Intelligent Ground Vehicle Competition 2022

Team Abhiyaan

Vajra

Indian Institute of Technology Madras

Design Report

I hereby certify that the development of the vehicle, Vajra, 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

Team Leaders

Shashank M Patil	shashankpatil0705@gmail.com
Amit Dodmani	amitdodmani4@gmail.com

Team Members

Business	**Electronics**	**Mechanical**	**Software**
Mahboob Ali	Adil Mohammed K	Sudharsan Raja	Ashwin Nayar
Sreeram Krishna R	AMS Arun Krishna	Aneesh Bhandari	Suraj Rathi
Raj Surya Kalieswaran T	Niyas Mon P	Drishti Darshan	Dhanush Anantharaman
	Adityan CG	Nikhil Chavan	Dhruv Maroo
	J Snehan	Atharva Jamdade	Hitesh Pawar
	Ruthika Kodiganti		Jnanesh Rompilli
			Suneet S
			Sachidanand V S

Project Members

Mechanical Module

Name	Department	Email ID
Atharva Jamdade	Mechanical Engineering	me18b102@smail.iitm.ac.in
Sudharsan Raja	Engineering Design	ed20b064@smail.iitm.ac.in
Aneesh Bhandari	Engineering Design	ed20b012@smail.iitm.ac.in
Drishti Darshan	Aerospace Engineering	ae19b103@smail.iitm.ac.in
Nikhil Chavan	Engineering Design	ed20s011@smail.iitm.ac.in

Software Module

Name	Department	Email ID
Ashwin Nayar	Naval Architecture and Ocean Engineering	ashwin.an001@gmail.com
Suraj Rathi	Mechanical Engineering	suraj.rathi00@gmail.com
Dhanush Anantharaman	Chemical Engineering	ch20b036@smail.iitm.ac.in
Hitesh Pawar	Mechanical Engineering	me18b161@smail.iitm.ac.in
Jnanesh Rompilli	Electrical Engineering	ee20b052@smail.iitm.ac.in
Suneet S	Engineering Physics	ep19b033@smail.iitm.ac.in
Sachidanand V S	Engineering Physics	ep19b010@smail.iitm.ac.in

Electronics Module

Name	Department	Email ID
Adil Mohammed K	Engineering Design	ed19b041@smail.iitm.ac.in
AMS Arun Krishna	Electrical Engineering	ee19b001@smail.iitm.ac.in
Niyas Mon P	Electrical Engineering	ee20b094@smail.iitm.ac.in
Adityan CG	Electrical Engineering	ee19b003@smail.iitm.ac.in
J Snehan	Electrical Engineering	ee19b027@smail.iitm.ac.in
Ruthika Kodiganti	Electrical Engineering	ee20b065@smail.iitm.ac.in

Business Module

Name	Department	Email ID
Mahboob Ali	Chemical Engineering	ch19b066@smail.iitm.ac.in
Sreeram Krishna R	Naval Architecture and Ocean Engineering	na19b012@smail.iitm.ac.in
Raj Surya Kalieswaran T	Aerospace Engineering	ae19b008@smail.iitm.ac.in

Draft version May 16, 2022
Typeset using LaTeX default style in AASTeX631

Contents

1. INTRODUCTION

Team Abhiyaan is a team of 20 interdisciplinary students enrolled in undergraduate and postgraduate engineering programs at the Indian Institute of Technology (IIT) Madras. Charged by a common passion for autonomy, we work in the student run Center For Innovation (CFI). We considered all the failures and shortcomings that Virat had faced in IGVC 2018 and 2019 and reassessed our thought process to come up with novel and innovative solutions. Vajra is our fourth prototype manufactured for the purpose of participating in IGVC and this design report serves to document all the details. Vajra is superior to its predecessor Virat in terms of both performance and reliability.

1.1. *Team Organization*

The team is organized into 4 modules: Mechanical, Electronics, Software and B & D (Business and Design); based on the expertise and domain knowledge that is required in building the vehicle. Modules have clear cut roles and responsibilities and team members do contribute to multiple modules. The team is overseen by a Faculty Advisor and a single Team Head. The Team-Head coordinates between the modules, organizes meetings, evaluates progress and reports to the faculty advisor. Each module has a head who is responsible for progress within the module. Module heads report progress to the team heads and each other.

The B & D module looks after all non-technical tasks including talent acquisition, public relations, strategic industrial relations etc. The Mechanical module has members working in Design, Simulation and Manufacturing of the prototype. It overlaps with the Software and Electronics team in the area of Control systems and Mechatronics. The Electronics team deals with the electronic circuitry, sensor data acquisition & signal processing and actuation. They overlap with the Software team in the area of embedded systems and microcontroller programming. The software team works with the development of software solutions for Autonomous Navigation.

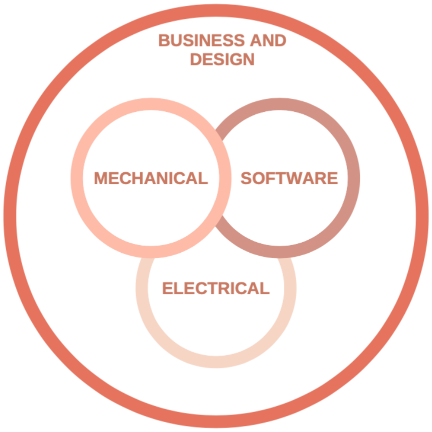


Figure 1: Team structure

2. INNOVATIONS

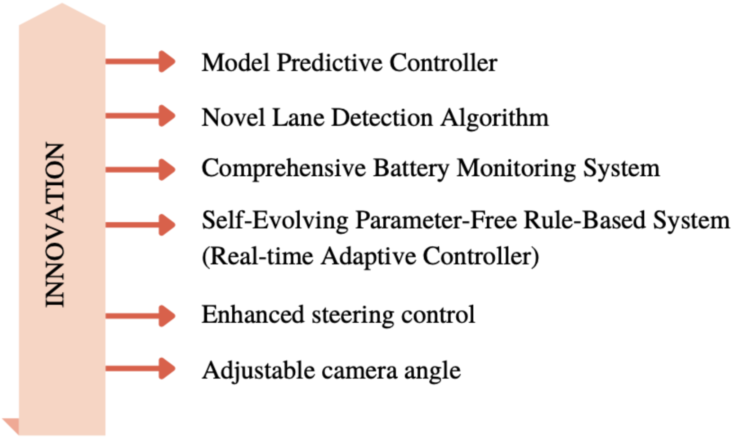


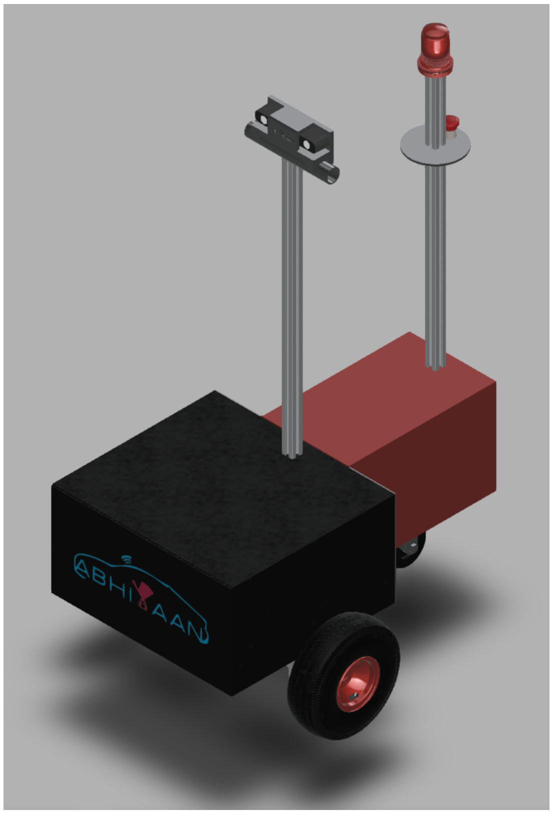
Figure 2: Innovations

3. DESIGN PROCESS

The current team members were primarily trained in online mode because of pandemic situation. Software module members gained expertise through simulation tools like Gazebo, while Electronics and Mechanical module members were trained in CAD tools like Eagle, LTSpice, Fusion 360 and Ansys.

When the pandemic biobubble was removed during November 2021, the team officially came to Centre For Innovation, IIT Madras and started working offline. Several challenges and constraints were faced in technical, logistics and procurement phase because of pandemic and inexperience of team members working in real-hardware setup.

The design process by the mechanical module was kickstarted by analyzing the shortcomings in the previous year's vehicle, Virat. After ideation and brainstorming the mechanical module came up with a new design for this year's vehicle, Vajra. The major improvements being better steering control and modularity. A CAD Model was then made on Autodesk Fusion 360 taking constant inputs from the electrical and software teams.



Electronics module spent initial couple of weeks were spent trying to get familiar with the previous hardware setup. Once the previous vehicle was in working condition, the shortcomings were noted down. Initially the tuning of control algorithms took significant time and effort, hence we decided to develop a self-evolving algorithm. Most of the PCB components were legacy components and out-of-stock, hence the entire PCB had to be redesigned.

The software module started by identifying each challenge we will face. The module created a simulated test setup targeting each problem, and focused on solving it in isolation. We also created and collated our own datasets for training models. After multiple iterations of our algorithms and code, we began to combine the different components, slowly building up the final stack.

4. MECHANICAL DESIGN

The primary objective of the mechanical module of Team Abhiyaan is to build a robust and efficient vehicle, taking into account software and electrical modules' requirements. While designing the vehicle, its structural rigidity, weight optimization, low center of gravity for increased stability, modularity, aesthetics, and compliance with rules were considered. Various problems faced by the previous year's vehicle were overcome.

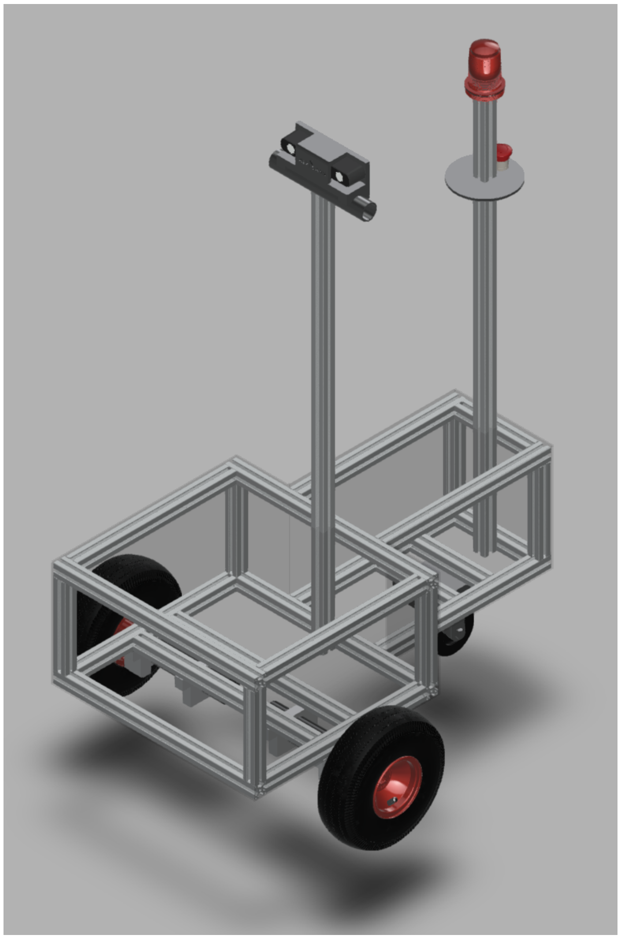


Figure 3: Robot CAD

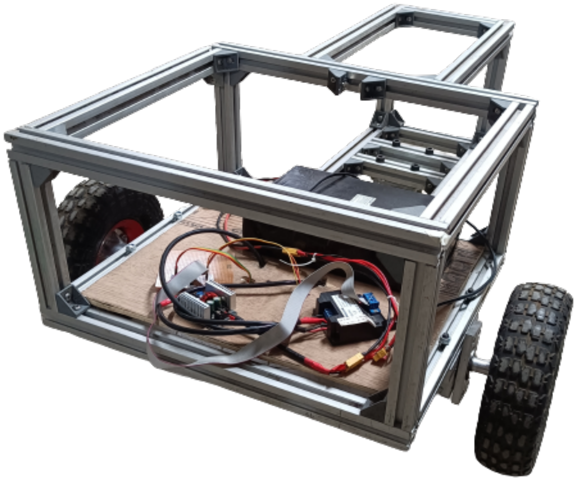


Figure 4: Manufactured robot

The process of designing the chassis consisted of:

1. **Material selection** — The chassis is the structure that bears the overall load of the vehicle and it needs to be strong and rigid enough to carry this load. Hence, we decided to use aluminum for the vehicle frame, wheel hubs, and motor mount due to its relatively low weight for its strength and cheaper cost compared to other materials
2. **Chassis design and analysis:**
   (a) **Components:** The vehicle consists of three wheels, two wheels connected to two independent motors (Maxon RE50 + Planetary Gearhead GP 52c) in the front, one caster wheel setup on the rear side, a stereo camera (StereoLabs ZED 2i) , strobe light, and the PCB housed inside
   (b) **Modularity:** The frame is extremely modular and serviceable as it is made of standard extrusion bars and angle brackets. The camera-angle is adjustable as well
   (c) **Space Efficiency:** Keeping the massive transportation constraints in mind, the vehicle is very compact yet it can support significant loads
3. **Motor and gear combination** — The possible motor types and gear ratio are selected by calculating the torque required for driving the vehicle at maximum required speed. Total effective effort can be given as:

$$T = \frac{\mathrm{TTE} \cdot D}{2 \cdot n}$$

where T = torque per wheel, TTE = total tractive effort, D = diameter of powered wheel.

$$\mathrm{TTE} = \mathrm{RR} + \mathrm{GR}$$

where, RR = Force necessary to overcome rolling resistance, GR = Force required to climb an inclination.

$$\mathrm{RR} = M \cdot g \cdot C_{\mathrm{rr}}$$

where M = Total weight of the vehicle, g = gravity, C_{rr} = surface friction.

$$\mathrm{GR} = M \cdot g \cdot \sin(\theta)$$

where θ = maximum inclination

$$\omega = \frac{2V}{D} \cdot \frac{60}{2\pi}$$

where ω = angular velocity of wheels.
For $M = 50\,\mathrm{kg}$, $g = 9.81\,\mathrm{ms}^{-1}$, $C_{\mathrm{rr}} = 0.055$ (grass) and $D = 254\,\mathrm{mm}$ (10 inch), $\mathrm{RR} = 27.06\,\mathrm{N}$, $\mathrm{GR} = 127.34\,\mathrm{N}$, $\mathrm{TTE} = 154.4\,\mathrm{N}$, $T = 9.8044\,\mathrm{Nm}$
Therefore,

$$V_{\max} = 2.23\,\mathrm{ms}^{-1}\ (5\,\mathrm{mph}) \implies \omega_{\max} = 167.67\,\mathrm{rpm}$$

$$V_{\min} = 1.34\,\mathrm{ms}^{-1}\ (3\,\mathrm{mph}) \implies \omega_{\min} = 100\,\mathrm{rpm}$$

According to the above results, the following combination of motor and gear was chosen:

S.No.	Specification	unit	Value
1	Nominal Voltage	V	24
2	Nominal Current	A	10.8
3	No load speed	rpm	5950
4	Nominal torque	mN m	405

S.No.	Specification	unit	Value
1	Reduction	-	66:1
2	Max. continuous range	N m	30
3	Mass inertia	$\mathrm{g\,cm}^2$	16.7

4. **3D modeling of parts** — We used Autodesk Fusion 360 for modeling the vehicle and to create manufacturing drawings. The chassis, all the sensors, motor assembly, wheels, and vehicle covering were modeled and assembled on the software. We also made sure that the vehicle looked aesthetically pleasing and incorporated some stylish designs while keeping all the IGVC rules in mind

5. **Major Changes** — We have improved the design of our frame considerably with the major change being the alternate location of the caster wheel on the chassis frame, this reduced the effort required for steering significantly. Unlike our cars which utilize a differential so that the rear wheels can rotate while turning and the turn is guided by the front wheels (with a positive caster), the front wheels on our vehicle are made to rotate at different speeds so that the vehicle turns and the caster wheel merely supports the rest of the weight and follows the turn.

4.1. *Manufacturing & Assembly*

1. The aluminum extrusion bars were then cut and assembled with the help of L-clamps using power tools like drilling machines and angle grinders.

2. Motor mounts and wheel hubs were manufactured using CNC machines for high precision

3. For the sheet metal components, laser-cutting, shearing machine, and press brakes were used. Some of the sensor mounts were 3D printed.

4. A strict timeline for manufacturing was followed so that ample time was devoted to troubleshooting and reiterating the design

5. **Weather Proofing** — Vajra can be operated in the event of light rains as all the electrical and software components have been covered with suitable waterproofing material. The stereo camera (IP66 rated for dust and water resistance), the LIDAR, and the GPS body are not enclosed since they are waterproof

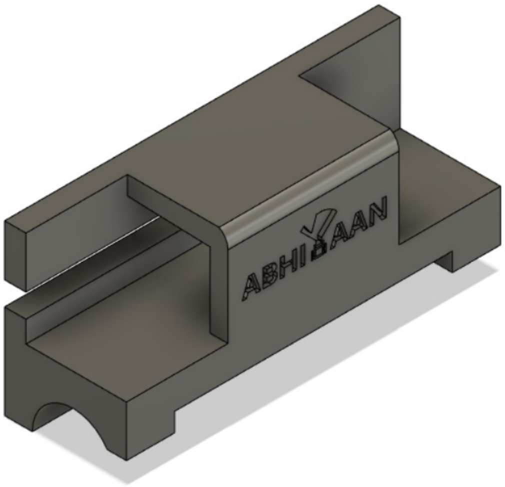


Figure 5: Stereocamera Mount

5. ELECTRICAL DESIGN

A major objective of Vajra's electrical system was to create a more resilient system through various fail-safe mechanisms, and more robust components. Prominent improvements from the vehicle's previous versions have been seen in the areas of

1. Developing an in-house comprehensive Battery Monitoring System for safe and efficient use of Lithium-Ion Batteries
2. Designing a compact and customized PCB for robustness and reliability.
3. Developed a robust control algorithm for achieving high precision in speed control

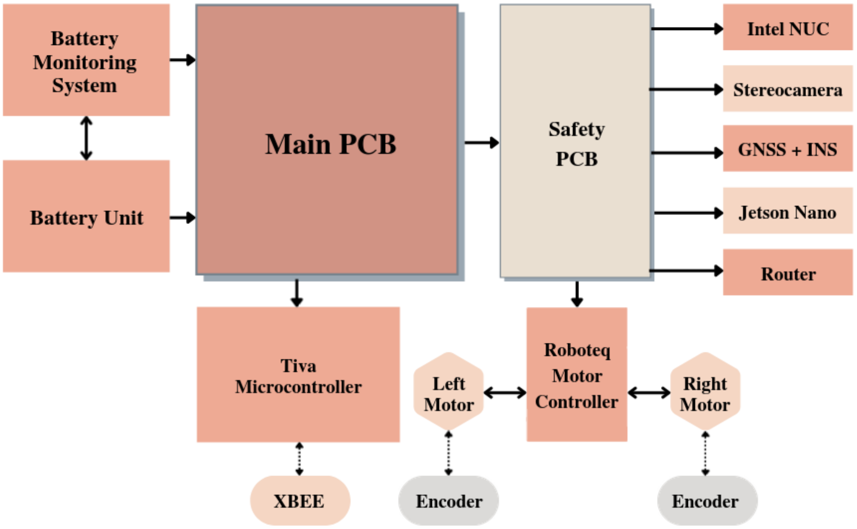


Figure 6: Overall electrical system

5.1. *Control PCB*

The Control and Safety PCBs of Vajra have been designed in Autodesk EagleCAD. The Control PCB houses the entirety of electronics components - the Relays, Microcontrollers, LED Driver, Power isolators, and Wireless Modules.

1. The power supply to every main component and sensor is controlled using electromechanical relays through the Tiva TM4C123GXL microcontroller.
2. Power and Signal Isolators have been used for the safe use of encoders and motor controllers.
3. To denote whether the vehicle is in autonomous or manual mode to the user, the vehicle is equipped with a LED driven by the LED Driver ILD6070.

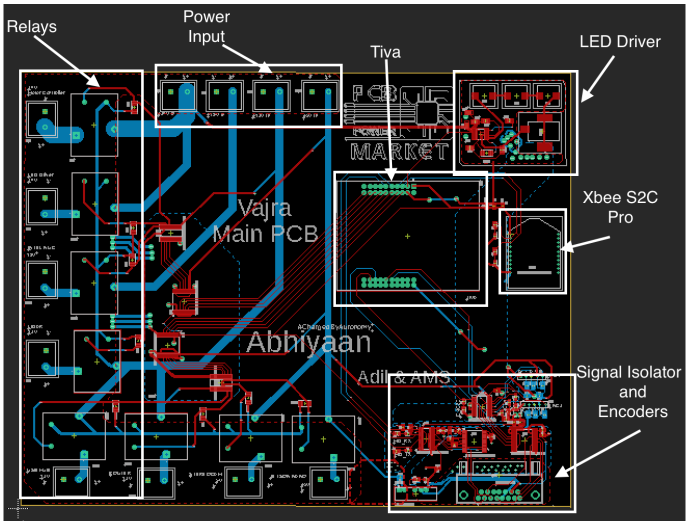


Figure 7: Control PCB

Several test points and LED Indicators have been placed throughout the PCB for troubleshooting any problem. The PCB has been designed following most of the general PCB guidelines minimizing noise and electromagnetic interference.

5.2. *Safety PCB*

The Safety PCB is used to connect the main components and sensors - Intel NUC, Jetson Nano, StereoCamera, Motor Controller, etc to their respective power supplies. Safety has been given utmost importance - with placing fuses and reverse-polarity protection diodes in all of the power-transmission cables. It has been made separate from the main PCB for ease of use during the development stages.

5.3. *Emergency Stops*

The vehicle can be stopped at any point of operation through the following measures:

1. Mechanical Emergency Stop
2. Wireless Emergency Stop

Both the Mechanical E-Stop (Red Color Push button) and Wireless E-Stop only disconnect the power to the motor without affecting the components.

The Wireless E-Stop is implemented using XBee S2C Pro wireless modules which have a range of up to 2 km. When the XBee receives a signal for Emergency Stop, the information is immediately communicated with the Tiva microcontroller through UART communication. Tiva immediately switches off the power to the motors through electromechanical relays.

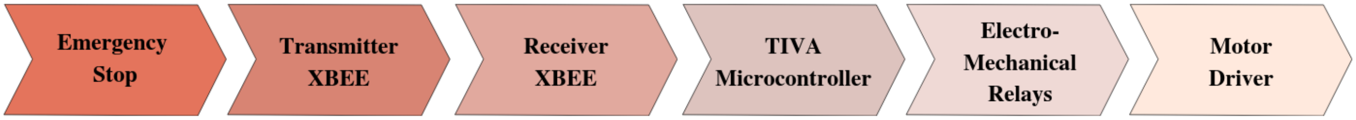


Figure 8: Emergency stop flow diagram

5.4. *Power Distribution*

The Vehicle is powered by a single Lithium-Ion Battery 25.6V 48Ah (1250 Wh). With the help of the off-the-shelf DC-DC step-down Buck converters, Each of the components is powered at the rated voltage levels as shown in the figure. All the electrical connections were made using XT-60 connectors that satisfy the required current ratings.

We had earlier designed our PCB with onboard-Buck Converters using TPS55288 IC - But due to the unavailability of the IC, we have removed it from the current PCB iteration.

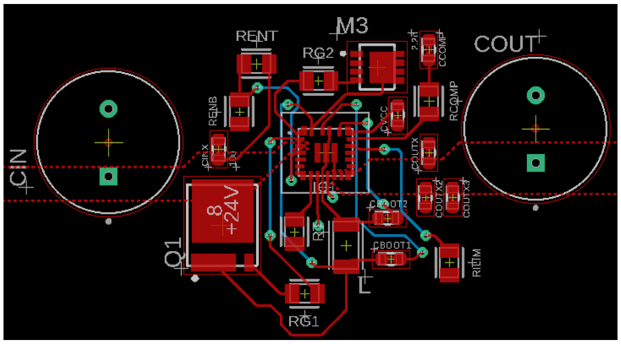


Figure 9: TPS55288 Voltage Regulator

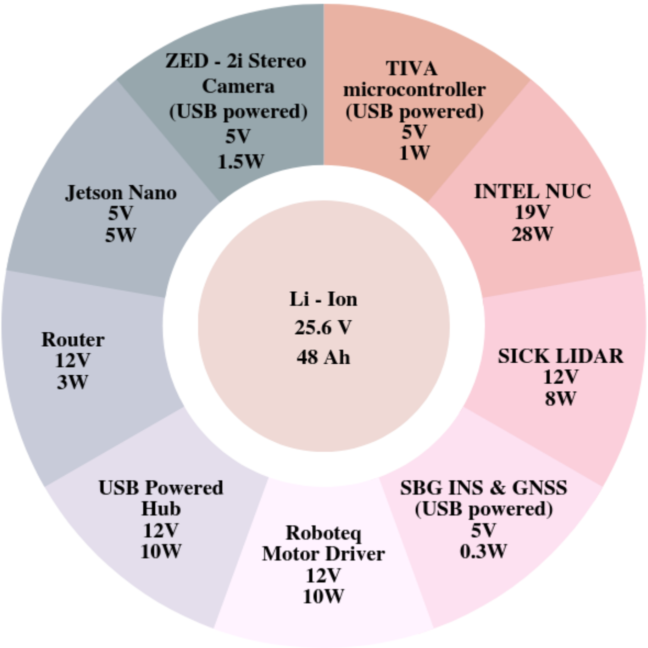


Figure 10: Power Distribution System

Provisions for 2D LIDAR have been kept in place, in the case the software module requires them. Based on practical observations, we were able to conclude that the Maxon motors required to drive the vehicle had the maximum power consumption estimated to be around 220-240 W.

We estimate that the total power consumption would be around 300W - which would result in a runtime of around 3 hours which was observed during our trial runs. The battery was operated only under nominal conditions (avoiding under and overvoltage) for improved battery health.

5.5. *Battery Monitoring System*

For the safe and efficient use of the Lithium-Ion Battery, a comprehensive and reliable Battery Monitoring System is being developed. The BMS consists of an Analog Front End (AFE) Device BQ76930, a microcontroller BQ78350, and FET Driver BQ76920.

The companion microcontroller BQ78350 communicates with the AFE device to monitor individual cell voltages (all seven series cells), temperature, battery current, State of Charge, and Health. The SOC is implemented using the Current Integration Method.

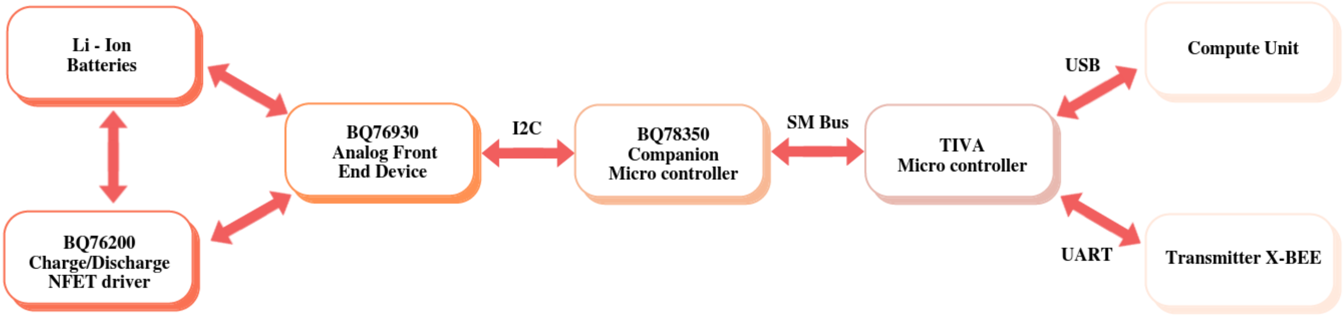


Figure 11: Battery Monitoring System

The health of the battery is periodically communicated to the Tiva Microcontroller by BQ78350 through SMBus Communication protocol. This is relayed to both the Compute Unit and to the user through the XBee wireless module.

The schematics for the circuit have been designed in Altium, with the PCB Layout in progress. Tiva Microcontroller code for communication with the BMS and XBee has been implemented and verified.

5.6. *Controller Design*

The Maxon motors that drive the vehicle are controlled by Roboteq SDC2160 high-performance dual-channel brushed DC motor controller with closed-loop feedback using Avago HEDM-5500 quadrature encoders.

The motor controller has inbuilt PID control whose gains can be tuned to attain maximum precision in speed control. If there are any drastic changes in the weight of the vehicle, such as a different payload, PID control gains have to be appropriately tuned again.

5.6.1. *SPARC Controller*

To avoid this constant re-tuning of gains, a novel ANYA-based Self-Evolving Parameter-Free Rule-Based Controller(SPARC) system has been implemented. This ANYA-based system is a real-time Adaptive controller that uses selected previous data samples as representative focal points describing the plant (here, the vehicle).

This method compares the closeness of the current data point with the representative focal points; The focal points with common properties to the current datapoint contribute more to the control input than the others. Since the controller is entirely data-driven, the controller is auto-tuned in every iteration - thus, specific tuning when the load changes are no longer required.

Based on practical experimentation, we observed that the SPARC controller had a very high precision in speed control. Although an improvement was observed, the SPARC controller is overkill for Vajra. Hence, we decided to revert back to the traditional PID controller. The SPARC controller has been reused in the throttle control of the Golf cart where the load variation, non-linearity of the system, and noise disturbances are much higher. Mean Squared Error(MSE) was significantly reduced in the golf cart compared to traditional PID and Fuzzy controllers.

6. SOFTWARE DESIGN

The software stack is built with modularity in mind. We have tried to modularize every independent aspect of the software stack. This leads to mainly four independent modules:

1. Perception
2. Localization and mapping
3. Path planning
4. Controller

There's also a monitor node which monitors the status of all the modules and their submodules. The modularity leads to better interoperability with other such modular architectures. It also leads to easier debugging and development. The overhead to this modular approach is only the extra communication required between each of the modules, which is practically negligible when using any good inter-process communication framework.

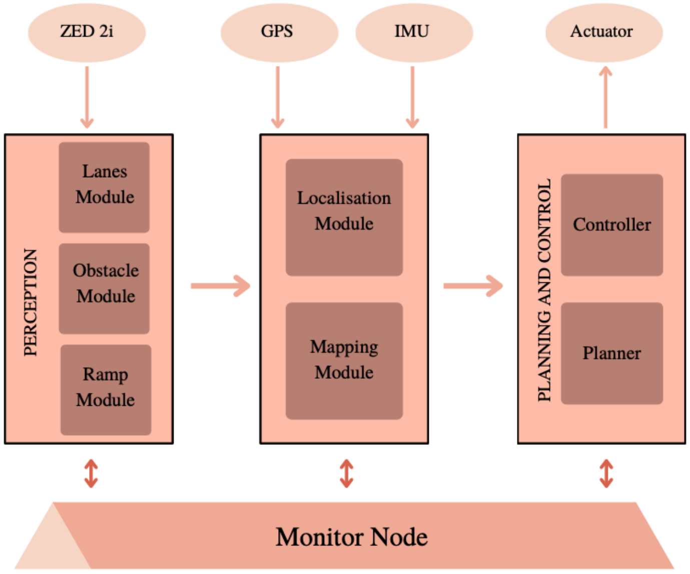


Figure 12: Overall Software Architecture

We use the Robot Operating System (ROS) for Vajra. ROS facilitates continuous data transmission between all the modules and nodes even when running on multiple devices. Our software is written in C++ and Python. We have prioritized efficiency and stability, and made it resilient to the failure of any node. The modular approach makes it easier to do so, leading to a very robust software stack.

6.1. *Perception: Object and Lane Detection*

The ZED 2i stereo camera is used to perceive the environment. It provides us with a high quality 50 Hz depth image stream. From this stream we use a combination of RGB and depth data to perceive lanes, obstacles, and ramps.

The previous iteration of our AutoNav vehicle used a 2D Lidar. This year we switched from the LiDAR to the Stereo Camera as it provides sufficient precision at the ranges we work in. The Stereo Camera is also much less expensive.

6.1.1. *Object Detection*

Multiple methods of object detection are fused. On the raw 3D point cloud data a coordinate filter and outlier filter is applied. The remaining points are down sampled onto the occupancy grid used by the path planner.

A model using YOLO-v5 architecture is run on the RGB video stream. Using transfer learning and a custom made dataset we had the model work with our required object classes. The model's output is then projected to 3D space. The final results are added to the occupancy grid used in the costmap.

We had planned to track the obstacles in time via Kalman filtering. This will help us with model failure and occluded obstacles. We also wanted to look into ways to directly combine depth and height data with the ML model. Both of these features were not necessarily required for this task, but this is something we are actively exploring for our spec 2 vehicle.

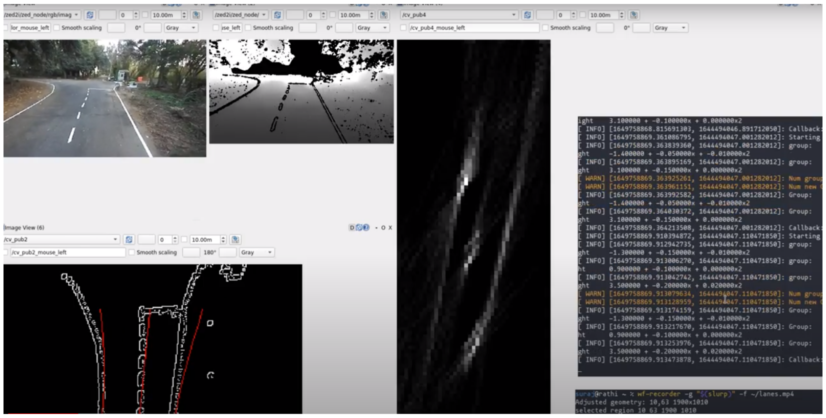
Figure 13: Object detection

6.1.2. *Lane Detection*

Lane detection is done using a heuristic based OpenCV pipeline. The procedure is as follows:

1. Apply a series of point cloud filters
2. Transform the image to a bird's-eye perspective
3. Apply filtering based on color, edges and height
4. Apply some post processing before projecting it to the ground plane to obtain a three dimensional point cloud
5. Then, this data is fused with the output provided by a pretrained LaneNet model (from the Tusimple dataset). It is based on semantic segmentation and encoder-decoder layers.

We developed a new module for lane detection. It filters the output of the previous module using characteristics of the lane, namely, the shape, thickness, and separation. The lanes detected in previous frames are used to augment the current detection, i.e. the algorithm has "memory". We have implemented this method, but further testing is required to improve robustness and deal with long term lane occlusion.



```
ight     3.100000 + -0.100000x + 0.000000x2
[ INFO] [1649758868.815691303, 1644494046.891712050]: Callback:
[ INFO] [1649758869.361086795, 1644494047.001282012]: Starting
[ INFO] [1649758869.363839360, 1644494047.001282012]: group:
ght      -1.400000 + -0.050000x + -0.010000x2
[ INFO] [1649758869.363895169, 1644494047.001282012]: group:
ght      3.100000 + -0.150000x + 0.000000x2
[ WARN] [1649758869.363925261, 1644494047.001282012]: Num group
[ WARN] [1649758869.363961151, 1644494047.001282012]: Num new
[ INFO] [1649758869.363992582, 1644494047.001282012]: Group:
ght      -1.400000 + -0.050000x + -0.010000x2
[ INFO] [1649758869.364030372, 1644494047.001282012]: Group:
ght      3.100000 + -0.150000x + 0.000000x2
[ INFO] [1649758869.364213508, 1644494047.001282012]: Callback:
[ INFO] [1649758869.910394872, 1644494047.110471850]: Starting
[ INFO] [1649758869.912942735, 1644494047.110471850]: group:
ght      -1.300000 + -0.150000x + -0.010000x2
[ INFO] [1649758869.913006270, 1644494047.110471850]: group:
ht       0.900000 + -0.100000x + 0.000000x2
[ INFO] [1649758869.913042742, 1644494047.110471850]: group:
ght      3.500000 + -0.200000x + 0.020000x2
[ WARN] [1649758869.913079634, 1644494047.110471850]: Num group
[ WARN] [1649758869.913128959, 1644494047.110471850]: Num new
[ INFO] [1649758869.913174159, 1644494047.110471850]: Group:
ght      -1.300000 + -0.150000x + -0.010000x2
[ INFO] [1649758869.913217670, 1644494047.110471850]: Group:
ht       0.900000 + -0.100000x + 0.000000x2
[ INFO] [1649758869.913253976, 1644494047.110471850]: Group:
ght      3.500000 + -0.200000x + 0.020000x2
[ INFO] [1649758869.913473878, 1644494047.110471850]: Callback:
_
```

```
suraj@rathi ~ % wf-recorder -g "$(slurp)" -f ~/lanes.mp4
Adjusted geometry: 10,63 1900x1010
selected region 10 63 1900 1010
```

Figure 14: New Lane Detection Module

6.1.3. *Ramp Detection*

The iterative RANSAC algorithm is applied on the three dimensional point cloud provided by the stereo camera to detect the ramp. The ramp boundaries are obtained and are treated like lane lines in the path planner. This enables us to smoothly integrate the ramp with the rest of the software stack.

6.2. *Localization and Map generation*

Figure 15: Localization Results

The main sensory data for localization comes from the Camera, INS, and GNSS. For the fusion of the above data, a nonlinear unscented kalman filter is used on the 15-dimensional state of the vehicle. The above strategy facilitates the integration of multiple sensors with different in a continuous manner. The Fig-15 shows the results of one of our tests where we took the vehicle on a 450m loop. The loop closure error was under 5m (around 1

The above data is used to provide the coordinate frame transforms between the vehicle and the frame attached to the ground. An occupancy grid is updated from the lane and obstacle data. This will be used for path planning. We also add an arc-shaped fake obstacle, 5 meters behind our vehicle in the costmap, in order to prevent the path planner from generating a path that requires Vajra to go backwards. Currently, the map isn't being saved for future runs but this can be done in the future.

6.3. *Path Planning*

The path planning module receives the GPS waypoints and tries to plan a path through the waypoints with the help of a global and a local planner. The planner searches for a feasible and the optimal path. It does this through various algorithms, like A*, Dijkstra, rapidly-exploring random trees (RRT) and so on. We use RRT for our planners, since it seems to work well with differential drive vehicles. RRT explores the search space by trying to form connections between random samples and the nearest state in the currently explored tree.

The planning module mainly consists of two planners, the global planner and the local planner. Both the planners run at different frequencies, the local planner being run much more frequently. The role of the global planner is to plan a path considering all the global waypoints and obstacles (found in the global costmap). The local planner's role is to plan a local path, which tries to fit the global path while also accounting for the local obstacles (like barrels and dynamic/moving obstacles, found in the local costmap)

6.4. *Controller*

The controller's role is to make sure that our vehicle follows the path generated by the planner as closely as possible. For this purpose, a non-linear model predictive controller with the receding horizon mechanic was designed and implemented. The amazing open source IPOPT library as the backend.

We set up the MPC problem that tries to find the optimal acceleration at each time step. We can apply various constraints ranging from limits on the acceleration to maximum deviation from the path. The controller can adapt well to changing conditions.

MPC provides a flexible, open and intuitive formulation in time domain. Due to its ability to optimize the current time slot while keeping future time slots in account, motion of the bot will be much more continuous and smooth. This makes the bot much more efficient when it comes to power consumption and energy loss. It allows for constraints on state parameters.

6.5. *Monitor*

The monitor script makes sure that all components of the software stack are operating as expected. It sends alerts whenever one of the nodes fails and can try to recover the dead node if possible. Otherwise it will take emergency actions. It also provides various statistics and diagnostics regarding the running nodes.

6.6. *Simulations*

Our testing strategy for Vajra involved testing our algorithms on simpler test scenarios before running them on the full vehicle model, i.e. we test individual modules while assuming the ground truth results from other modules before testing as a whole.

Most of our testing was done using Gazebo, an open-source platform as it is highly integrated with the ROS communication framework. We also used Airsim, a more realistic simulated environment in order to carry out image processing lane detection testing as it has more accurate lighting and textures. For our simulations in Gazebo, worlds that closely resembled the IGVC track were created using meshes. The barrels and lane lines were then added. An exact replica of our vehicle was made in URDF format making sure to add the required physics (joints, collisions, inertia) and sensors as plugins. The sensor data thus obtained was fed into the various other submodules for us to test our algorithms.

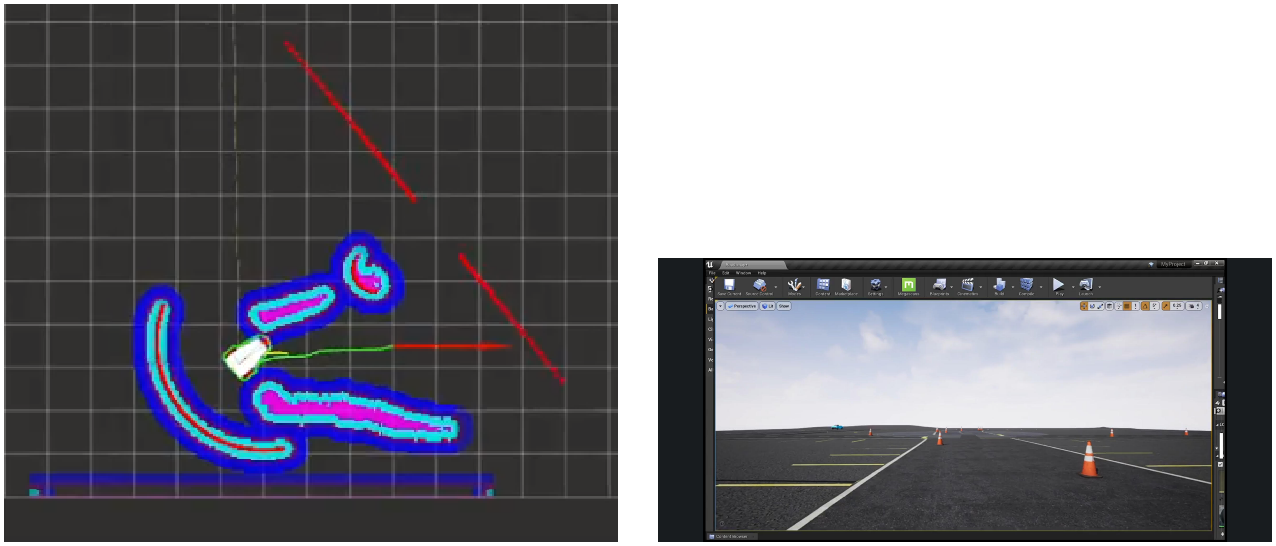


Figure 16: Glimpses from our Simulation Testing

7. FAILURE MODE AND RESOLUTION

7.1. *Mechanical*

S.No	**Failure Mode**	**Resolution**
1	Vibrations caused by loosening of the bolts	Use washers and lock nuts
2	Wobbling or bending of wheels	Tighten the nuts joining the wheels to the computer as soon as problem is observed

7.2. *Electrical*

S.No	**Failure Mode**	**Resolution**
1	Components should not be connected in reverse polarity — else it will destroy the component	Reverse-Polarity protection diodes have been placed in all the power-transmission lines
2	Encoder fails resulting in uncontrolled motors	Motor-Stalling has been introduced in this version. If no rotation is detected through the encoders when power is applied for a given time, the power to the motor is cut by the motor controller
3	Loss of Serial Communication between motor controller and Jetson Nano	Watchdog timer has been implemented that stops the motor if no command has been received for a given period of time
4	Suppose there exists a direct path between the signal ground of the motor controller and ground terminal of power supply. If the ground terminal of the motor controller is disconnected while the power terminal is connected, all the current will flow through the signal ground, damaging the controller and its peripherals.	Ground terminals are always connected to the controller — only the power terminals are disconnected. The signal and power lines of the motor controller are completely isolated. RE0505S is used to create an electrically isolated power supply for signal transmission from the main supply — effectively acting as a secondary battery. The data signal transmission lines are also isolated with the help of digital isolators.

7.3. *Software*

S.No	**Failure Mode**	**Resolution**
1	Blocking/occlusion of obstacles or lanes by other obstacles	Persistence mechanism has been implemented, which extrapolates the previous positions of the lanes and obstacles to estimate their current positions in case they are not visible
2	Crashing into an obstacle, hitting a dead end or reaching an area where no more navigation is possible	Recovery behaviors have been implemented and tested for navigation and planning, which can be used effectively to recover from such scenarios

8. COST ESTIMATION

S.No.	Component	Retail Cost (USD)	Team Cost (USD)
1	SBG Ellipse E IMU	4100	0
2	Intel NUC	1000	0
3	Printed Circuit Board	120	0
4	Roboteq SDC2160	125	125
5	TIVA TM4C123GXL	15	15
6	XBEE S2C Pro	15	15
7	Maxon Motors	937	629
8	Maxon Gears	738	508
9	Encoders	130	130
10	ZED 2i	500	500
-	Total	7,680	2,042

9. ACKNOWLEDGMENTS

We would like to thank our institute IIT Madras for providing us a platform to work on our own autonomous robot and giving us the chance to participate in IGVC 2022. We would like to thank the Center For Innovation (CFI) for supporting us through this journey to IGVC. We would also like to thank our faculty advisor Prof. Sathyan Subbiah for guiding us all along.