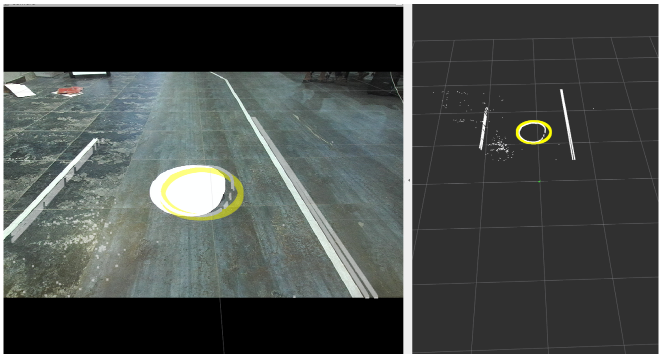


Figure 12: Potholes and lanes detection

5.3.2. *Costmaps*

A costmap is a map which where each cell is assigned a cost based on its occupancy, a higher cost is indicative of a higher probability of being an obstacle. The point cloud data obtained from the stereo-camera, the detected potholes, and lanes are processed in three layers.

1. **Spatio-Temporal Voxel Layer:** The novel Spatio-Temporal voxel layer, which makes use of efficient hierarchical data structures to keep track of detected obstacles (Spatial). Further, this layer utilizes the concept of voxel decay, wherein detected obstacles disappear from the map over time (Temporal), eliminating the need of the computationally expensive raytracing procedure to clear obstacles. Utilizing this layer over conventional layers has offered us significant improvement. On an experimental setup with a 10hz dense stereo RGBD camera and an 11th Gen i7 CPU, the process runs at 20-25% of a core updating the map at 9hz, compared to the 90-100% with conventional means updating the map at 4hz.
2. **Potholes and Lanes Layer:** Potholes and lane point clouds detected from the perception stack are integrated into the costmap.
3. **Inflation Layer:** Each obstacle is artificially inflated to enforce a safe clearance.

This approach of integrating all the obstacle information on a costmap eliminates the need of secondary logic or programs to control Vikram, allowing for efficient planning. Once this map is generated it is passed on to the path planners.

5.3.3. *Pathplanning*

Vikram plans its path by first generating a global plan, a high-level plan from the current position to the next waypoint using the A* algorithm on a global occupancy grid, then a suitable local plan is generated using the TEB planner (Timed Elastic Band) on a local occupancy grid.

The TEB planner is an online optimal local trajectory planner. It generates a local plan by optimizing the global plan for minimum trajectory execution time, separation from obstacles and compliance with the kinodynamic constraints. This is done by solving a sparse optimization problem. This planner was chosen above other planners owing to its compliance with non-holonomic models, in this case the differential drive model.

For efficient planning, the local plan is updated at a higher rate and an occupancy grid of higher resolution ($0.05m$) is used and control commands are issued at 25hz this allows for finer control on the vehicle, essential for obstacle avoidance. Whereas the global plan is updated at a lower rate of 6hz while using a low-resolution occupancy grid ($0.1m$), this gives the planner time to generate optimal global plans.

Once a plan is generated, it is converted to a set of velocity commands. Velocity commands are resolved into the required rpm set-points for the wheels using the differential drive model. These set points are passed to the SPARC controller to be executed.

5.4. *Map generation*

A 2D lidar was used to generate a map of a particular area. However, the mapping did not improve the performance of the system, and the extra computation required did not justify the output. Despite the potential benefits of mapping, such as providing a more accurate perception of the environment, the team considered the costs and benefits before implementing such a solution.

5.5. *Goal selection and path generation*

Vikram executes a mission autonomously once a set of waypoints are provided from the starting point to the goal, in the form of absolute position (such as GPS coordinates). Vikram seeks the waypoints in order while providing feedback to the mission controller with the current status. The mission control loop is illustrated in detail in Figure 13.

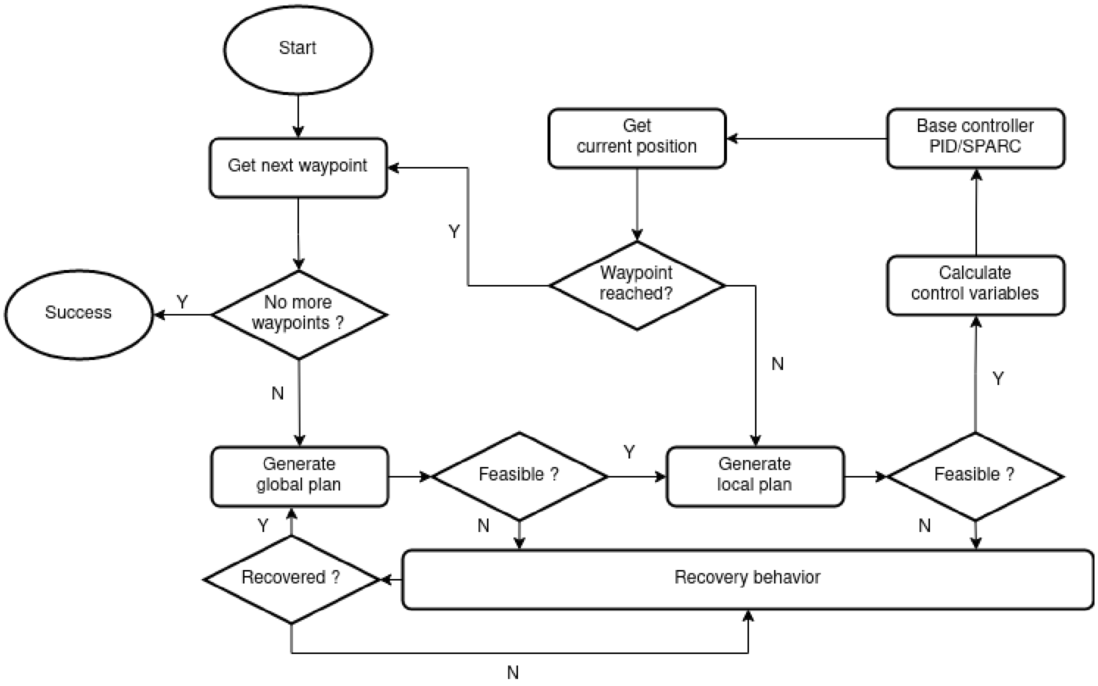


Figure 13: Mission control loop

5.6. *Computation Unit*

During the development of the previous prototype, named "Vajra", issues were encountered with distributed computing, which utilized three different devices for computing purposes, namely, Intel NUC, Raspberry Pi, and Nvidia Jetson. While distributed computing has its own advantages, it was observed that a significant improvement in performance was achieved when the computation was done on a single device. Specifically, the preferred device for this year's prototype was the Intel NUC Enthusiast with onboard Nvidia 2060 GPU.

An improvement in performance was attributed to the reduction in latency for data transfer between the GPU and CPU, as well as the elimination of a potential failure point, such as bandwidth, latency and network breach issues. The decision to utilize the a centralised computing for this year's prototype was made based on these factors.

6. DESCRIPTION OF FAILURE MODES, FAILURE POINTS, AND RESOLUTIONS

6.1. *Vehicle failure modes (software, mapping, etc) and resolutions*

Failure point	**Cause**	**Resolution**
Inaccuracy in Encoder readings	Bad road conditions can translate to inaccurate representation of localisation when encoder reading is converted to pose and velocity	Fuse Zed odometry with the encoder's odometry
Loss of camera data	This can happen due to various reasons like unexpected obstacles, bad weather conditions, etc.	Encoder data is present to ensure localisation continues, and the vehicle does not go out of control
Loss in a data stream	Unaccounted errors in code or sensor driver issues could lead to a certain node to stop publishing data	Heartbeat has been deployed to ensure appropriate measures are taken when a certain data stream is lost. See 6.3

6.2. *Vehicle failure points (electronic, electrical, mechanical, structural, etc) and resolutions*

6.2.1. *Mechanical*

Failure point	**Cause**	**Resolution**
Gear disengagement between motor and encoder gears	Axial thrust produced by helical gears	Switch from helical to herringbone gears
Wheel wobbling with respect to the motor	Loosening of press fit between the coupler and motor	Providing provision for fastening with lock nuts
Vibrations create unreliable camera sensor data	Vehicle moving on asphalt at reasonable speeds will face vibrations that make camera based algorithms less accurate	Build a robust suspension system

6.2.2. *Electrical*

Failure Point	**Cause**	**Resolution**
Power Surges damage connected components	Inrush currents and Electrostatic Discharge creates sudden voltage and current spikes	Power and signal isolators and fuses so that the discharge and inrush currents are not propagated.
Roboteq Motor Controller may stop responding to serial commands	Firmware bugs or driver issues	Reflash the firmware onto the controller
Loss of connection with the Wireless E-Stop	Communication faults or going out of range of the transceiver	Heartbeat on the wireless E-Stop to ensure communication is always possible, and stop the bot if connection is lost.

6.3. *All failure prevention strategy*

1. **Heartbeat:** This program built on ROS. The program consists of a client side process which monitors a device/process and checks for timeouts, measures latency, and error states and passes the status to a server. The server side process keeps track of all the device/process statuses and issues warnings if there are non-fatal errors or issues commands safely stop the vehicle if fatal errors are detected. This program written to be highly modular and therefore can be easily customized to monitor any device or process in general. Figure 14 shows the schematic for the specific configuration of heartbeat for Vikram.

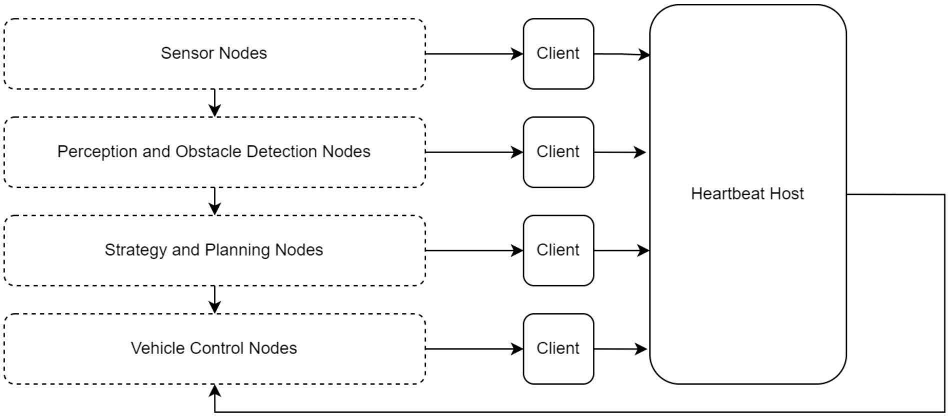


Figure 14: Heartbeat schematic

7. SIMULATIONS EMPLOYED

1. **CFD Simulations for Cooling Systems:**

- **Component Housing:** Four high volume flow rate axial fans (85cfm each) have been used to circulate air inside the main compartment. Airflow was modelled using CFD analysis tools, and components were placed accordingly, optimizing airflow requirements for each component and weight distribution of the vehicle. The system has been found to dissipate about 200W of heat during normal operation. (Fig. 15)

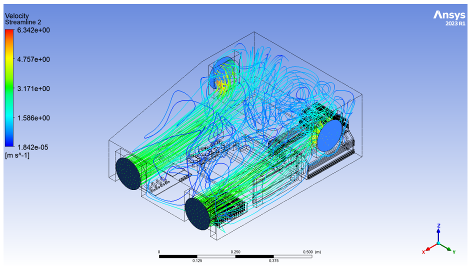


Figure 15: Airflow after component placement

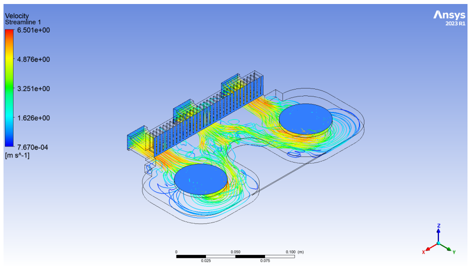


Figure 16: Airflow in Intake Manifold

- **NUC:** The NUC has a factory fitted thermal regulation system, yet it suffered from overheating. Hence, an external air channeling housing was developed consisting of internal fins and radial flow fans (2.4cfm each) to overcome the airflow resistance offered. The housing itself can be divided into the intake and the exhaust manifold, with the effective operating temperature drop coming to about 4°C. (Fig. 16)

2. **Structural Analysis of the Suspension Components:**

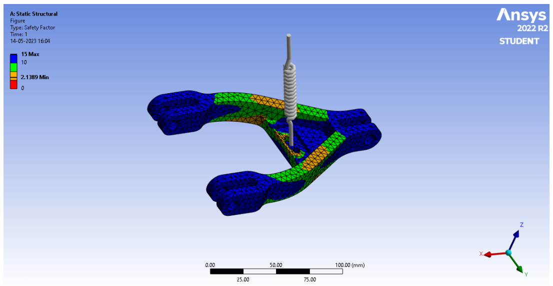


Figure 17: Static Wishbone Analysis

Wishbones and 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. Finite Element Analysis tools have been used along with CAD Modeling software to design the optimized parts.

3. **Integrated Simulation using Gazebo and RViz:** RViz is a 3D visualization tool that allowed us to visualize our robot's model and its sensor data in a simulated environment. Gazebo on the other hand provides a physics engine that complements RViz in simulating the physical properties of the robot model and its environment, enabling us to test and refine the control algorithms while simultaneously testing the ideal performance of Vikram in various scenarios. The combination of these two allowed us to achieve a great head start on our first real-life prototype. Very little major changes were required on the actual vehicle because of extensive testing on the simulated environments. Some images from the simulations deployed are shown in figure **??**

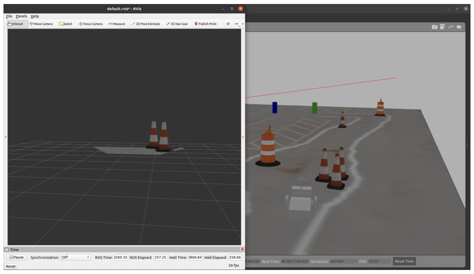


Figure 18: Ground view of the simulation

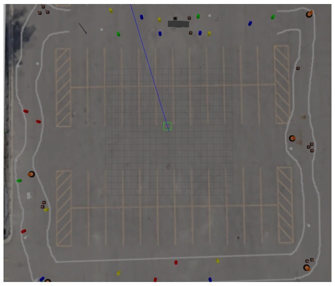


Figure 19: Top view of the simulation

4. **Controller Simulations** Simulations were employed to test PID and SPARC controllers on different system models. An example is illustrated for a linear and non-linear system by plotting response and Root-Mean-Square errors to periodic step inputs
(Green: Setpoint, Blue: SPARC, Red: PID)

(a) $\frac{dv}{dt} = k_1 \times \text{effort}(t - 0.3) - k_2 \times v$

(b) RMS Error

Figure 20: Linear Model

(a) $\frac{dv}{dt} = k_1 \times \text{effort}(t - 0.3) - k_2 \times v|v|$

(b) RMS Error

Figure 21: Non Linear Model

We observe that SPARC is marginally better than PID in a linear system, but the improvement is very evident while adding non-linear elements.

8. COMPONENTS AND SYSTEMS TESTING

8.1. *Suspension Testing*

The vehicle underwent multiple repeated rumbler strip tests, with and without a suspension system to prove improved stability. An evident difference in IMU readings was observed, with the flattening of jerk variations. The graphed data below illustrates this contrast.

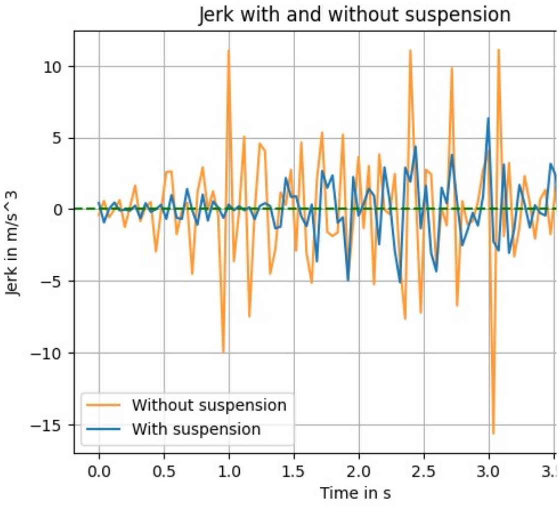


Figure 22: Jerk vs time

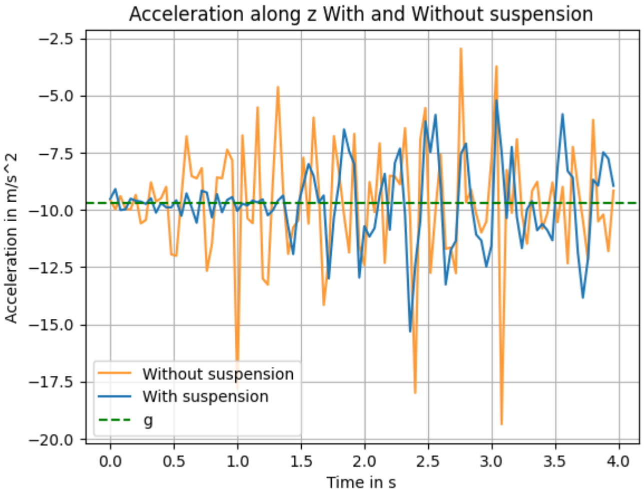


Figure 23: Jerk vs time

An appreciable change in the standard deviation of the acceleration and jerk readings:

Evaluating Element	**Extreme analysis**	**Nominal Analysis**
Acceleration	2.59g	1.71g
Jerk	4.08g/s	1.90g/s

9. INITIAL PERFORMANCE ASSESSMENTS

Evaluating Element	**Extreme analysis**	**Nominal Analysis**
Computing Unit	180W	80W
Runtime	1.25 hours	2 hours
Motor Currents (Mean)	6A + 6A (5 mph)	4A + 4A (1 mph)
Motor Currents (Peak)	12A + 12A	8A + 8A

Presently localization, pathplanning, lane detection, pothole detection and obstacle detection are working on the vehicle. Both PID and SPARC velocity controllers are able to set precise velocities, however a response delay of around 250ms is observed. Suspension Systems give evident improvements in system damping. The vehicle is able to navigate ramps with upto 20% gradient