Indian Institute of Technology Bombay

Design Report

15th May, 2017

SeDriCa

Ankit Sharma	ankit.s@iitb.ac.in
Rishabh Choudhary	rishabh.choudhary@iitb.ac.in
Ravi Jain	ravi_jain@iitb.ac.in
Anjan Kumar Patel	anjanpatel79@iitb.ac.in
Rohit Bhor	r.bhor@iitb.ac.in
Vatsal Kansara	vats.kansara@iitb.ac.in
Krishna Sandeep	krishna09@iitb.ac.in
Surya Teja	surya.teja@iitb.ac.in

I hereby certify that the design and development of the vehicle SeDriCa, described in this report is significant and equivalent to what might be awarded credit in a senior design course. This is prepared by the student team under my guidance.

Merchant

Prof. S. N. Merchant
Department of Electrical Engineering
Email : merchant@ee.iitb.ac.in

Contents

1 Introduction

Team from the Indian Institute of Technology (IIT) Bombay has designed and greatly improved the vehicle 'SeDriCa' from last year to compete in the 25th Intelligent Ground Vehicle Competition (IGVC). SeDriCa continues the tradition of smart, autonomous, differentially-steered vehicles using advanced sensors and actuators running with integrated software. The team consists of students of various engineering disciplines, many of whom have worked on Mahindra Rise Driverless Car Challenge (aims to build India's own Driverless technology), ASME Student Design Challenges and various other national and international robotics competitions. The vehicle consolidates features of mechanical, electrical, and software subsystems that accentuate simplicity and safety.

2 Organization

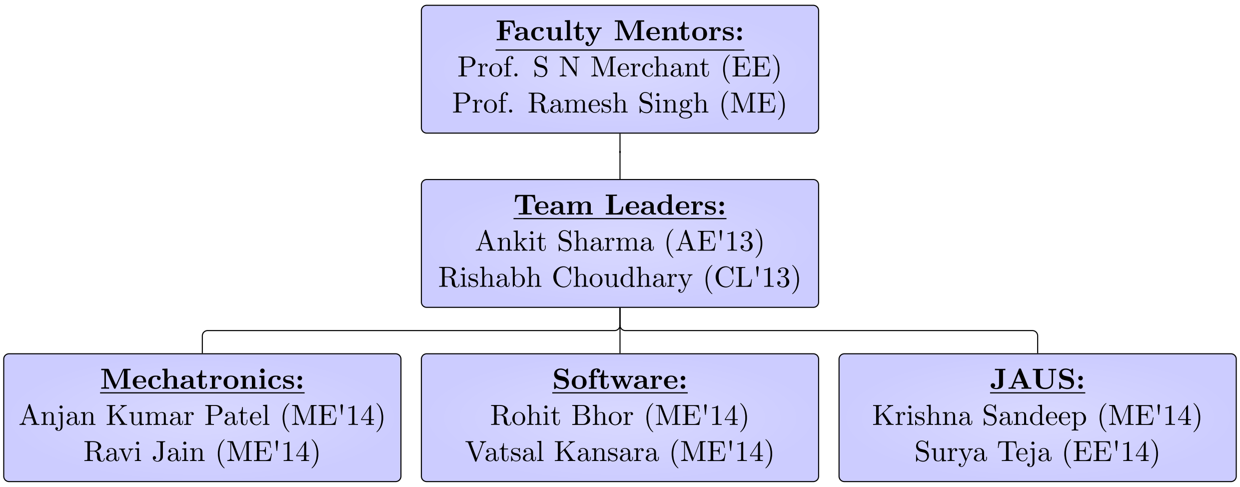

Figure 1: Team Structure

The team consists of eight members. Team leaders are responsible for organizing team meetings, funds, public relation and also maintaining open communication between the sub-systems. The mechanical team focuses on designing the chassis, mounting the sensors, and building various mechanical components. The electrical team focuses on designing circuit boards, planning and maintaining proper distribution of the power to the various sensors, actuators, and other electrical components, and wiring the robot efficiently. The software team works on vehicle localization, detection of obstacles and lanes, mapping, path planning and enabling IOP.

3 Cost Estimate

Table 1: Cost Analysis of SeDriCa

SeDriCa's component	**Retail cost (in USD)**	**Team cost (in USD)**
Ampflow A28-400 Motors	1200	1200
Sica Aluminum profile	800	0
Asus GR8 II mini-PC	1350	1350
Spartan AHRS-8	1350	0
SJCAM SJ4000 Camera	200	200
Sick LMS 111 LiDAR	4300	0
Roboteq HDC 2450	700	350
Atlaslink GNSS GPS	4500	3095
US Digital S1 encoder	240	240
Li-ion 6S Battery Pack	1000	1000
Tactics Transmitter	250	250
Manufacturing and Other	1800	1800
Total	17690	9485

In addition to the cost of parts, each of the students have spent approximately 14 hours per week on the IGVC project. Thus amounting to approximately 600 hours each during the academic year 2016-17.

4 Design Process

The team has used a six step strategy while designing the vehicle. The six steps, illustrated in Figure 2, are: analysis of the problem statement, planning, designing, building & integrating, testing, and then evaluating the vehicle performance. The first step in our design process was to identify the problem. The team analyzed its previous IGVC performance and tried to eradicate the problems faced last year, thereby enhancing the vehicle's performance. Further, the next step in our design process was planning. It involved brainstorming on various tasks and deciding whether the team should pursue a solution or not. The necessary step of building and integrating involved implementation of our final design ideas on the vehicle and ensuring that the closed loop system is operational (based on analysis and simulation reports) and integrating all the modules. In the testing phase, we rigorously tested all our modules to ensure the accuracy and efficiency on track, similar to the actual competition.
In the end, the team evaluated the performance of the vehicle. If the test results were not optimal, we cycled back to the problem analysis and repeated the steps. Major factors which contributed to the fast convergence of the design process were extracting maximum features from the sensor data and also working on the cross sensor coordination. This process increased the command accuracy and boosted our trust in the module.

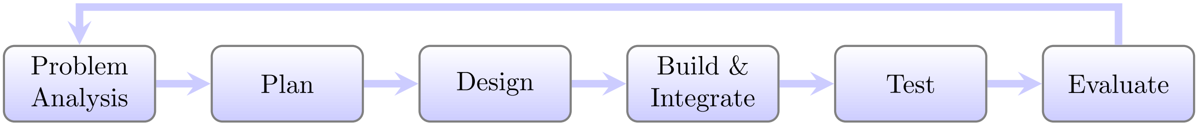

Figure 2: SeDriCa's Design Process

5 Innovation

5.1 Mechanical Innovations

- **Modularity and Compactness**: The vehicle has a close packed design and it also can be reassembled from scratch in less than an hour. All important parts are easily accessible as the vehicle has surface openings. An integrated suspension system with front caster wheel makes it space efficient.
- **Adjustable Sensor Mounts**: Custom mounts were manufactured such that degrees of freedom can be specified as per requirement.
- **Optimized Vehicle Weight**: Weight reduction from 120kg to 63kg is achieved by a full redesign of the vehicle, use of 20 mm sica Aluminum profiles and replacement of Aluminum from no load section with delrin polymer.
- **Stability**: Intelligent distribution of weight to achieve very low center of mass coincident with center of vehicle.
- **Encoder Covers**: Smart encoder coverings were manufactured in order to eliminate vibrations and asymmetry between motor and encoder axis.
- **Three Stage Gear Box**: A low cost chain sprocket gearbox consisting of three gear stages has been used to incorporate very low (0.2°) angular play in the system.

5.2 Electrical Innovation

- **Electromagnetic Wire Shielding**: Encoder wires are now protected from the random noise. Metal covering has been used to shield the cable from external magnetic field of the motors.

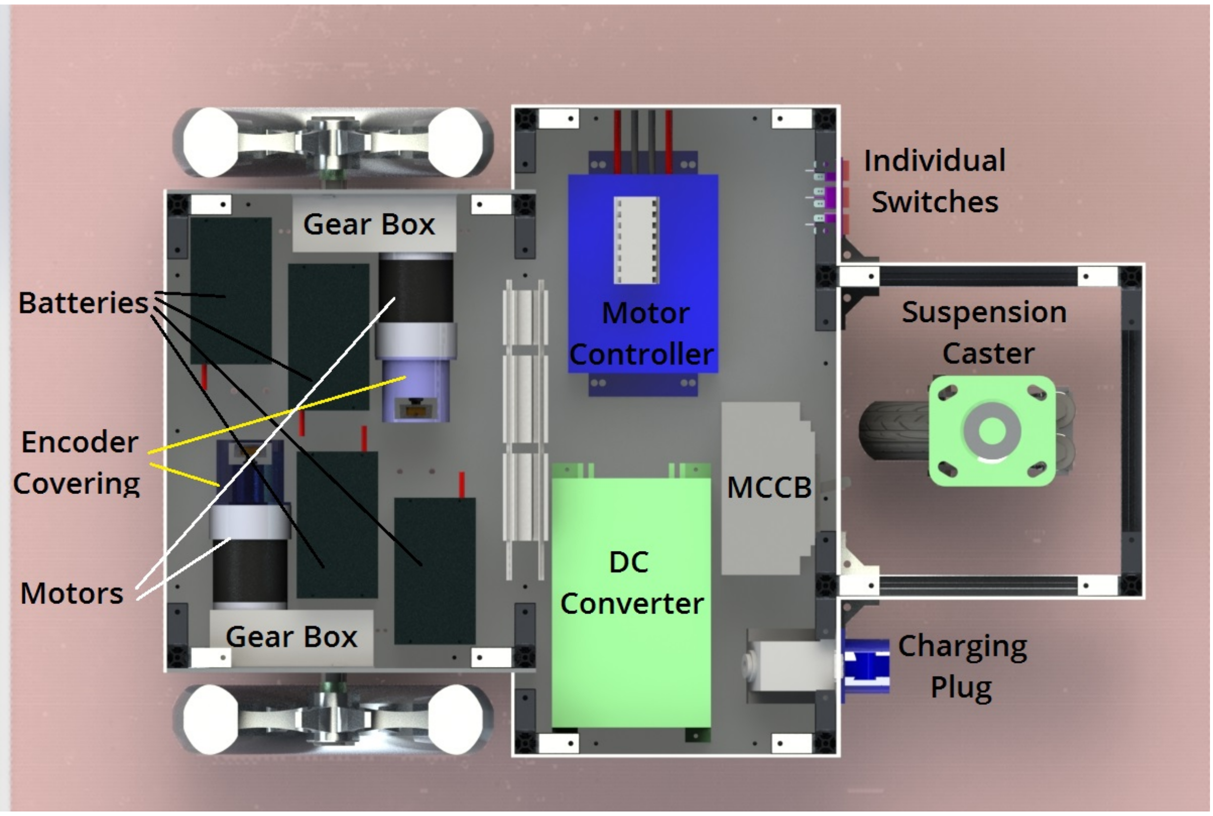

Figure 3: SeDriCa's Top View

- **Hard Cased Battery Packs with Real Time Voltage Monitoring & On-board Charging**: To ensure safety of the vehicle, Li-ion batteries are hard cased with temperature sensor inside. A voltage sensor and charging plug are placed outside to get real time battery voltage output and to charge the batteries without disturbing any other module.
- **Mini PC**: A mini PC, instead of a laptop, with on-board power supply has been used to do all the processing. This system takes less space on top of vehicle and also eliminates wastage of power on laptop screen.
- **MCCB**: A DC MCCB has also been connected to the battery to immediately shut down the system in case of short circuit or high current supply. It also works as the main switch of the vehicle.

5.3 Software Innovations

- **Color Channel Extraction**: Single color channel has been used in the vision module to detect and mask the colored barrels such that they do not appear while adaptive thresholding.
- **High Speed Motion Planning**: Developed a velocity function according to obstacle conditions and path status with a maximum velocity of 1.6m/s
- **Integrated ROS with JAUS**: : Instead of using ROS and JAUS implementation separately, we have bridged the two platforms with the help of shared memory. This has reduced extra hardware requirement and helped in efficient utilization of the system.

6 Mechanical Design

6.1 Overview

SeDriCa's mechanical design focuses on the tactical, lightweight and compact size of the vehicle. The structure of the vehicle is composed of sica 20 mm sections with aluminum plates for support and delrin plates for covering. These materials are chosen to make the vehicle lightweight, strong and easy to assemble and disassemble. Overall dimension is just 0.02 feet more than the specified minimum dimension of 2 feet x 3 feet to make the vehicle easier to move and also have a small turning radius. The front corners of the vehicle have been trimmed to provide clearance for the obstacles and thus avoid near collision. The center of mass has been kept low with maximum ground clearance possible. The three wheel design consisting of two driving wheels and a caster wheel enables quick and sharp turns.

Figure 4(a): SeDriCa's Final CAD Model

Figure 4(b): SeDriCa's Skeleton View

6.2 Vehicle Chassis

SeDriCa's chassis consists of aluminum 20mm sections with two major parts and a three wheel stable system with maximum parts close to the ground to lower the center of mass. The rear chassis is optimally designed with two drive motors along with their gearboxes, sensor mounts and four rechargeable batteries which have been precisely fit into a compact volume. The middle part of the chassis provides a larger enclosed space housing all the electronic components which includes a battery charging system that just requires an external cable to plug in without disturbing any part of the vehicle.

6.3 Body

Most of the body of SeDriCa consists of Aluminum 6061-t6 which is a light weight material having more strength than most of the materials with same weight. Delrin is used for covering the side parts. The upper part of SeDriCa includes an adjustable camera mounting system which enables the camera base to be situated at different heights and also the two cameras position with variable angles overlap which makes testing easy, also providing various configurations which can be explored. The camera base is made by laser cutting with proper markings to allow for easy reorientation of the camera. The front caster wheel consists of a suspension attached to it which enables easy maneuver on small bumps.

6.4 Sensor frame

The compact placement of batteries and motors brings the encoders very close to the wires with high currents whose electromagnetic fields affect the data transmission from them. A metallic encoder cover was specifically built to restrict the interference between the power and data cables of the driving motors.

6.5 Weather Proofing

SeDriCa is designed to withstand light precipitation during the run. Each part of the vehicle has been properly covered with Aluminum and delrin plates to ensure no water can leak into the body. The cameras are enclosed in water resistant covers and other sensors like GPS, IMU and LIDAR are suitably water proof already by build.

7 Electronic and Power Design

7.1 Overview

The electrical system of SeDriCa has been improved from the last year's vehicle. The control system consists of Roboteq HDC2450 motor controller, RC transmitter, and an ASUS mini-PC. Various safety features are also an important part of the new system. This design focuses on safety, reduction in power consumption and fully customizable dynamic control. The complete electrical wire routing is shown in figure 5.

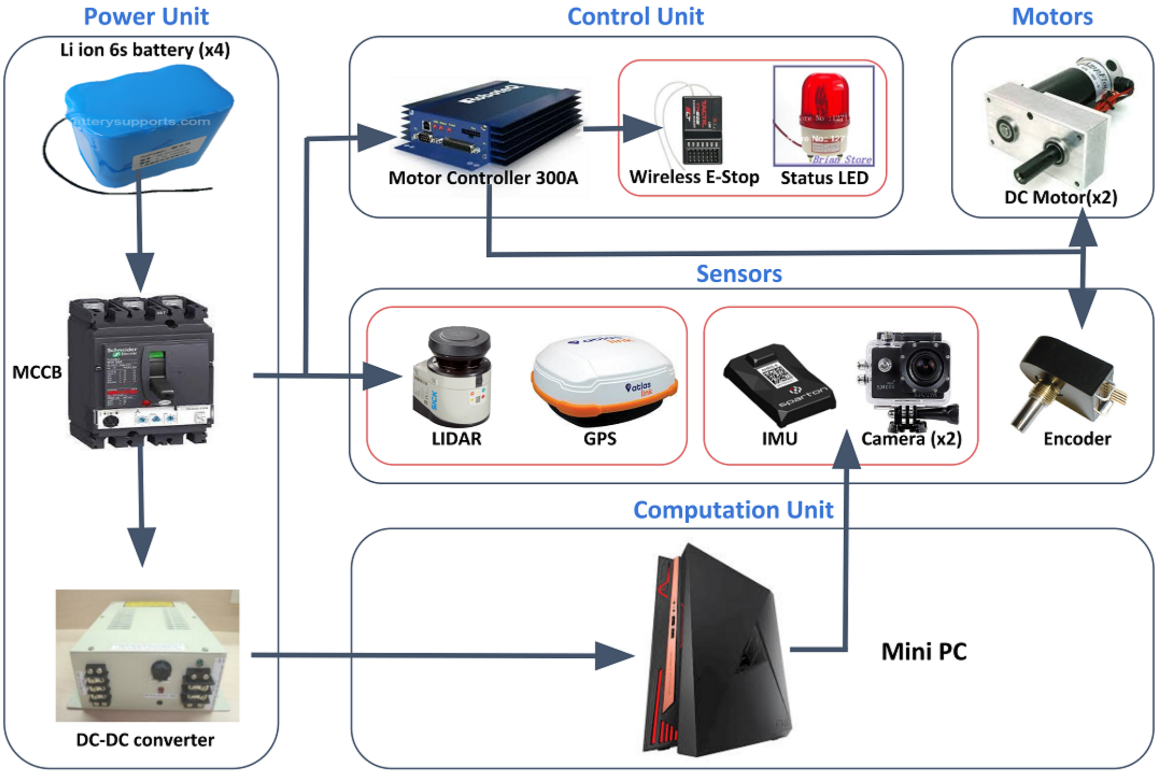

Figure 5: SeDriCa's Wire Routing

7.2 Power Distribution

All components in the vehicle are powered by a single source: four 6S(six cells) Lithium ion batteries connected in parallel to produce 24 VDC. Each of the battery has a capacity of 12000 mAh, hence all four together have a capacity of 48000 mAh.

Table 2: Power requirement of the electrical components

Electrical component	**Max power consumption**	**Operating voltage**	**Source**
Atlaslink GPS smart antenna	4.5 W	24 VDC	Li-ion battery pack
Sparton AHRS 8	0.875 W	5 VDC	Laptop via USB
SJCAM SJ40000 cameras	10 W	10 VDC	Laptop via USB
Ampflow A28-400 motors	750 W	24 VDC	Roboteq HDC
ASUS GR8 II mini-PC	230 W	19.5 VDC	DC-DC converter
Sick LMS111 LIDAR	8 W	24 VDC	Li-ion battery pack

As Figure 5 shows, Li-ion battery pack supplies power to the motors through the Roboteq motor controller and also supplies the 24 VDC to GPS, LIDAR and DC-to-DC converter that converts it to 19.5V DC for charging the laptop. Having a single source for everything, coupled with an easy “plug and charge” system makes charging SeDriCa simple and convenient. It takes around 4-5hr to fully charge the batteries from low voltage levels (~ 22V). Total power consumption in worst case scenario is 1003.375 W. To make sure battery voltage does not go below 22 V, a digital voltage display is placed for real time voltage monitoring.

$$\begin{aligned}\text{Maximum power consumption} &= 1003.375W\\ \text{Minimum time available} &= \frac{\text{Battery Capacity}}{\text{Total max power consumption}}\\ &= \frac{1209.6}{1003.375}\\ &= 1.21hr\end{aligned}$$

Generally, operating power consumption is less than half of the max power consumption. Hence, the vehicle can run up to 2.5 hr with all electrical components and sensors working at the same time. This year, SeDriCa’s battery backup has been increased as compared to the last year due to the complete overhaul of the control system and efficient buck-boost DC-DC converter instead of last year's inverter for charging the laptop. A new power distribution unit for each sensor and controller with an integrated switch panel also aids in power savings. Complete electrical wire connections were done using XT90 and phoenix UK-10 connectors suitable for handling the maximum possible currents in the connections.

7.3 Electronics Suite Description

As seen from the figure 4, the mini-PC is used for computing the sensor data from cameras, LIDAR, IMU, GPS and encoders and a motor controller is used for driving two high power motors in a closed loop. A 2.4 GHz receiver-transmitter is used for manual vehicle control as well as a wireless emergency stop. A 24V DC status LED light is mounted in the vehicle to differentiate it in manual and autonomous mode. A 2D obstacle map formed with LIDAR was fused with lane data from cameras to give final occupancy grid for path planning.

Table 3: Electrical Component’s Specifications

Components	**Specification**
Computational unit	ASUS ROG GR8 II mini-PC, 16GB, i7−7700, Nvidia GTX 1060, 3 GB
Motor Drivers	Roboteq HDC2450, 2X150 Amp channel, 60 VDC, USB enabled
Encoder	US digital S1 optical, Incremental, 1024 CPR
Spartan AHRS 8	Data sampling filtered to 2 kHz, Baud rate : UART, USB enabled
GPS	Hemisphere Atlaslink, Accuracy < 0.6 m,
LIDAR	Sick LMS111, Range - 20m, 25 Hz, Ethernet Enabled
Camera	SJCAM SJ4000, 170 degree diagonal FOV, fish eye lens, USB enabled

7.4 Electronic Safety

Modular case circuit breaker (MCCB) has been used to prevent very high currents (more than 250 Amp) for short circuit protection and it also acts as the main switch for the system. Small display unit is placed for real-time monitoring of the batteries'voltage. The system has two separate E-stop switches: one implemented through hardware and the other through software. The hardware one is a red colored push switch which is the primary safety mechanism for SeDriCa. The software safety system is through the remote controller without cutting the motor power. A transmitter (wireless E-stop) sends stop commands up to a range of 1km and if the Roboteq controller fails to receive a valid signal within 10 milliseconds, it will stop the vehicle instantly. The indicator light signifies the state of the vehicle: blinking light for the autonomous mode and a solid light for the manual mode. Batteries were packed in Aluminum cases to protect the system from fire/explosion. High ampere rated connectors (XT-90) were used at battery terminals and other electrical components'connections.

8 Software Strategy : Navigation

The software has been developed in C++ and python and integrated using Robotics Operating System (ROS). It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and much more which had helped us greatly in creating all the modules. The logic flow designed is optimal for the hardware and gives good refresh rate from the final decision making module.

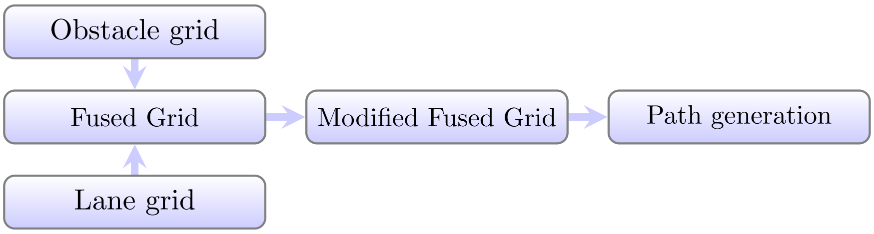

Figure 6: Software Integration Flow Chart

8.1 Vision

8.1.1 Overview

The vision module has been developed such that it is unaffected by aberrations in weather conditions and gives accurate detection of lanes and potholes in the arena. Two wide angle action cameras 120° are used with a 40° field overlap for capturing image frames with effectively 200° field of view and are processed using opencv libraries.

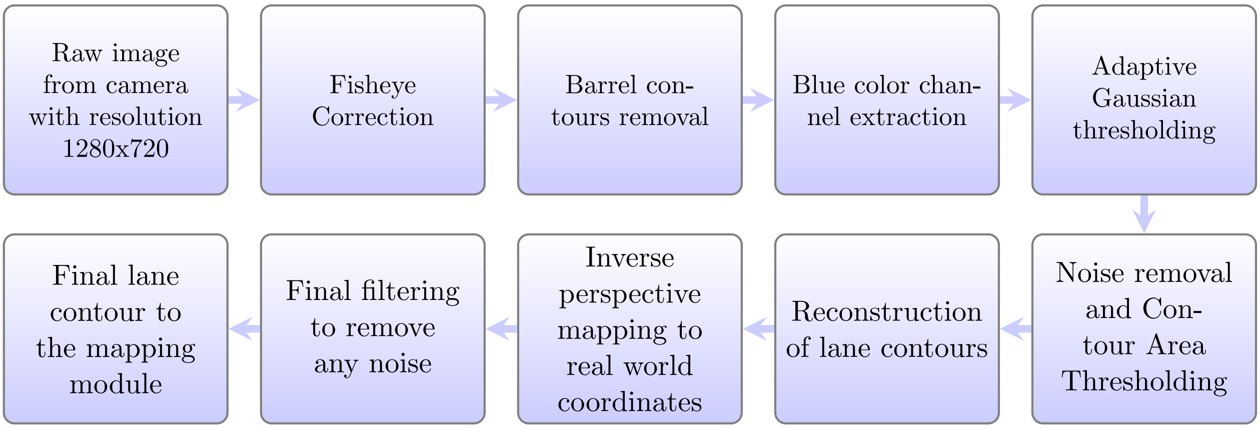

Figure 7: Vision Module Flowchart

8.1.2 Software Strategy

The current algorithm for lane detection uses Adaptive thresholding at its core. We perform the following steps on both camera inputs to successfully detect lanes which are of white color and are continuous/dashed. A detailed description of the algorithm is given below:

- As the cameras used have a wide angle lens, the raw image needs to be processed to remove curvature effects. From Fig 8. the curvature of ground can be noticed which is dominant at the edges. Fig 9. below shows the image after fish-eye correction.

Figure 8: Raw Image

Figure 9: Image after Fish-eye Correction

- Next, we separate the resulting image into its 3 constituent color channels blue, green and red, further processing only blue the channel. The reason of working on single channel is that it is easier to process single channel image as compared to 3 channel one, this reduces computation time. Besides this, it also helps in removing unwanted noise resulting from sharp shadows.
- Then we apply Adaptive thresholding which highlights pixels of high intensities in the given kernel size. Image filtering is the next step which is achieved by picking up the median intensity. This filtering technique is very popular for removing salt and pepper noise.
- After this, we get an image in which the lanes are highlighted with presence of some noise (which was not removed in median filtering due to its size or white color). Then contours are detected and filtered by thresholding their area such that only lane contours are left.

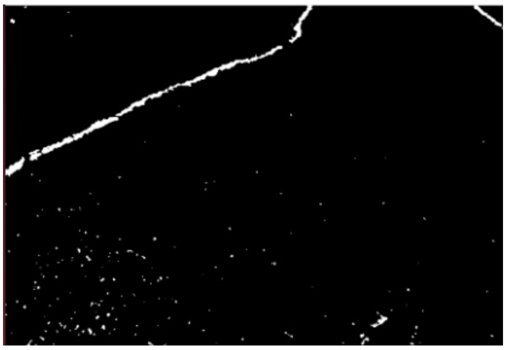

Figure 10: Adaptive Gaussian Thresholding Output

Figure 11: Image after Median Filtering

- Above steps will detect the lanes in the image as seen in the camera view. To get its top view (which is required for path planning) we need to apply the following algorithm:
    - Inverse Perspective Mapping: The image we see via a camera is a rectangular image which we need to map to the ground coordinates. It is necessary to incline the camera in such a way that the horizon is not visible in its field of view. To find the image projection on the ground, we do the following:
      i) Map the 4 corners of the camera image to the ground co-ordinates
      ii) Construct the entire image on the occupancy grid by linear interpolation from 4 points

8.2 Localization

Localization of the system is handled by fusing the data from different on-board sensors namely: GPS(Hemisphere AtlasLINK), IMU and feedback from wheel encoders. Due to inefficiency in implementation of SLAM algorithms in given environment because of high computational power and sparsity, localization is handled independently from mapping without taking Lidar data into EKF module. Following the EKF process, the localization module takes sensor data to estimate the current robot state and performs prediction and correction steps based on the earlier and the latest robot states.

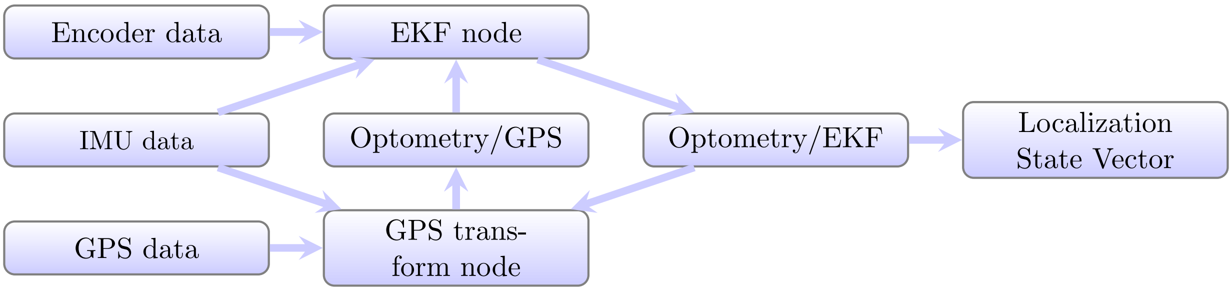

Figure 12: Localization Process

8.3 Obstacle Detection and Mapping

8.3.1 Overview

The system uses Sick LMS111 LiDAR to detect obstacles. The laser sensor has a 270 degree field of view and an operating range of 20 meters. We only consider the obstacles that are inside a 10 meter radius around the vehicle. The obstacles are then mapped into a 2D occupancy grid according to their relative positions with respect to the vehicle. Another node detects the lane information and forwards a grid containing lane data. These both obstacle and lane grids are fused together taking into account the relative positions of the LiDAR and the cameras. The occupied cells in the grid(i.e., with lane/obstacles) are magnified so as to ensure smooth motion planning and also to maintain an appropriate gap between lane/obstacles and the vehicle. The map also differentiates between lane and obstacle data which would help the path planner in more efficient planning.

8.3.2 Software Strategy

ICP algorithm: We mapped the course using a SLAM technique called Iterative Closed Point (ICP).

- The ICP algorithm, which is one of the scan-matching methods, calculates the closest position iteratively by adjusting the motion vector and rotation matrix of the model.
- The algorithm iteratively estimates the relative pose of the robot between two consecutive scans and maps them to the global reference frame as a global map.

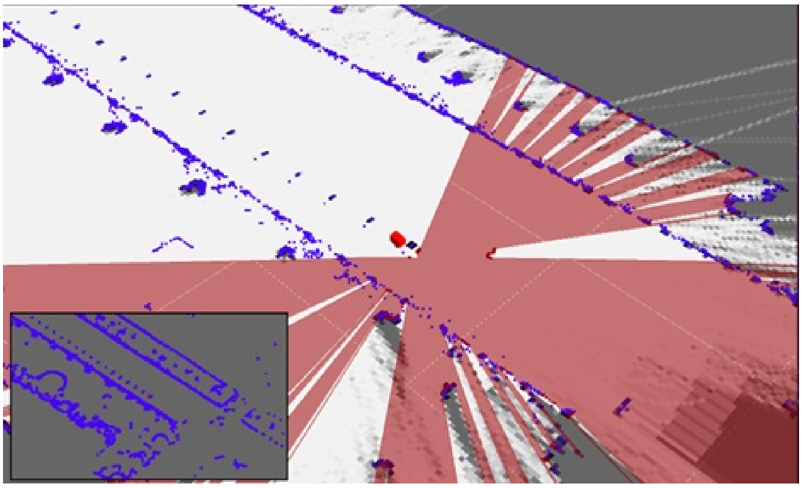

Figure 13: Map Generation using ICP Algorithm

Global Map:

- But, the SLAM techniques work effectively only if the environment is dense which is not the case for the IGVC course. So an alternative method with the localization data estimated from an Extended Kalman Filter (EKF) algorithm has been used.
- We map the entire course by stitching the local map data using the EKF estimated localization data and form a global map which can be used in further runs.
- Using a global map (i.e., the map of the whole course) would be very useful in planning as we would not need to adjust the path frequently and thus also plan for longer distances.

8.4 Navigation

8.4.1 Map Acquisition and Processing

Modified lanes and obstacles data on a 2D grid along with way-point data (distance and direction to goal) are subscribed by the planning node. The center of the vehicle from which planning takes place can be set according to the track conditions. Keeping the center at the back helps in better planning but keeping it in the front helps in achieving higher velocity. Thus for running at lower velocity, the center is kept at 3/5th of the length of the vehicle from front. The width of the lanes and magnification of the obstacles can be increased so that the vehicle can safely maneuver but it may increase the possibility of a 'no-path found' scenario. Lower values may in-turn lead to collision and crossing over between the lanes. The map data received can be modified dynamically according to the velocity and acceleration of the vehicle using the above constraints. Running the vehicle at high velocity requires higher magnification of obstacles but not much dependence on lanes. The map comprises of 180 thousand pixels which accurately estimates the environment.

Figure 14: Navigation and Planning Flowchart

8.4.2 Path Generation

The navigation system allows the vehicle to obtain information about its current position in the local and global environments. Atlas Link GNSS Smart Antenna and Sparton AHRS-8 IMU have been used to give the vehicle the ability to locate way-points which are then used to calculate a desired angle of heading for travel. This desired heading will then give the vehicle a goal orientation, which will be the direction of the shortest distance to that way-point. Then the goal is set such that the path is achievable. If in some scenario, no path can be found, the heading is changed to last achievable heading to ensure that the vehicle does not get stuck. This is done by keeping a history of achievable local locations on the map. If the goal is found to be on an obstacle, it is moved away from the vehicle until it is in the clear zone.

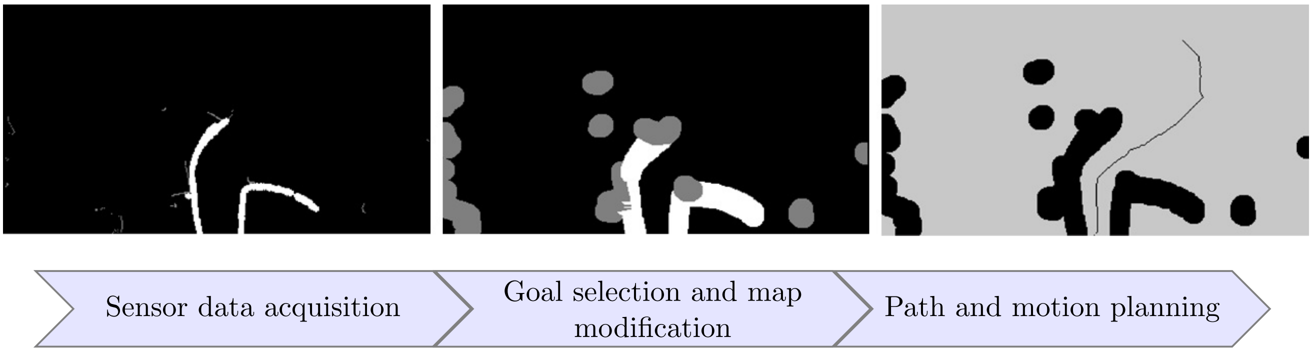

Figure 15: Path Planning Flow

After this, a heuristic cost map is created with different cost functions which includes an exponential function emerging from the obstacles which diminishes as one goes further from them, the distance transform which essentially creates a cost gradient that helps in keeping the vehicle away from all the surrounding obstacles, a cost which increases as the lane end approaches and a goal cost. The weight given to these costs can be adjusted to keep the vehicle at a distance from lanes or from obstacles. The first parameter for lanes of both sides and obstacles helps in keeping away the vehicle from them. Changing the second parameter to lower the value helps in smoothening the lane entry. The goal cost and local goal data is used to find the safest and shortest path by using D* algorithm. The calculated path is straightened and smoothened such that the speed of vehicle can be high and the path does not overlap with the obstacles. These parameters are refined to the best smoothened and straightened path. There are two parameters for smoothening, alpha and beta, which control ability to "change" & "remain as it is" in path. Straightening is also controlled by a set of parameters. These codes run on different processor threads to achieve better processing speed.

8.4.3 Motion Planning

The motion planning node subscribes the planned path and the integrated map. The map is used to estimate the maximum velocity and angular velocity of the vehicle in that region. This is achieved by three factors : Distance to nearest obstacle, difference between current and desired heading and distance of obstacle just in front of the vehicle. Less difference between heading and desired heading ensures higher average velocity. If the obstacle is in the critical radius of the vehicle then the velocity is exponentially reduced. The velocity and angular difference is sent to control module which uses a PID algorithm with constraints according to the velocity of the vehicle to maintain it on the planned path. The PID values are dynamically adjusted using the obstacle density and distance from nearest obstacle. The saturation value of angular velocity is set such that the vehicle does not misbehave in unexpected conditions or if the goal is behind the vehicle. When the vehicle reaches the GPS location, the velocity is decreased until it is inside an achievement radius. If next GPS location is there, the goal is set according to that GPS way-point. This continues until there are no GPS locations left to achieve.

9 Software Strategy : Interoperability Profiles

9.0.1 Overview

Interoperability is a characteristic of a product or system, whose interfaces are completely understood, to work with other products or systems, in the present or the future, in either implementation or access, without any restrictions. Interoperability Profiles are used to facilitate interoperability between controllers, robotic platforms, and payloads. This challenge demands its implementation following Joint Architecture for Unmanned Systems (JAUS) by SAE. We used the JAUS Tool-set in order to build a JAUS compliant software. The following system topology is required -

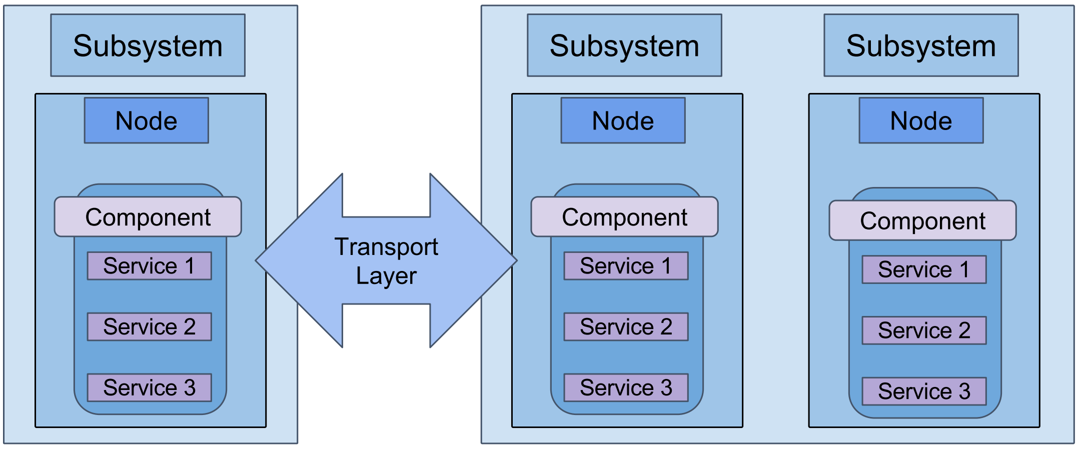

Figure 16: System Overview Showing Subsystem-node-component Topologies

9.0.2 JAUS Tool-set (JTS)

JTS, a free software, provides a very powerful and user friendly environment to develop JAUS related softwares. It comes with a Graphical User Interface (GUI), which makes it a lot easier for the developer to understand and implement the required system topology and modify it as needed. JTS is capable of producing source code in C++ as well as in Java. The services are created by defining the messages and protocol behaviors, and then finally integrating them altogether. Later, service sets are created which are finally used to build the required components.

9.0.3 JTS - ROS Integration:

We have used Robotic Operating System (ROS) as the platform for implementing autonomous capabilities in our vehicle. To integrate JAUS Standards in our modules we have combined JTS and ROS. This has been done internally through data transfer over shared memory. Hence, alongside adhering to JAUS's set of standards, we have developed an interconnected system, making use of the hardware as efficiently as possible.

10 Failure Modes, Failure Points and Resolutions

Failure modes/points	Resolution/remedies
Mechanical failure:	
Damage to the chain-sprocket of the gearbox due to prolonged overload or sudden high impact	Spare chain-sprockets are available
Loosening of nuts and bolts with time	Lock nuts are used at all places
Decrement in spring constant of the front suspension	Extra springs are there for replacement
Water seepage in the vehicle body	The inside of the volume has an extra layer of polythene for prevention of water seepage in the electrical system
Electrical failure:	
Motor controller malfunction	Mechanical or wireless E-stop safety switch stops the controller immediately on push and toggle respectively. Spare motor controllers and sensors are available
Damage to electronics, cables of sensors, wiring between the motors, the controller and the batteries etc.	Conduits and glands, used to secure cables can be easily removed to debug and locate faults, MCCB is used for short circuit protection as well as a high ampere switch
Overheating may lead to high temperatures inside the machine	Fast removal of outer body covers is possible and the sensitive components will be unharmed.
Software failure:	
GPS, IMU, LIDAR and other components not working	Try reinserting connector cables, run through the source workspace and give administrative permission to the USB ports
The vehicle is not reaching the exact GPS location or misbehaves while in autonomous mode	Make sure every sensor and publisher is working and also checking nothing is in front of the vehicle

11 Simulations

Gazebo, an open-source platform has been used for the simulations. Gazebo has been used in unison with ROS (Robot Operating System). All the links, joints and sensors on the vehicle can be defined.

The sensor data is published on different ROS topics which can be used to test localization, path-planning, mapping, obstacle avoidance and other algorithms we are employing. To aid in getting a more realistic view of the course, a 3D environment was also created that accurately depicts many of the obstacle types at the competition, along with a true model of the vehicle. This allows us to have an easy understanding of what the vehicle is seeing when it encounters a situation, which facilitates easier debugging. As Gazebo simulates everything in an ideal virtual world, noise has been added to the sensors using Gauss-Markov error model to incorporate the fluctuations observed in the real world.

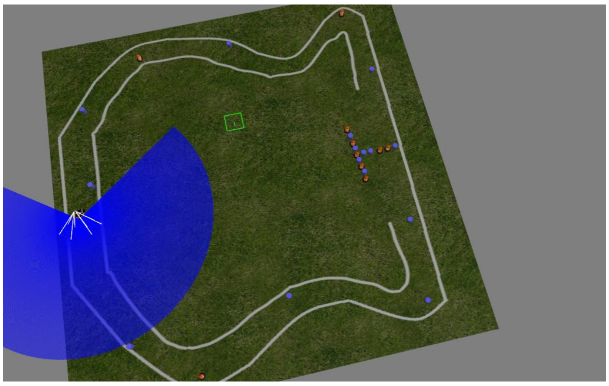

Figure 17: Auto-Nav simulation in Gazebo

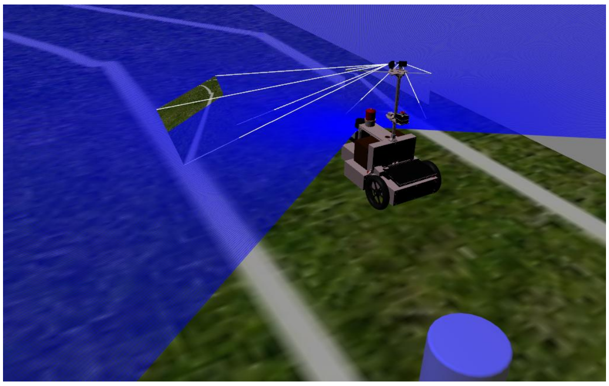

Figure 18: Lane and obstacle detection simulation

12 Testing

SeDriCa was tested in an open environment resembling the IGVC competition layout. The white lines were made by sprinkling POP powder on the grass. Orange and blue colored barrels and traffic cones as obstacles were placed in the arena to test the vehicle in a pretentious competition environment.

Figure 19: SeDriCa testing on a small track similar to the actual competition one

Mechanical: SeDriCa was first designed from scratch using the software package SolidWorks 2016 which was followed by analyzing the stress and strains at critical points using the software package ANSYS Workbench 14.5 version. After many iterations the final vehicle has been fabricated using appropriate low weight and high strength materials at the right place to ensure structural integrity. This assured that the frame wouldn't fall apart after being assembled and loaded. It was tested in rough terrain, wet surfaces and an angle of inclination up to 25° using a ramp to ensure the robustness.
Electrical: The electrical system has also been tested in our lab. Each of the sensors were checked in the lab before going for testing outside to ensure proper power and data connections thus preventing severe failure.
Software: Each software module from lane detection to obstacle avoidance was tested, initially using bag-files for tuning the values of variables. In motion planning, different PID values were tested for different maximum velocity runs and then the best one was chosen for our final run.

13 Initial Performance Assessments

SeDriCa has been built to meet the requirements strictly as per the IGVC rules. The operational performances of the general vehicle has been listed below:

- Operational speed of 2.5 mph (5 mph max) during runtime.
- The estimated battery life is about 1.2 hour if used continuously but it ran up to 2.5 hours during testing.
- The vehicle has successfully climbed and descended more than 15° incline ramp without any issue of tipping or bouncing off which was more than our expectations.
- All obstacles can be detected from 0.1m to 20m at 1mm steps and during testing all visual objects were detected up to 5m away at a refresh rate of 20Hz.
- The vehicle hits the way-point between 0.1m to 0.75m until moving on, when approaching a way-point.
- The vehicle handles various complex obstacles easily e.g. potholes by fitting a circle using Hough transform, dead ends and traps by an empirical turn clockwise and explore protocol.
- The actual test results show significant match with the predicted ones which displays the precision with which everything is put together.

SeDriCa was able to perform way-point navigation, lane following, obstacle avoidance, lane detection, mapping and IOP with efficiency and repeatability.