

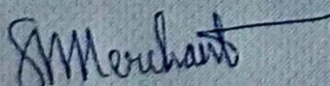


SeDriCa, IIT Bombay IGVC - 2016

Design Report

Statement from Faculty Advisor:

I hereby certify that the development of the vehicle, SeDriCa, described in this report is significant. This report has been prepared by the students under my guidance.


Prof S N Merchant

SeDriCa - DESIGN REPORT FOR IGVC 2016

Indian Institute of Technology Bombay

A Gupta, S Jain, A Sharma, R Choudhary, A Kishore, A Kumar R, T Kotwal

Prof. S N Merchant (merchant@ee.iitb.ac.in)

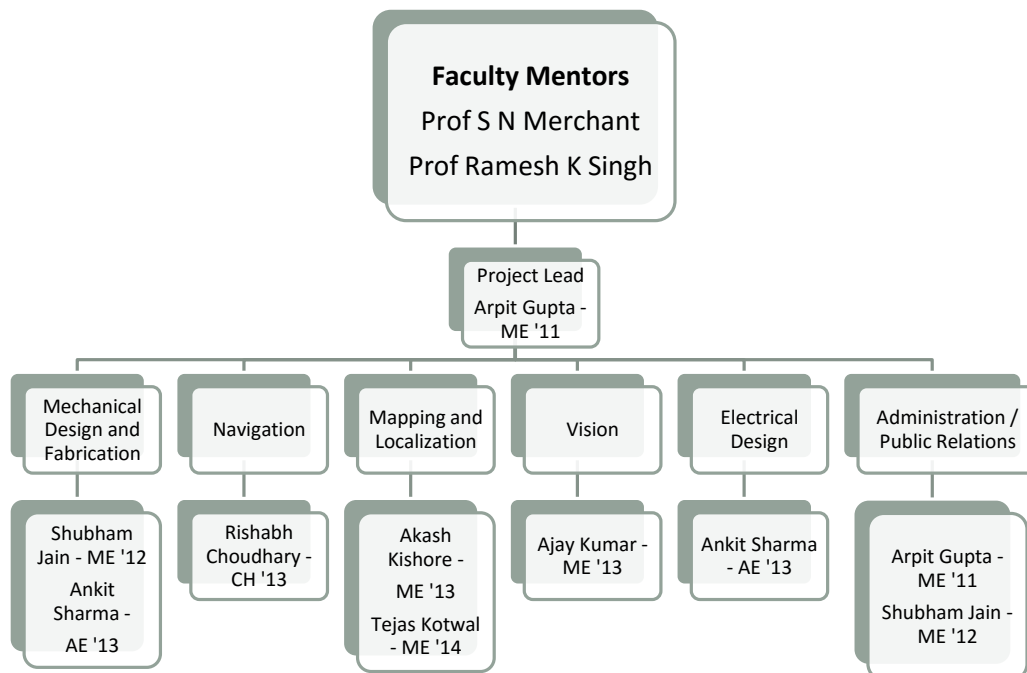
1. Team Overview

1.1 Team Selection

The Intelligent Ground Vehicle Competition provides an excellent opportunity for the student fraternity of the Indian Institute of Technology Bombay (India) to explore the design and implementation of advanced unmanned systems. The IIT Bombay team consists of students of various engineering disciplines, many of whom have worked on Mahindra RISE Driverless Car Challenge, ASME Challenges and various other national and international robotic challenges.

1.2 Team Identification

The team has 7 student members from various engineering disciplines. Considering the small size of the team, a flat team structure was opted for. The team has students involved in multiple roles as per requirement of the project. The team structure is as following:



1.3 Design process

Two major factors contribute to the fast convergence of the design process:

- Extracting maximum features from sensor data
- Excellent cross sensor correlation to increase accuracy and confidence level of features

The following flowchart explains the design process and flow diagram of the complete process.

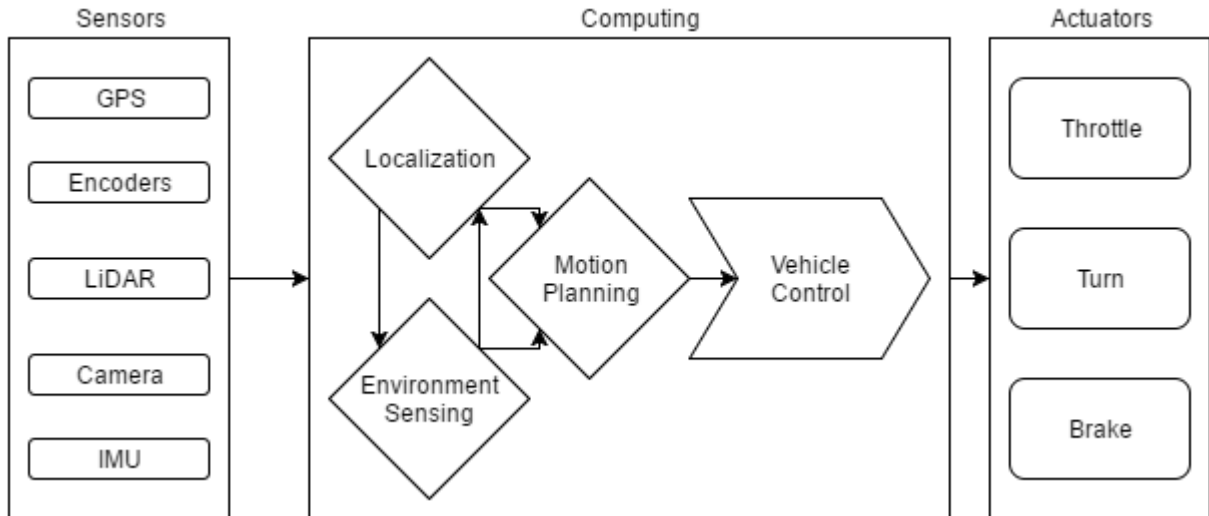


Figure 1:

2. Innovation in Vehicle Design

2.1 Robustness

- a. Housing with two bearings is used to get smooth ride even during high loads and shocks.
- b. Flange coupling is used to take care of any misalignment between motor and housing. This coupling prevents transmission of high loads and shocks to the motor.
- c. High strength steel (EN-24) is used in drive shaft to take high loads and unwanted shocks.

2.2 Efficient space utilization

- a. The vehicle has three layer/floors (G, 1, 2) with G floor containing powertrain and controllers, 1st floor containing batteries, inverter, suspension and payload, and 2nd floor having laptop, LiDAR, GPS and camera.
- b. This arrangement helps in maximizing space utilization and increases the stability of the vehicle by having low center of gravity.

2.3 Modularity

- a. The system is highly modular with each floor being independent of each other. It came out to be extremely useful when individual changes were to be made in the electrical or mechanical aspect of machine in a short span of time.
- b. This independence also makes the system clean with very few tangling wires in the system.
- c. Also this design comes in very efficiently in transportation, the three blocks can be fit to occupy very less volume and can be engaged-disengaged in about 30 minutes.

2.4 High degree of freedom

- a. Camera has 4 degrees of freedom, 2 each for translation and rotation. This helps in adjusting the camera view according to the requirement.
- b. LiDAR has 2 degrees of freedom which helps it in adjusting according to obstacle height.

3. Vehicle Structure

3.1 Salient features of SeDriCa are:

- It is a simple 4 wheeled differential concept with two driving motors and two castors, one at the front and other at the back.
- Chassis of the vehicle is made using Aluminum 6063-T6 profile pipes.
- Drivetrain comprises of motor, flange coupling, bearing housing and wheels.
- The height of the machine can be adjusted as per the requirement of the camera.
- Suspensions and dampers are incorporated to ensure smoother ride and reduce equipment failure due to shock and vibration.

Table 1. Design Specifications of SeDriCa

Parameter	Value
Dimension	
Length*Width*Height	43*26*56 inch
Wheel Diameter	20 inch
Front Castor Diameter	10 inch
Back Castor Diameter	6 inch
Mass	
Chassis	24 kg
Sensors (LIDAR, GPS, IMU & Camera)	3 kg
Processing Unit (Laptop & Dock)	3 kg
Panel (PLC, HMI, Batteries & Controller)	15 kg
Drive Train (Motors, wheels, couplings)	25 kg
Suspension Assembly	2 kg
Total	72 kg
Motor Specifications	
Outer diameter	50 mm
Nominal Speed	5680 RPM
Nominal (Max. continuous) Torque	0.405 Nm
Nominal Power Rating	200 W
Gearbox Specifications	
Outer diameter	52 mm
Reduction	66 : 1
Max. continuous torque	30 Nm
Output	
Nominal (Max. continuous) Shaft Speed	86 RPM
Nominal (Max. continuous) Shaft Torque	26.73 Nm
Max Speed	5.1 mph

3.2 Frame structure

- The main focus for choosing Aluminum profile sections is to make a modular and easy to assemble system.
- Load bearing capacity, functionality and design were decisive factors while choosing profile.

3.3 Powertrain

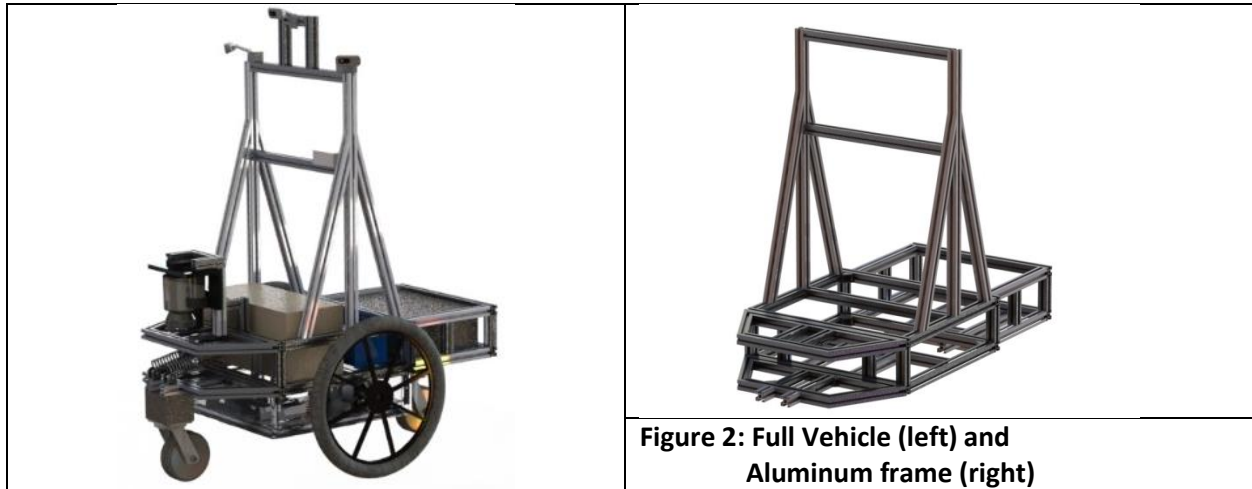
- The maximum load on machine is during a 15° incline. Taking the mass and diameter as mentioned in the table 1 we get load on each motor as

$$\zeta = \frac{mgR\sin\theta}{2} = \frac{72 \times \sin 15 \times g \times 0.254}{2} = \mathbf{23 \text{ Nm}}$$

- The machine has to achieve an average speed of 1 mph (0.44 m/s). We designed the machine to have a max speed of five times this value. Hence, the speed of the motor at rated load is

$$\omega = \frac{4v}{r} * \frac{60}{2\pi} = \frac{0.44 \times 5}{0.254} \times \frac{60}{2 \times \pi} = \mathbf{83 \text{ RPM}}$$

- Flange Coupling was chosen keeping in mind the possible misalignment between motor shaft and shaft from housing.
- Two bearings were used in a housing for enhancing strength and robustness of the drive assembly.
- Self-aligning bearing is used to prevent high stresses due to misalignment between motor shaft and shaft from housing.



3.4 Structure Design

- The shape of structure is designed for easy maneuverability in crunch spaces.
- Efficient space use, low center of gravity and stability were factors while choosing the shape.

3.5 Suspension

- Suspension works along the pitch direction as per the requirement of the terrain.
- This also reduced requirement of space and time for spring mounting.
- The vehicle also has custom designed anti vibrating mounts for laptop, LIDAR and Camera to ensure minimum noise in the sensor readings and to reduce equipment failure.

3.6 Weatherproofing

- The frame is completely covered with acrylic sheets. The sheet is low in weight and also prevent any unwanted material from entering the machine.
- The camera mount is made waterproof to prevent it from drizzle.

3.7 Efficient use of material

- Aluminum is used to make frame of the vehicle to reduce the weight
- Industrial grade steel (EN 24) is used in drive shaft and coupling to take high loads.
- Acrylic is used to cover the structure to reduce weight.

4. Description of Electronic and Power Components

- The vehicle is driven by two 200 W Maxon Motors. The driving battery is directly connected to the Motor Controller which is responsible for powering motors.
- A separate power system is there for uninterrupted CPU performance. The same battery system is also used for powering LIDAR and other sensors.

4.1 Power Distribution

Table 2. Design Specifications of SeDriCa

Average Power Drawn (W)	Peak (W)	Average (W)
Driving Power		
Motors	400	120
Roboteq HDC2450	10	8
CPU Power		
Sick LMS 111 LIDAR	25	20
Sparton AHRS 8	0.33	0.33
VectorNav VN-200 GPS	0.2	0.2
USB Cameras	2	0.8
ThinkPad P50 Laptop	170	120
Status Light	10	6
Total	617.53	275.33

4.2 Driving Power

- One 10000mAh/22.2V Battery connected to Roboteq HDC2450 Motor Driver Powers both Maxon motors. The controller's two motor channels can either be operated independently or mixed to set the direction and rotation of a vehicle by coordinating the motion of each motor.
- The Emergency switch disconnects the motors which doesn't let the vehicle to stop suddenly. It also keep the motor controller powered up to diagnose the problems.
- The vehicle has non removable driving battery so while charging we need to switch off the mains supply. Charging can be done with 5 Amp charger. At average load of Motor and Motor Driver (128 W) we can expect about two hours of run time.

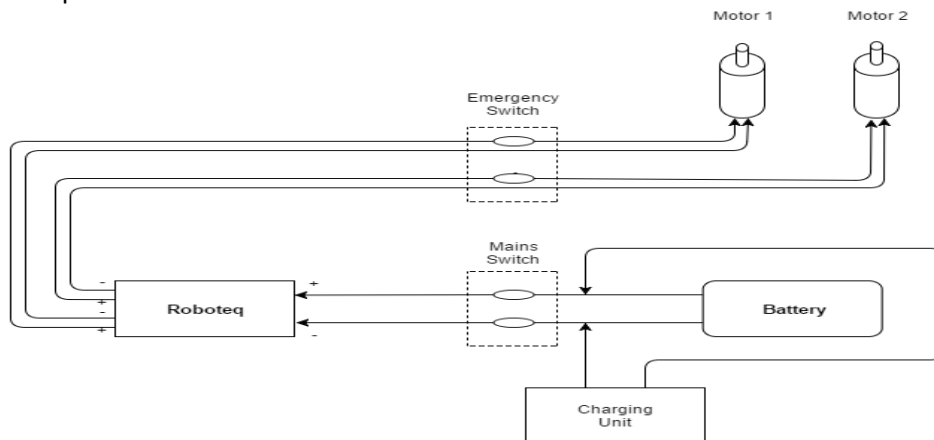


Figure 3: Drive Power Circuit

4.3 CPU Power

Four 12000 mAh Battery/ 22.2 V are connected parallel to Inverter which Powers Mechanical Docking Station of Lenovo P50 Laptop. The P50 is the main and only processing unit for the vehicle. The laptop itself powers IMU, GPS and 3 Cameras via USB. The 22.2V raw supply is also used for LIDAR. Total average load is around 150W. We get around 6 hour runtime of all the sensors and Laptop. The battery pack takes around 4 hour to fully charge. We can always charge the pack when the vehicle is not in the motion with the inverter even if the sensor and Laptop is powered on.

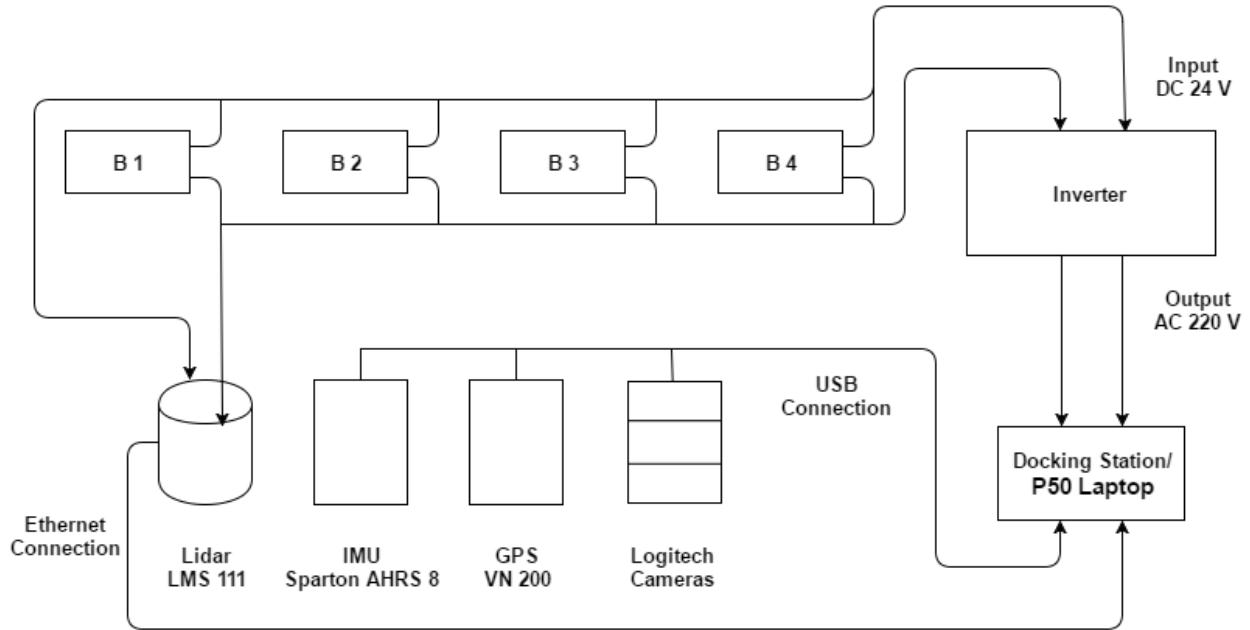


Figure 4: CPU Power Circuit

4.4 Specifications

Table 3: Specification of Electrical Components in SeDriCa

S No	Item	Specifications
	Battery	1. Driving Battery - 13000 mAh/ 24 V Li-Po 2. Component battery – 48000mAh/24 V Li-ion
1	Laptop	Lenovo P50, Intel Xeon Processor, NVIDIA Quadro GPU, 8 GB RAM, 170 W
2	Motor Driver	Roboteq HDC2450, 2X150 A channel, USB enabled
3	Motor	Maxon – 200 W, Max Speed – 86 RPM
4	IMU	VN – 200, Angular resolution – 0.05°
5	GPS	Hemisphere AtlasLINK, Accuracy < 0.6 m
6	LIDAR	LMS111, Range - 20m, 25 Hz, 0.25°
7	Cameras	Logitech C920, horizontal FOV - 70°

5. Description - Software Strategy and Mapping Technique

5.1 Lane Detection

The Image Processing module of the vehicle is designed to detect lanes and flags accurately and position them accurately on a local occupancy grid. This occupancy grid is later sent to the mapping module which fuses the grids of Image Processing Module, LIDAR and sends the fused map to the path planning module where the lane information and the flag information is used.

5.1.1 Lane Detection Algorithm:

The Lane Detection Algorithm is based on color filtering where we search for white colored pixels (RGB values more than 200 in an 8-bit encoded image). Usually when we apply such filters on the image, we do get the lanes accurately, but along with the lane we also get some oddly illuminated pixels due to sunlight

or some reflected light falling on the camera lens etc. These regions on the image are very irregularly shaped contours, much like the sunlight patches on the image shown.

5.1.2 Using Lane Properties to remove irregularities:

We need to remove the noise coming in the image and use the properties of lane to detect them. This is done by a simple observation. The lanes detected by the camera are generally quite straight and long. This means that if we bound the contour with a tight fitting rectangle, the length of the rectangle is at least 5-6 times in magnitude when compared to its breadth. This would reject a lot of contours and give us only straight lanes.

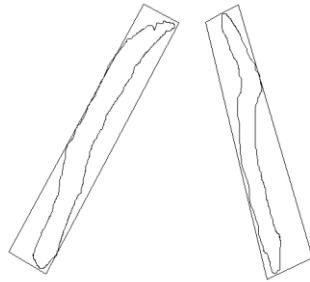


Figure 5: Result after Length to Breadth Thresholding of Lane Data

We check the bounding rectangle area and compare it with the contour area. In most cases, the bounding rectangle is quite large when compared to the lane contour when the lane curves. This means that the curvature of the lane forces the rectangle to have comparable dimensions. This would mean that the actual contour area is quite small compared to the bounding rectangle area. This means that the lane is curving and thus these lanes are also detected.

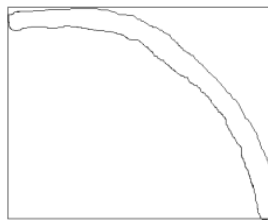


Figure 6: Exception Handling in Length to Breadth Thresholding

5.1.3 Inverse Perspective Mapping:

Every camera has a particular field of view horizontally and vertically. Based on the camera position, every camera has a particular region in which objects are visible. If we attach the camera at a height and incline it to avoid the vanishing line, we get the following mapping of the pixels to the ground coordinates.

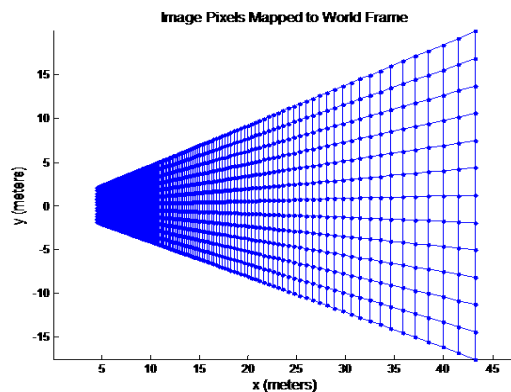


Figure 7: Inverse Perspective Transform (IPT)

The image we see via a camera is a rectangular image which gets mapped to the ground coordinates in this fashion. So to find the image mapping on the ground, we do the following:

- Map the 4 corners of the camera image to the ground
- Construct the entire image on the ground by linear interpolation from those 4 points

This method of reconstructing the image has been tested to give very accurate mapping of the camera data to the real world.

We have applied this concept to all the three cameras that are attached on the vehicle (Left, Middle and Right). Then we have combined the occupancy grid obtained from all the three cameras into a single grid that provides the information on the lanes in front of the vehicle. A sample output is shown in the first figure.

Now, the lanes extend behind the car as well, but the field of view of the camera is limited. Thus as seen in the first figure, there is an empty space in front of the car which has no lane. Thus to fill that, we extend the lanes that are detected by the camera till the last row in the occupancy grid. This is done by extending a straight line from the lowermost part of a lane and the nearest point which is at least 1m away from the lower most point. This distance criterion is chosen so that the lanes do not cross each other. The final output of the lane detection module is this interpolated grid map as shown.

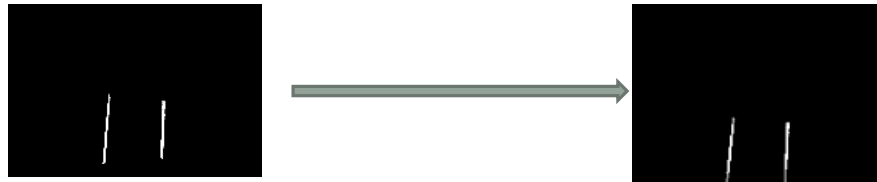


Figure 8: Interpolation of Lane Data

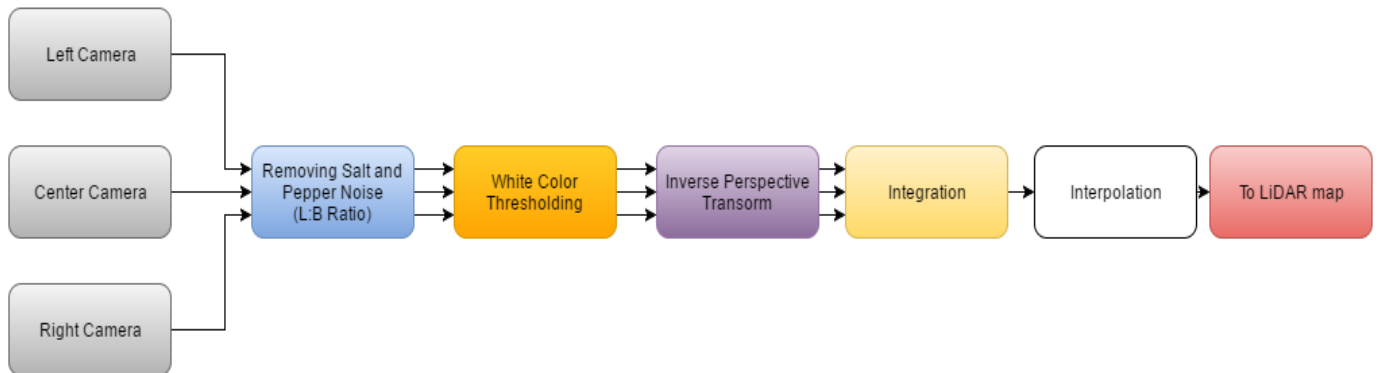


Fig 9: Flowchart for Image Processing Module

5.2 Sensor Map

The basic mapping module completely depends on the data provided by the sensors. A node in ROS first converts the LIDAR data into 400x800 occupancy grid with the car in the middle. This accounts for the obstacle detection and mapping part of the basic mapping module. Two nodes from IP give the corresponding line and flag occupancy grid. The mapping module saves 5 history of the lane map and employs a probabilistic method to help predict if the surrounding environment do consist of lanes(This has been done to take into account the conditions when the IP module may/may not wrongfully defects a lane). Finally all the three occupancy grids - obstacle map(shown in white), lane map(shown in red) and flag map(not shown but green in color) are fused together in one-one correspondence on the final 400x400 Gridmap.(fusion has been done keeping in mind the exact position of LIDAR and cameras and appropriate transformations gave been made)

Pros:

- a) Fast (map updates at 15 Hz)
- b) All possible data from the sensors are there on the final map.
- c) Map can be easily fused with sensors from various sensors. Hence, highly scalable.

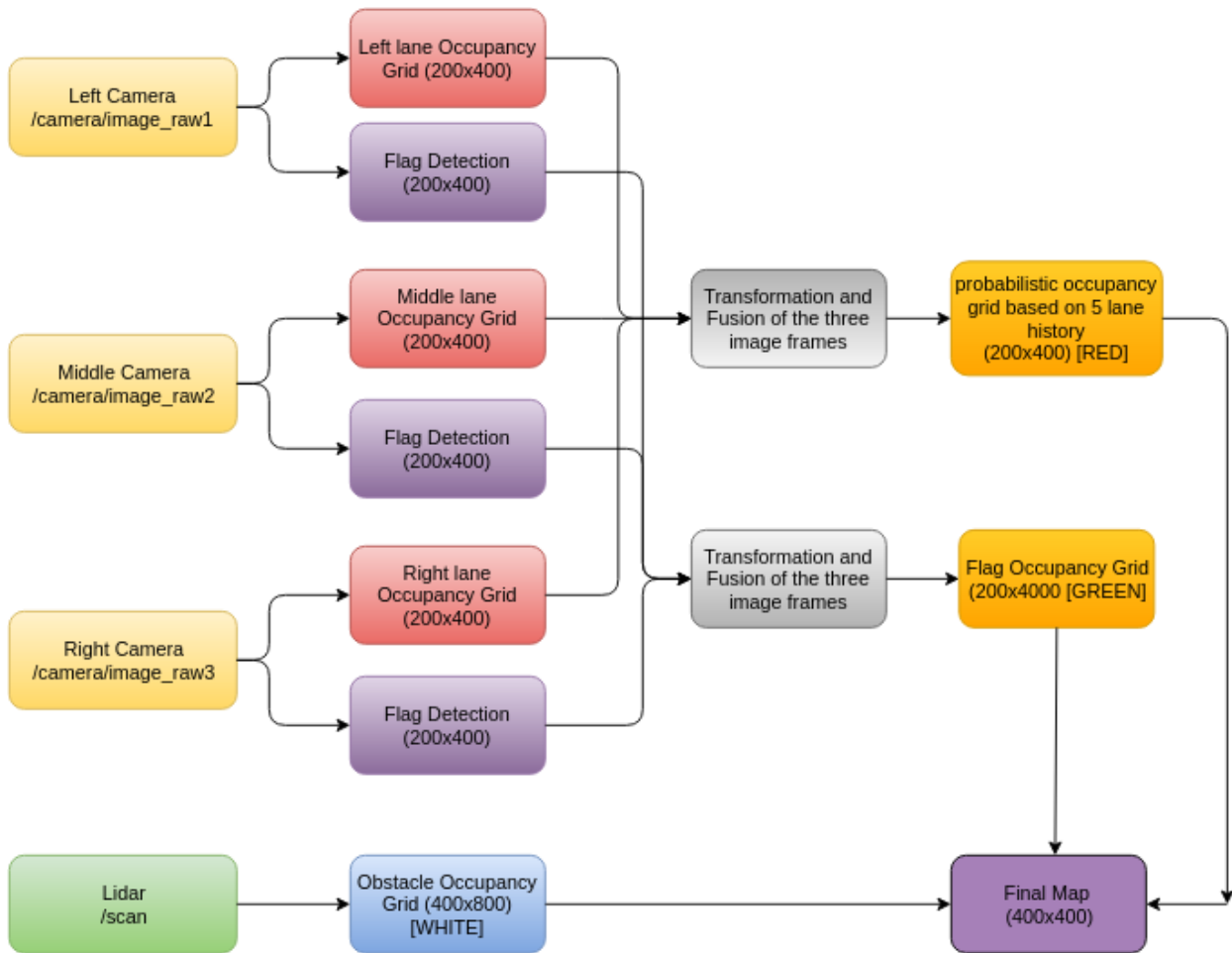


Fig 10: Flowchart for Map Fusion Module

5.3 MAP Fusion

Individual occupancy grid maps generated above are combined together and are stitched to together with a global occupancy grid map (generated in an incremental fashion using previous local grid maps).

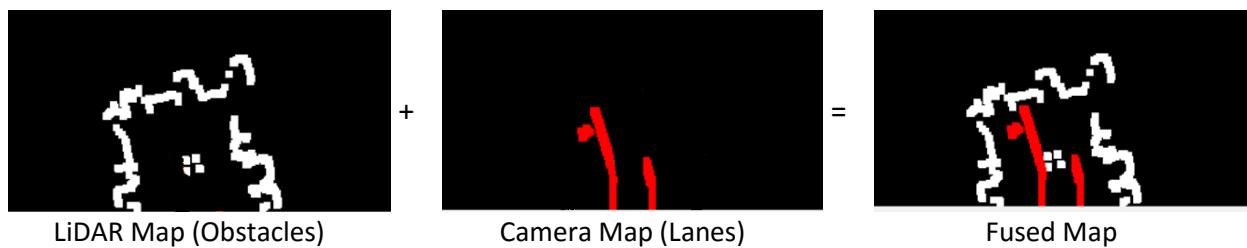


Figure 11: Result after Map Fusion (Lanes + Obstacles)

5.4 Localization

The environment in IGVC is sparse, thus traditional SLAM algorithms such as G-mapping were found to be unreliable. In order to increase the accuracy of our positioning, we use a Kalman Filter to integrate data from the GPS system with data from an INS (Inertial Navigation System) and encoders. Positioning even if no satellite is visible was also possible using this approach.

Kalman Filter is an extremely effective and versatile procedure for combining noisy sensor outputs to estimate the state of a system with uncertain dynamics. In GPS/INS/Encoders integration case, noisy sensors include GPS receivers INS and Encoders components, and the system state include the position, velocity, acceleration, attitude, and attitude rate of our vehicle. Uncertain dynamics include unpredictable disturbances of the vehicle and unpredictable changes in the sensor parameters.

The relatively low data output rate of GPS receivers does not meet the cm level accuracy requirements for robot localization in a global plan. This problem becomes more serious when the potential temporarily loss of a GPS signal occurs or phase ambiguity resulting from cycle slips considered. INS provides the dynamics of motion between GPS epochs at high temporal resolution and complements the discrete nature of GPS in the occurrence of cycle slips or signal loss.

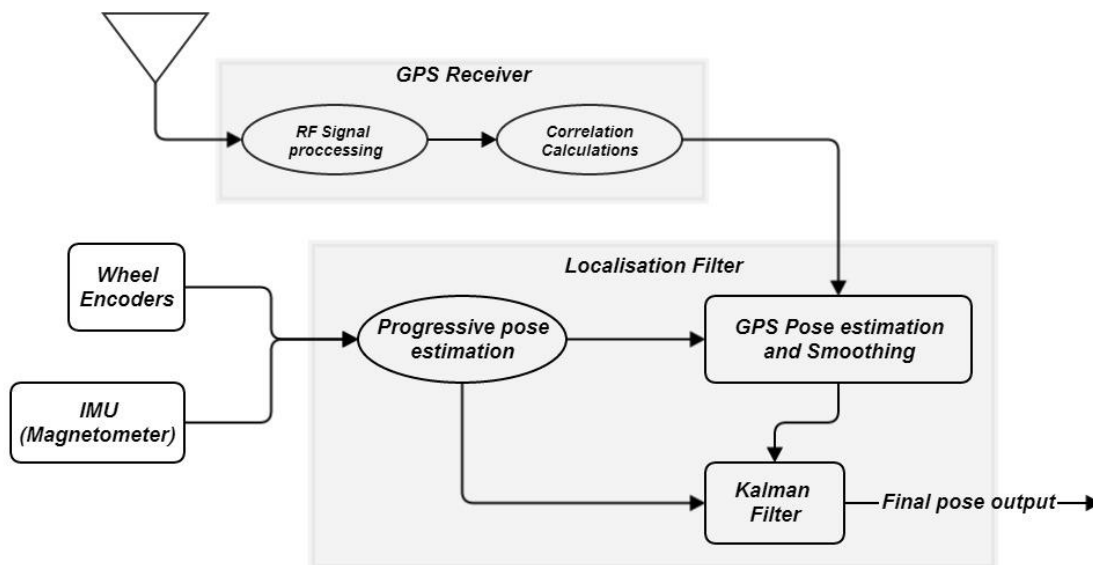


Figure 12: Localization Module

5.5 Path planning

The path planning module of the vehicle has the task to generate an optimum path to reach the next goal (GPS point) while avoiding obstacles and keeping the vehicle between white lanes.

For path planning a variant of **D* algorithm** (dynamic A*) is used which demands a grid map to solve and start and end cells in the grid.

The module requires the following inputs:

- Environment map :
The local map which has the obstacle and lane information and is continuously updated while the vehicle is moving.
- Information of next GPS point :
The heading to the next GPS point is passed on to the module which enables it to select a goal point in the local map.

Another variable is used in the module which specifies the distance to be planned in the local map.

For example - distance to plan = 5 meters and heading to next point = 35 degrees due north.

There are **three heuristics** used for doing A* searches namely:

- **Goal cost:** This cost increases radially away from the goal point.
- **Nearest obstacle distance cost:** The cost creates Voronoi diagram and thus helps the path finder keep maximum distance from all obstacles in the local map.
- **Obstacle boundary cost:** This one is high near the obstacles and dies out as we go away a few centimeters and ensures the path to be always a little away from the obstacle.

For creating the Voronoi cost an OpenCV function distTransform is used. Below is a sample obstacle map of an area with multiple rooms.

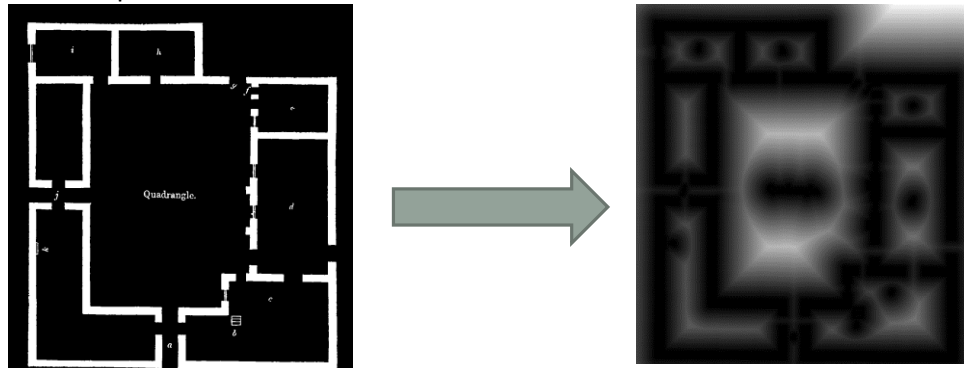


Figure 13: Result after Distance Transform; White Region indicates low cost

Output after applying distTransform function gives this image and the pixel values are used to assign cost to a particular cell.

More the pixel value less the cost assigned. Thus the obstacles are dark and very high cost.

Following figures shows the A* search on the map with all the three mentioned costs. Red region is the explored one and green pixels belong to the final path that the algorithm has found.

Starting from a point in room i to a point in room d.

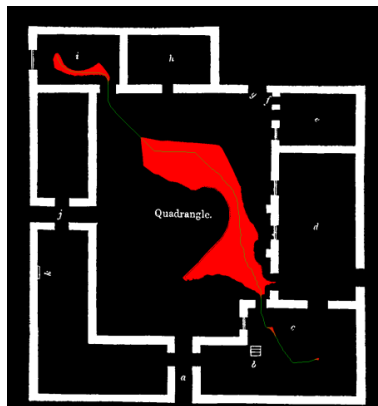


Figure 14: Explored area while finding path

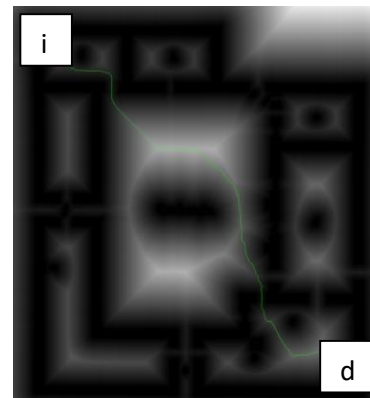


Figure 15: Final path from Start to End Point

5.6 Motion Planning

After the path cells are found, an Interpolator fits a smooth curve on the path and passes the final path information to the motion planning module.

The motion planning module requires the following inputs:

- The path cells' location in global frame
- The current position of the vehicle
- The distance to the next GPS point
- The obstacle map

The motion planning module has two tasks:

- Output a desired heading due north in which the vehicle should move currently.

- The velocity of the vehicle.

The motion planner calculates the current desired heading by simple coordinate geometry and a velocity planner is assigned to the task of deciding the maximum velocity in the current state.

The inputs to velocity planner are:

- Obstacle density in the local area
- Distance to the next GPS point

The obstacle density is simply found by the percentage of the map that is filled or occupied and the distance to next GPS point is provided by the global planner which has the current location and all the GPS points in sequence.

The velocity planner assigns a lower velocity if the obstacle density is high or the vehicle is near the next GPS point and keeps it high in open areas and in between GPS points.

Final part of the planner is the command module -

It takes the following inputs:

- Current heading (yaw) of the vehicle through the onboard IMU.
- Desired heading and velocity from the velocity planner.
- Maximum angular velocity that can be given as command to the vehicle's motor driver

The command module uses a ROS Roboteq interface which allows the onboard computer to connect to the ROBOTEQ controller in the vehicle.

The command module has a PID controller programmed to attain the desired heading as quickly and smoothly as possible.

6. Description of failure modes, points and resolutions

Table 4: Analysis of Failure Modes and methods of resolutions

Failure mode	Resolution/tests/remedies
Mechanical failure: Damage to motors' gear box due to prolonged overloads Loosening of nut/bolts	Flange coupling is used to prevent damage to motor & its gearbox. The vehicle is tested many times in a similar scenario as the competition. If properly done it can withstand the loads. Solidworks stress analysis is done on the vehicle's links and there is a huge safety factor.
Electrical failure: Motor controller malfunction Damage to electronics Damage to the cables of sensors Damage to the wiring between the motors, the controller and the batteries	Conduits and glands are used to secure cables Safety enclosures are used with IP-65 and above ratings to withstand the conditions during test runs Suspensions and dampers are used to minimize vibrations in the vehicle
Software failure: Runtime errors Special blockade cases Emergency scenario	The thoroughly designed test cases ensure proper vehicle operation. Emergency stop in software is present The vehicle automatically stops if any of the required sensor stops giving data.
Severe vehicle failure	Emergency stop button on the vehicle which can also be remotely accessed.

7. Simulations

Simulation is a very helpful and reliable tool in a manufacturing/development process, as it provides a low-cost, secure and fast analysis tool. It enables us to model a real system for the purpose of understanding the behavior of the system so as to take decisions quickly and reliably.

Gazebo, an open-source platform is used for simulations. Gazebo is used in unison with ROS (Robot Operating System). For this Hokuyo vehicle (which is a differential vehicle) is used in gazebo with all the essential sensors onboard. All the links, joints and sensors on the vehicle can be controlled. ROS topics enables the different sensors to communicate with each other.

Plugins have been written to collect the data published by different sensors, so that they can further be used for testing localization, path-planning, mapping, obstacle avoidance and other algorithms we are employing.

Though Gazebo simulates everything in an ideal virtual world but noise is added to the sensors using Gauss-Markov error model to incorporate the fluctuations observed in real world. Also, the Hokuyo vehicle is modeled in a way to take into account the weight, inertia, etc. of the vehicle. A virtual world is developed which looks close to the race course, where the vehicle will be going to run in the competition.

7.1 Sensor Simulation

The following sensors have been simulated in Gazebo and currently being used by other modules.

7.1.1 LIDAR

LIDAR is one of the most reliable sensor used to detect and track objects with high accuracy. LIDAR attached on the vehicle has a field of view of 270° and range of 20m. LIDAR tracks the obstacles in its sector region and returns the distance of the obstacle from it. Figure 16(a) shows that the blue region is unhindered while there are some obstacles which are tracked by the LIDAR. The output data is sent to a ROS topic which is then visualized on RVIZ (a topic visualization software in ROS) as shown in Figure 16(b).
(a) LIDAR in Gazebo (b) RVIZ visualization of LIDAR

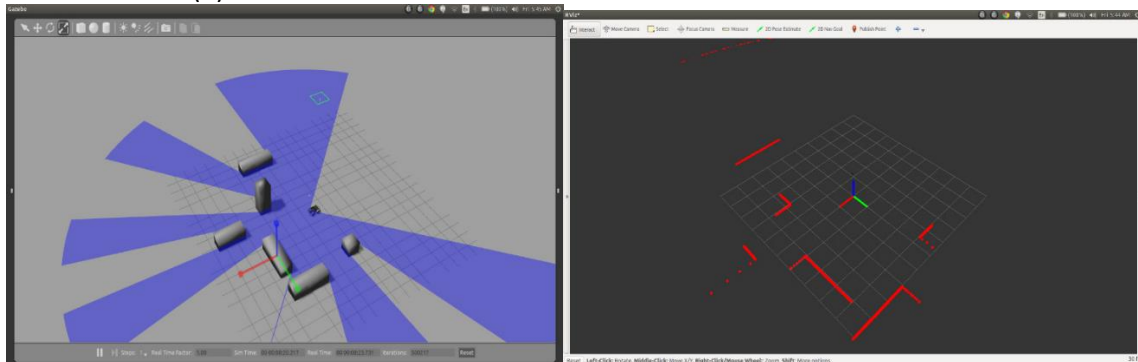


Figure 16: Simulation of (a) LIDAR in Gazebo, (b) Rviz visualization of LIDAR

7.1.2 Monocular Camera

A monocular camera is an essential sensor for detecting and classifying lanes, vehicles, pedestrians and other obstacles. 3 monocular cameras are mounted on the vehicle which publishes real time image data on a ROS topic.

The output image is then subscribed to a code which fuses the image and detects the white lanes which are marked RED in Figure 14. Output of only one of the cameras is shown.

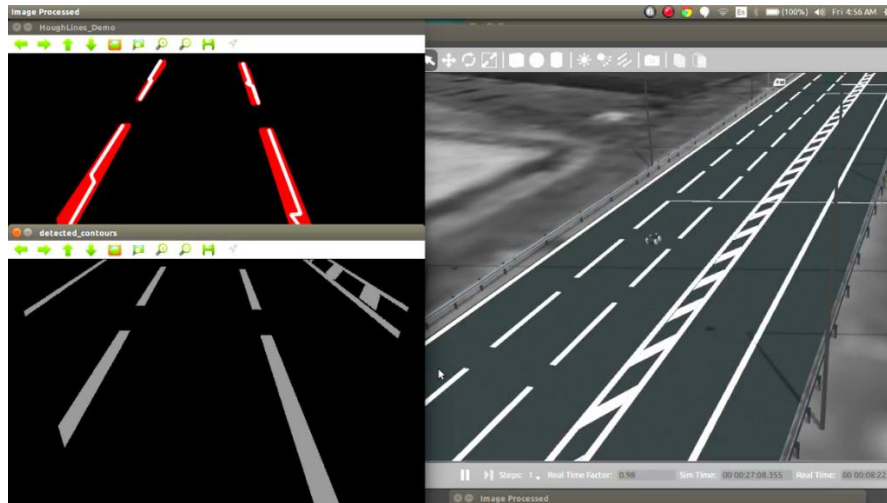


Figure 17: Simulation of Lanes on Road and Lane Detection Algorithm

7.1.3 Inertial Measurement Unit (IMU)

The inertial measurement unit (IMU) is the most important sensor for measuring the angular velocity, acceleration and heading of the vehicle with respect to a reference axis (global north). This information is later on required for localization. We have implemented a virtual IMU on the vehicle which gives its heading angle w.r.t. north.

7.1.4 GPS and wheel encoders

Besides these, we also have a GPS mounted on the Hokuyo vehicle. Wheel encoders are mounted on all the four wheels.

8. Performance Testing

Table 5: Testing Parameters and Result for SeDriCa

System	Test
Mechanical	Vehicle is tested on rough terrain
Obstacle and Lane Following	Stays 25 cm away from objects and lanes at all times
Localization	Accuracy ~25 cm with IMU+GPS+ Encoders
Mapping Accuracy	Local Map accuracy based on Localization (~25 cm)

9. Initial Performance Assessments

This section discusses the performance of SeDriCa so far.

Table 6: Performance parameters of SeDriCa

Parameter	Performance prediction	Performance result
Speed	1.5 m/s (3.5 mph)	1.3 m/s (2.9 mph)
Ramp climbing ability	15°	>15°
Reaction time	0.1 sec	0.2 sec
Total time of power supply	2 hrs. (driving)	3-4 hrs. (driving)
Obstacle detection distance	0.5 m - 20 m	1 m - 20 m
Waypoint navigation	0.6 m	1 m

10. Expenditures

10.1 Monetary Expenditure

Table 7. Cost of Various Components

Item	Capital Cost (USD) ^a	Team Cost (USD) ^b
Maxon Brushed DC Geared Motors	2384	2384
Roboteq HDC2450 Motor Controller	645	0
SICK LMS 111 Laser Range Finder	4300	0
Sparton AHRS-8 inertial sensor	1200	0
Atlas Link GPS Antenna Hemisphere	4500	3095
Vectornav VN-200 INS	3200	0
Logitech Webcams C 680 & C920	340	340
Spektrum Dx6i 2.4GHz trans-receiver	300	0
ThinkPad P50 laptop with Slice Battery	2800	2800
Analog I/O module	900	900
FTDI-2232H USB-Serial Modules	35	0
Starcom PCI to dual ethernet express card	200	0
Li Polymer Batteries	300	300
Li Ion Batteries	330	330
Invertor (DC to AC)	150	150
Raw Material for Chassis & Fabrication	2000	1000
Wheels & Castor	100	75
Miscellaneous Electrical Hardware	300	300
Total	23984	11674

1. a: it is the market price of the equipment.
2. b: it is the actual amount paid by team for acquiring new equipment. For old equipment, this cost is zero.
3. Fields marked in green have been acquired from the market

10.2 Labor Expenditure

The team is preparing for IGVC-16 for past 8 months. On an average, each person has given around 2-3 hours/day to the project. So, the total labor expenditure in the project is around 3000 hours

11. Conclusion

SeDriCa has proved to be quite rugged in myriad of situations and is quite effective in localization, maneuvering and path planning capabilities. The current system is thoroughly tested with over 10 km of testing. The team is now planning to shift to 3-D LIDAR for competition pertaining from 2017. The research has been started on Stereovision and Machine Learning Algorithms. They are going to be part of subsequent competitions. The team wants to improve upon its predecessor robot which participated in IGVC 2013.

12. Acknowledgement

The team would like to acknowledge IIT Bombay to give us the wonderful chance to participate in IGVC – 2016. Our faculty advisors, Prof S N Merchant and Prof Ramesh K Singh, have been always supportive for our project. We would also like to thank our alumni, Mayank Porwal, Kumar Keshav, Pranav Maheshwari and Ravi Yadav for their never ending support to us. We would also like to thank the student community of IIT Bombay who have always motivated us.