Indian Institute of Technology Kharagpur

Design Report

15th May, 2018

Eklavya-6.0

G. Rahul Krantikiran	krantikiran.68@gmail.com
Indu kant Deo	06274indukantdeo@gmail.com
Het shah	hetshah036@gmail.com
Aditya Rathore	arathore98@gmail.com
Harsh Maheshwari	harshmaheswari135@gmail.com
Sanskar Agrawal	sanskaragrawal107@gmail.com
Sohan Rudra	sohanrudra@gmail.com
Arijit Kar	arijit.kar14@gmail.com
Siddhant Haldar	siddhanthaldar14@gmail.com
Rohit Ner	rohitner1@gmail.com
Poojan Shah	poojanshah284@gmail.com
Shubham Mawa	shubhammawa1998@gmail.com
Ashwin Nehete	ashwinnehete@gmail.com
Rishik Keshari	rishik.kgp@gmail.com

SUPERVISING FACULTY STATEMENT

This is to certify that the engineering design present in this vehicle is significant and equivalent to the work that would satisfy the requirements of the senior design or graduate project course. Eklavya 6.0 has been designed with an objective to optimize performance in all areas and has a completely new mechanical design than its predecessors and significant revamp in electrical power distribution, efficient control, system architecture and intelligent navigation. I wish the team success for IGVC 2018.

15/5/2018

Professor Debashish Chakravarty,
Dept. Of Mining Engineering,
IIT Kharagpur

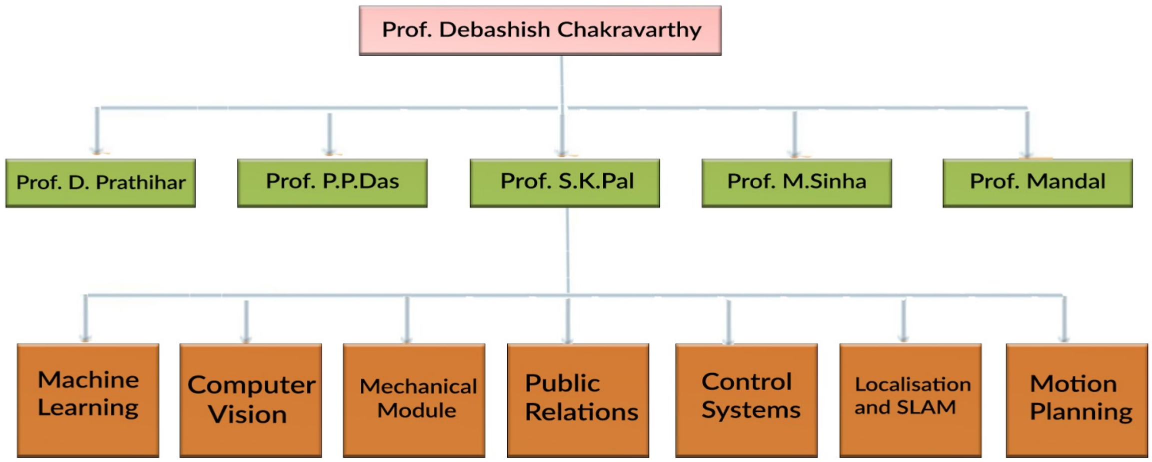
EKLAVYA 6.0 - TECHNICAL REPORT
AGV - Indian Institute of Technology - Kharagpur

Introduction

Team Autonomous Ground Vehicle (AGV), under the ambit of center for excellence in Robotics, IIT Kharagpur, has been pioneering the autonomous ground vehicle technology with the ultimate aim of developing India's first self-driving car. The team has been participating in IGVC since 2011 with the Eklavya series of vehicles. Eklavya 6.0, another feather in the cap of the research group is all set to participate in the 26th Intelligent Ground Vehicle Competition (IGVC), Oakland University. With new robotic innovations, this successor of Eklavya series is much more efficient in all aspects i.e mechanical, electrical and software.

TEAM ORGANIZATION

The effort behind this project was put in by a bunch of over fifty enthusiastic and intellectual undergraduate students from various departments of IIT Kharagpur.



Team Organization

MECHANICAL DESIGN

OVERVIEW

There were certain key ideas and objectives while designing Eklavya 6.0. The previous version of the bot had multiple scopes of improvement and at the same time making necessary changes in the existing chassis was not feasible. So a completely new chassis design was drafted this year. The drive was changed from front wheel steering mechanism to two wheel differential drive with a caster. Mounts were manufactured for all sensors. The chassis design was done keeping in mind housing for payload, batteries, and other electrical equipments.

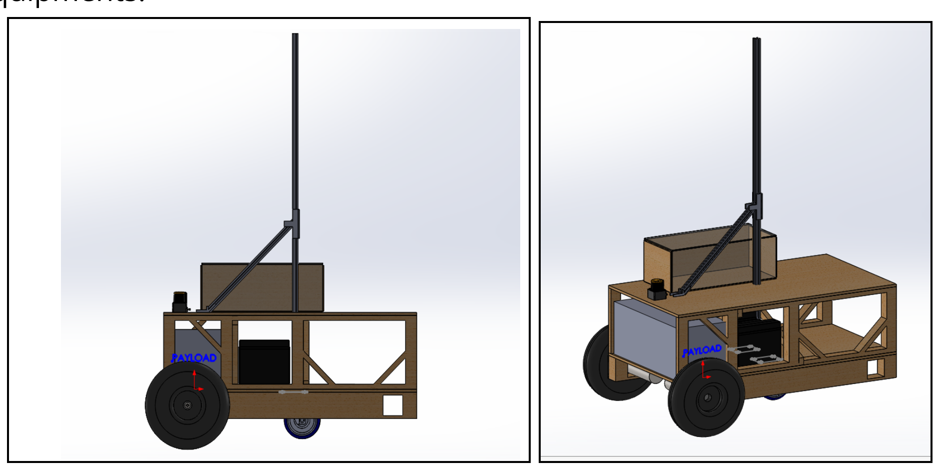


Fig.1. CAD Model

Innovations and Advantages :

1. Wooden Chassis :

The primary motive of choosing teakwood as the robot frame material was to ensure the ease of manufacturing. Wood is also advantageous with its vibrational damping properties over steel and aluminium alloy which were intentionally avoided due to machining difficulties and their rigid structure which introduce vibrations in the chassis.

2. Reduction of vibrations in the vehicle and the sensor mounts

The truss structure of the chassis is best suited to handle any kinds of stresses and shearing experienced by the vehicle in different directions. Triangular members have been introduced in the chassis considering the fact that triangular structure provide maximum stability. Horizontal beams stabilize the base to a large extent and prevent bending of the structure.

The reduction of vibration was a major component necessary for optimal performance of all the sensors. This was implemented by using the two crossbars which are alternatively in tension and compression restricting the lateral vibrations. Ansys simulations showed that the strain caused due to random vibration was reduced to a great extent (as shown in fig.2). The aluminium vertical shaft has been supported by an inclined member which absorbs the vibration.

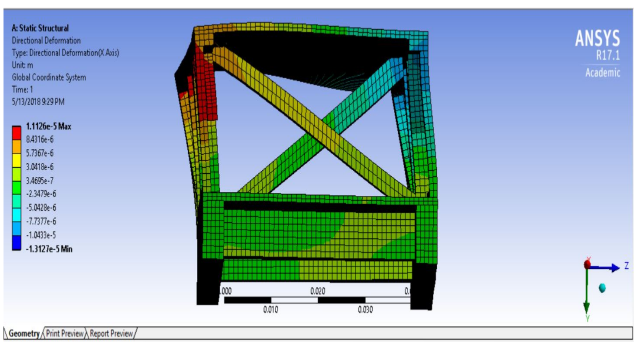


Fig.2.(i) With cross-member in position

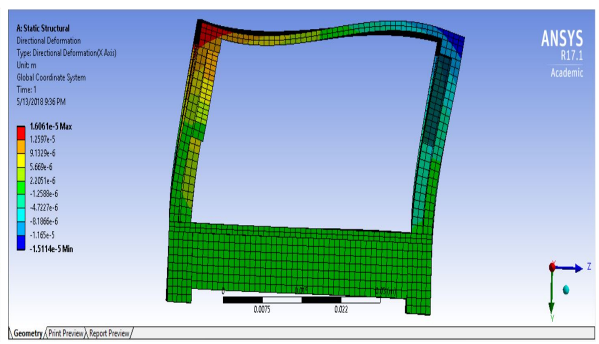


Fig.2.(ii) Without cross-member in position

3. Wheel Drive and Maneuverability:

We now moved on from steered drive method to two wheel differential drive with a caster. In the steered drive mechanism used in Eklavya 5.0, the BLDC motors were inefficient and prone to failures. To resolve that difficulty we are now using DC motors with the following specifications:

DC MOTOR SPECIFICATIONS:

1) Rated Torque : 14.2 Nm at 175 rpm
2) Power : 100 W
3) Rated Voltage : 24 V
4) Weight of Individual Motor : 1.45 kg

The movement of the vehicle is more precisely done by the two driven pneumatic wheels and the caster. Use of two wheel differential drive allows the bot to achieve zero radius turn about the axis consisting of the powered wheels. The caster easily aligns itself according to the turning conditions. The centre of mass of the bot has been kept significantly low towards the axis of the driven wheels to ensure zero radius turn about the centre of mass of the bot. The centre of mass of the bot has been kept significantly low towards the axis of the driven wheels to ensure zero radius turn about the centre of mass of the bot.

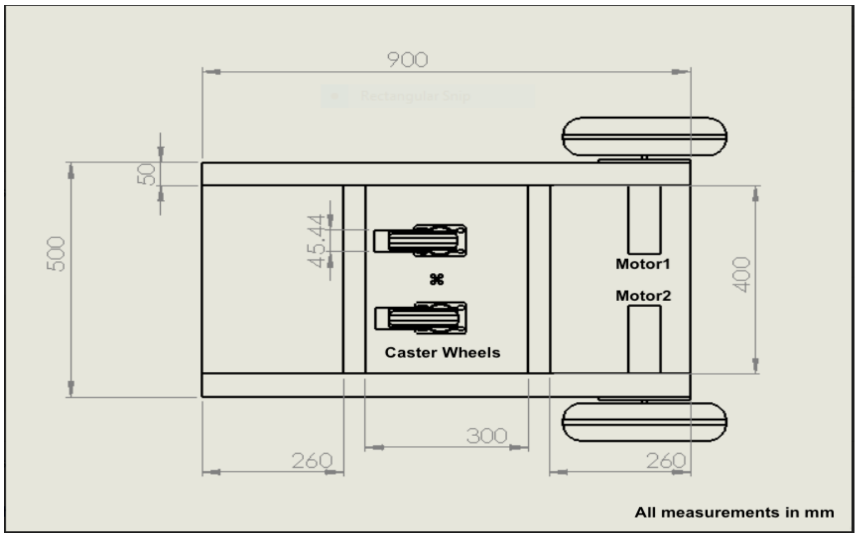


Fig.3. Motor & caster wheel placement

4. Front Caster Vs Rear Caster

Keeping the caster at the front of the vehicle would make the robot sway in the direction of the caster which is usually controlled by the motion of the powered wheels. However, on rough and undulated surfaces, the caster generates its own direction at every moment which makes the robot deviate from the desired path. So, the caster in Eklavya 6.0 is placed at the rear of the vehicle.

5. Wheel hub:

The wheel hub was designed from scratch. After several iterations, the team arrived on the following optimal design:

5.1. Motor Hub : The motors have been mounted on plates made of mild-steel material firmly attached to the horizontal wooden beams.

5.2. Couplers : The couplers are tightened on the shaft using two screws ensuring even surface contact with the shaft and the torque is transmitted from the motors to the wheels efficiently through couplers attached to the motor shaft connected using 6 nut-bolts to identical couplers attached to the wheel shaft. A detailed analysis of the couplers was done on Ansys to check for any torsional failures or bending (as shown in fig.4.)

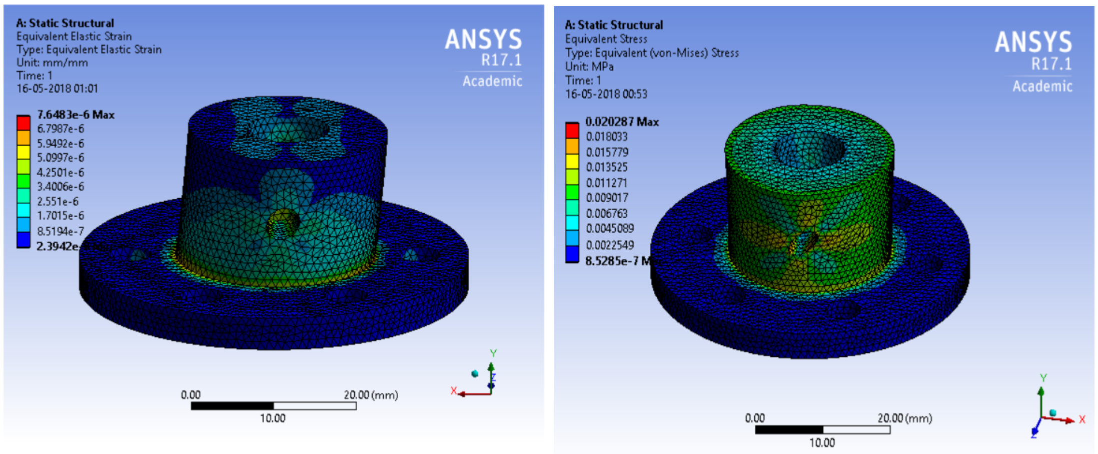


Fig.4. Coupler ANSYS analysis

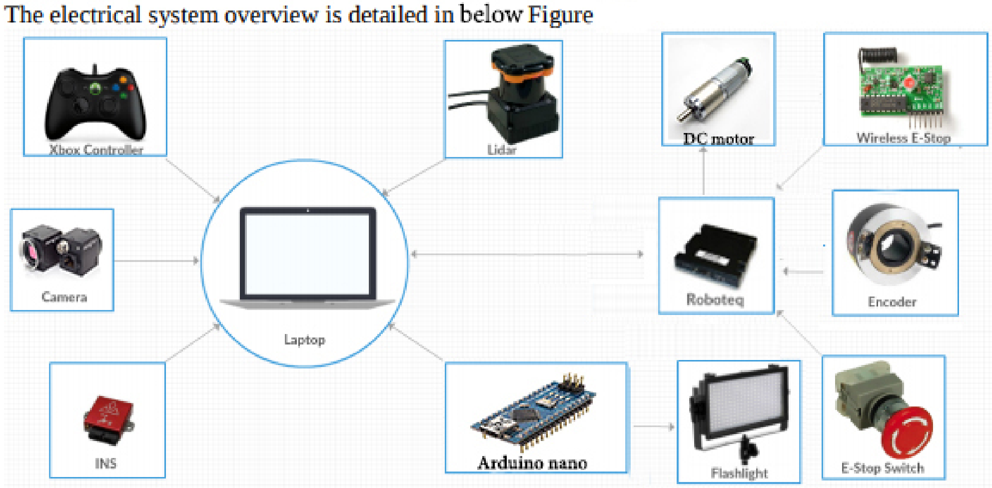
Electronics and Control Systems

1. Electronic and Power Design

1.1 Overview

The Electrical system of Eklavya 6.0 has been improved from our last vehicle. The system consist of 2 high torque DC motors, Roboteq MDC2230 Motor Controller, sensors like Lidar, Camera, Encoders, GPS and IMU, RF transceiver for Wireless Estop and a Laptop. The design focuses on safety, robustness and dynamic controls. The complete electrical routing is shown in Figure.

The electrical system overview is detailed in below Figure



Electronic Architecture

1.3 Sensor and Actuator specifications

Actuator Specifications

Geared Motor	● Operating Voltage :- 24V ● Current :- **Max -** 30 A **No Load -** 1.12 A ● Rated Torque:- 142 Kg Cm ● Gearbox Ratio: 1:6
Roboteq MDC2230 Motor Driver	● Built-in high-power power drivers for two DC motors at up to 60A output per channel ● Dual Quadrature Encoder inputs with 32-bit counters. ● Up to 6 Digital Inputs for use as Deadman Switch, Limit Switch, Emergency stop or user inputs

Sensor Specifications

VectorNav VN-200	● 3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer, barometric pressure sensor. ● GPS-aided Inertial Navigation System (INS). ● Low power consumption. ● Accurate Signal output owing to Internal Kalman Filtering.
HOKUYO UTM-30LX Lidar	● Range of 30 m in 270 degree Plane of device ● Millimeter resolution in a 270° arc. ● Accuracy ±50 mm within a range of 0.1-30 m
BFLY- 23S6 Camera	● On-camera image processing: color interpolation, gamma, and LUT ● 16 MByte frame buffer; LED status indicator.
Planetary Encoder	● 2 Channel Quadrature Encoder ● 2000 CPR

1.4 Power Design

All components in the vehicle except motors are powered by a 12V Lead Battery and the motors are powered by two 12V Lead batteries in series. Each battery has a capacity of 17000 mAh. Specification of the Power consumption details of different parts of the Robot are given below.

Component	Quantity	Rated Power
Planetary Encoder Geared Motor	2	100 W X 2
Roboteq MDC2230 Motor Driver	1	10 W
VectorNav VN-200	1	0.5 W
HOKUYO UTM-30LX Lidar	1	8.4 W
BFLY- 23S6 Camera	1	2.5 W
GPS	1	0.5 W
Flashlight	1	4.8 W
Total		226.7 W

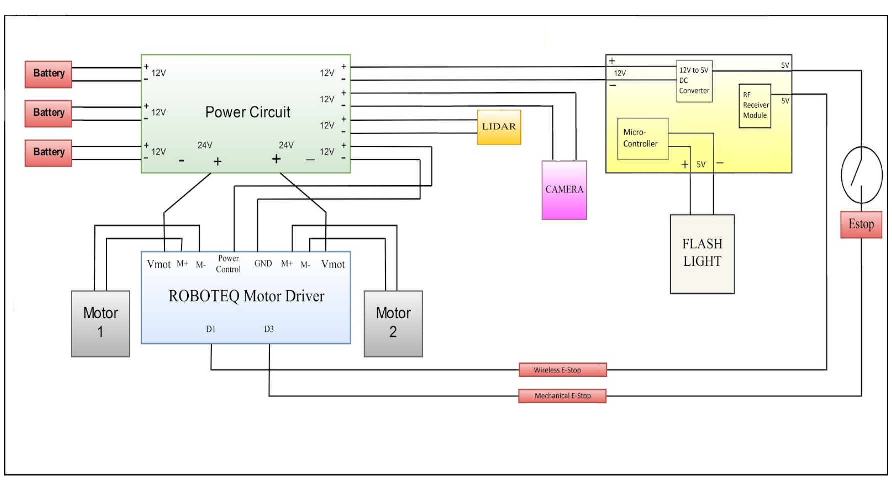
Hence the calculated run time of Eklavya 6 Motors with fully charged batteries is

$$Minimum\ Time\ for\ other\ components = \frac{1 \times 12000\ mAh \times 12V}{26.7\ Jsec^{-1}} = 7.64\ Hours$$

$$Minimum\ Time\ available\ for\ Motors = \frac{2 \times 12000 mAh \times 12V}{200\ Jsec^{-1}} = 2.04\ Hours$$

However, operating power consumption is less than half of the maximum power consumption. Hence, the vehicle can run up to 2.5 to 3 hours with all electrical components and sensors working at the same time.
Each battery takes approximately 1 hour to charge from a 12V DC and 4 Amperes supply.

1.5 Power distribution circuit



1.6 Emergency Stop

Mechanical

- There is a red coloured push button present on the Eklavya 6.0 which is the primary safety stop mechanism.
- When pushed, it sends a 5V high signal to the Roboteq controller's Digital Inputs which triggers the E-stop.

Wireless

- When the RF Transmitter module sends the stop command upto a range of 200 meters, the RF receiver module sends a 5V high signal to the Roboteq controller Digital input which triggers the wireless E-Stop.

1.7 Safety Systems and their Integration

- In order to ensure the sensors are safe from sudden voltage fluctuations capacitors, diodes, voltage regulators are integrated in the circuit design.
- Fuses of proper current rating are connected to ensure no damage to the electrical components and sensors. LED indicators are used to detect any power cuts / malfunctions in the battery.
- Each sensor has its own switch and individual sensors can be switched off if needed. Heat shrinks are used to cover open wires.

2. Control System :

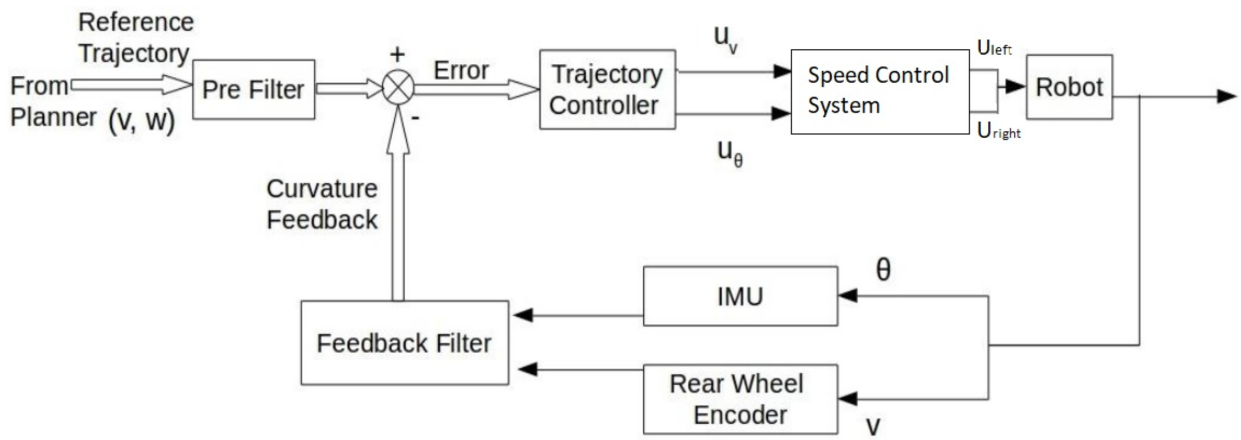
The speed control system and the curvature control system are the main control system of Eklavya 6.0. The control system is implemented on the Roboteq motor controller.

The speed control system tries to reject the environmental disturbances and tracks the given speed. The linear and angular velocity as received by the planner are converted to the differential velocities. PID control scheme is chosen because of its ease of implementation and the degree of freedom of tuning three parameters to achieve better performance. The speed feedback is obtained using the two front wheel encoders.

The experimentally tuned PID control scheme was verified by simulations on MATLAB. Using system identification techniques, a transfer function model was obtained for the two DC motors.

$$R = \frac{l}{2}\frac{V_l + V_r}{V_r - V_l} \; ; \; \omega = \frac{V_r - V_l}{l}$$

The Roborun utility of Roboteq helps in tuning the performance of the speed control system. The following block diagram explains the implemented control scheme.

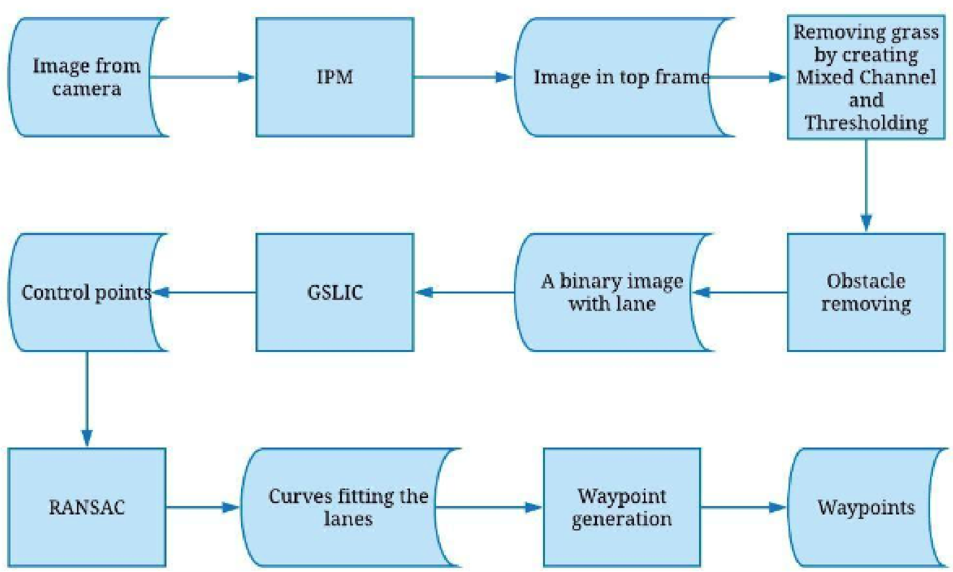


3. Innovations and upgrades

- Replaced brushless DC motor with brushed DC motors as Brushless DC motors are easily damaged and the hall sensors can also be unreliable at times.
- Used Arduino Nano, instead of Beaglebone and Arduino Mega (used in Eklavya earlier), which takes less space and has lower power consumption.

SOFTWARE

1. VISION :



Vision Module

1.1 Innovation and Upgrades

- Novel parabola fitting algorithm for lane detection and waypoint generation.
- Implemented CUDA enabled parallel programming to improve the efficiency of the Simple Linear Iterative Clustering algorithm.
- Optimized and implemented background subtraction algorithm for grassy environments.

1.2 Inverse Perspective Mapping

The vision system acquires an image of an object in front view. Our algorithm exploits the parallel nature of lanes ,thus applying an IPM transform produces an image of top view.

$$[x\, y\; 1]^T = H.[x^1\, y^1\; 1]^T$$

Where H is the homography matrix.

After converting to top_view, we apply normal thresholding and binarize the image such that it only contains lanes and white part of the obstacle. Now we need to reduce the effect of the obstacle.

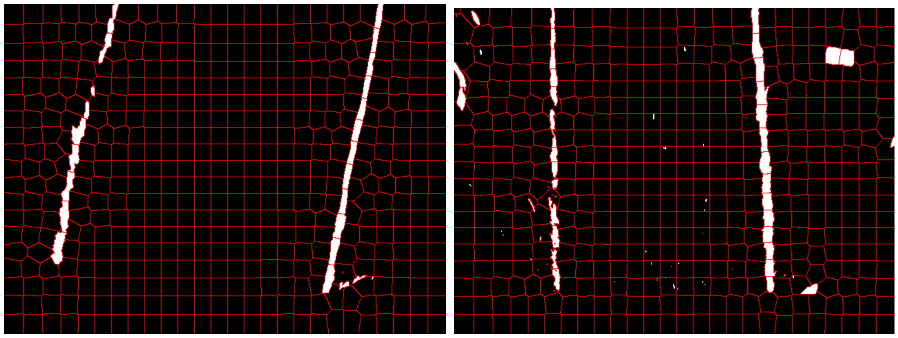
1.3 SLIC-Simple Linear Iterative Clustering

Our approach generates superpixel by clustering pixels in 5-D space [labxy] where [lab]correspond to lab color space and [xy]are our pixel coordinates.We used normalized distance metric for color similarity and pixel proximity in 5-D space.

SLIC is a clustering algorithm that tries to divide the image into a number of clusters such that each cluster has a uniform colour. The algorithm is quite similar to K-Means Clustering Algorithm. Centres for clusters are allocated at a local minimum in a region of a defined size. This is followed by an iterative process of assigning points to various clusters based on distance followed by updating the centre as the mean of the allotted points. The distance is calculated as

$$distance = \sqrt{\left(\frac{dc}{nc}\right)^2 + \left(\frac{ds}{ns}\right)^2}$$

Where $nc = (maximum\ cluster\ distance)$ and $ns = N(superpixels)$



We implemented a GPU based version of SLIC called G-SLIC.

1.4 RANSAC

Random sample consensus(RANSAC) is a computational paradigm used to fit a mathematical model to experimental data that contains noise. It is an algorithm which finds the noise and makes a model without the influence of it. It is robust to outliers. This can be broken down into two steps :

- The first is randomly choosing a sample subset from the given dataset with minimum number of elements required for generating any model (in our case quadratic curve) and fit it to the model on it.
- The second step is finding the number of inliers i.e. the data points that can closely fit themselves in the model.

These two steps are iterated and the best model is found.A model is considered best if it has maximum number of inliers. Once we get the control points, we need to fit one or two quadratics depending upon whether we have one or two lanes in view. After trying out a few equations we found the equation to depict the roads to be the best.

$$y^2 = \lambda(x - a)$$

We apply ransac to the center of superpixel clusters. This reduces the large part of obstacle as they are clustered together. We select 4 points, 2 pairs for each of the quadratic curves as required for defining each of the two curves. Then we check the minimum distance of each point from the quadratic and if its distance is less than a threshold value we classify them as inlier and as outlier otherwise. We iterate it 1000 times and check its score depending on the number of inliers, and consider the ones with best score. To improve the results we also used the fact that two lanes would not intersect each other in the top view and if the two quadratics happen to intersect they are not considered to be one of the possible equations of the lanes.

1.5 Pothole detection

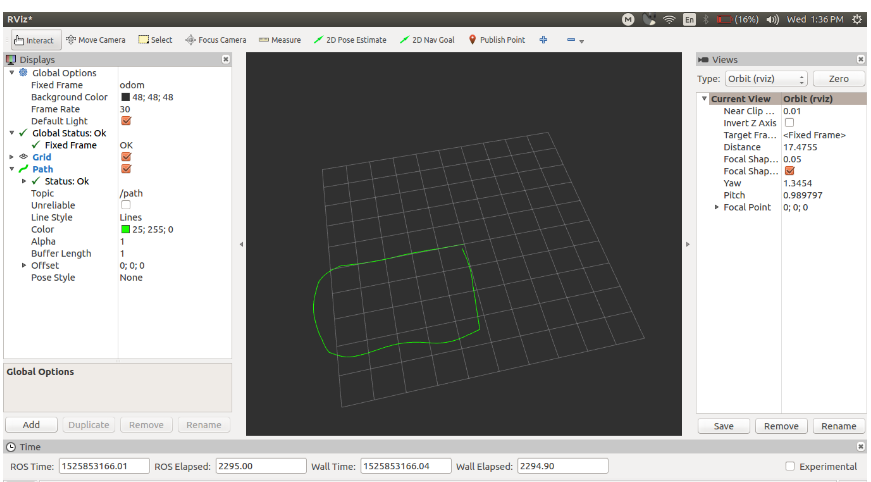
Due to the circular shape of the potholes, we deemed it fit to apply the Circular Hough Transform to detect circles of appropriate radius.

2. LOCALIZATION

2.1 Robot Localization

Localization means having a fairly accurate estimate of the robot's state at all times with respect to its surroundings. In order to localize itself, a robot is required to have relative and absolute measurements that contain information related to its position(x) and velocity(x). The sensors give their feedback about the robot's movement and environment around the it. Given this information, the robot has to determine its location as accurately as possible. The uncertain information needs to be combined in an optimal way. We use the Extended Kalman Filter(EKF) (for estimating x,y,(yaw)) for this purpose, which is a kind of Bayesian filter using Gaussian as it's priori. The Extended Kalman Filter is an improvement over the existing Kalman Filter as EKF can handle nonlinear state estimation by using simple Taylor-Series approximations of the nonlinear systems. The Kalman Filter successively fuses the data from the prediction step, which is done by the motion model of the robot(assumed to be differential drive in our case), with the measurement step, which is obtaining the data from the different on-board sensors.

We have two EKF ROS nodes running in our module. One is for the state and measurements in the odom frame which fuses the sensor data and gives predictions in the odom frame. This node takes in the sensor data and outputs the pose and velocity in the odom frame which is fed to the planner for further waypoint navigation. The other one is for the state and measurements in the map frame which incorporates the filtered GPS data in the measurement. This data is fed back to the first node, which is transformed to the "odom" frame through a static transform linkage and is fed into the EKF node in the odom frame.



We tested our robot in a closed loop to check the effectiveness of our localization package. Using a path plotting script written in python and rviz, the closed loop had an error of approximately 10 cm from loop closure.

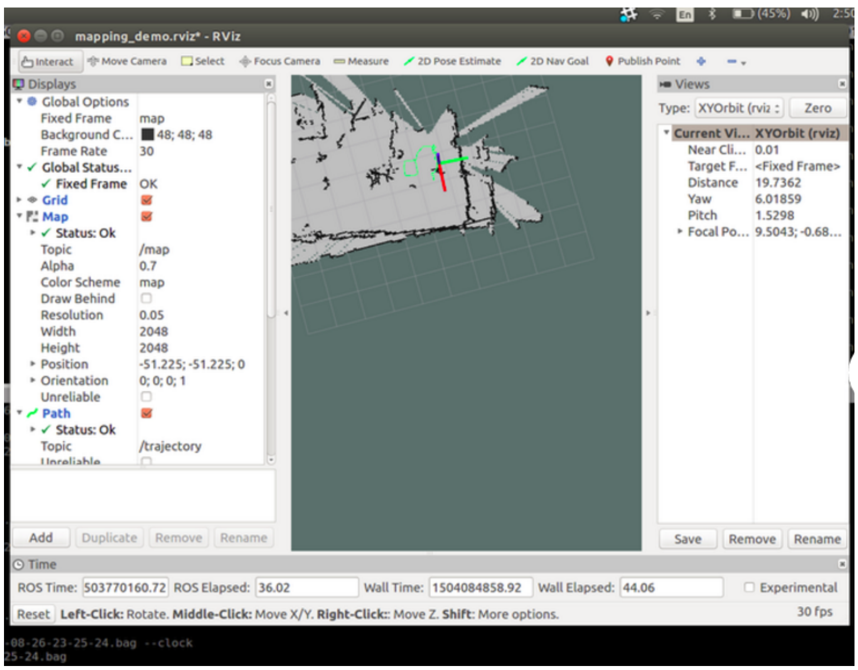
3. MAPPING

3.1 Introduction

Mapping is the key to spatial awareness of robot.The goal here is to construct map or floor plan and to localize in it using feature points. In order to determine the map ,the extraction of features point is essential.The selection of feature points depends upon the type of input data and sensor used.

3.2 GMAPPING

GMapping is a very effective Rao-Blackwellized particle filter to build grid maps from laser scan.Rao-Blackwellized particle filters uses a particle filter in which each particle carries an individual map of the environment.



Map generated using gmapping and Hokuyo Lidar

4. PLANNING

4.1 Innovation and Upgrades

- We have completely shifted from our earlier planner TP-RRT to TEB Local planner which eliminates the randomness and jerky behaviour of the vehicle.
- TEB is a much more optimized planner which takes into consideration a number of feasible paths and chooses the best from among them.
- We also have navfn as our Global planner(A^*) which is the most optimized global planner till date.

4.2 The Local Planner

The Local Planner does exactly what it means - calculates the path in a local environment. Given a destination to reach, the local planner generates the best path and the robot's belief throughout the path, considering all the obstacles and the kinodynamic constraints of the robot. These destinations are usually given by The Global Planner sequentially for the robot to follow until it reaches its ultimate goal.

The Base Local Planner provides a controller connecting the robot with the planner. A local planner creates a value function in the form of a grid called cost-map calculating the costs of traversing the cells of the grids. The controller optimizes over the cost-map of the planner sends the direction of traversal and velocity to the bot. In every cycle, a number of possible trajectories are generated and are filtered down to the feasible ones that do not collide with the obstacles.

The local planner we use is the teb_local_planner that implements the Timed-Elastic Band implementation for online trajectory optimization. The initial trajectory generated by a global planner is optimized during runtime with respect to minimizing the trajectory execution time, separation from obstacles and compliance with kinodynamic constraints of the robot such as satisfying maximum velocities and accelerations.

4.3 Global Planner

The Global Planner is responsible for handing out the best optimal path from the start to the destination considering the in between obstacles. It then generates a series of waypoints throughout the path and hands it over to the local planner which then decides the best trajectory to reach the intermediate goal. For the planning, we use the standard A* algorithm as our global planner which is heuristic based algorithm for graph based planner. Heuristic can be as simple as the Euclidean distance between the nodes, or the sum of the x and y distances(eg. Euclidean distance, Manhattan distance,etc.)

4.4 High Level Planner

Our algorithm has been implemented using the FSM (Finite State Machine) concept. An FSM is a mathematical model of computation in which an abstract machine can be in exactly one of a 'finite' number of states at any given time.The FSM keeps track of the observation from the surroundings and automatically switches the mode of the bot from moving along lanes to following externally given waypoints and vice-versa. This runs on top of our Local and Global planners.

4.5 Waypoints

The planner works fine, but unless there is a destination to reach, the global planner, and by extension the local planner, cannot function. So, someone has to take the responsibility of feeding the global planner with the destinations for it to calculate the path. Thus these two entities come into action:-

- Lane Navigator
- Waypoint Navigator

The Lane Navigator takes aid from the camera mounted on the robot to analyse the surroundings of the robot and thus detecting the lanes in which the robot has to travel. It gives a waypoint to the global planner that is within the lanes and is free of the obstacles. The Waypoint Navigator comes into action when there is no lane to guide the bot and only has the gps coordinates of the points in which the robot has to travel via. It takes in the gps-coordinates and converts it into the 'map' frame of the robot and then gives it to the global planner for further processing. The transition between these two states is controlled by a flag, inside_no_mans_land. When the bot is inside no man's land, which is the region of no lanes, the FSM is in the Waypoint Navigator state and when there are lanes to follow, it switches to the Lane Navigator state

COST REPORT

Component	Quantity	Retail Cost (USD)	Cost to Team (USD)
Roboteq MDC2230 Motor Driver	1	275.00	275.00
VectorNav VN-200	1	2600.00	0 (Sponsored)
BFLY- 23S6 Camera	1	575.00	575.00
HOKUYO UTM-30LX Lidar	1	4974.00	4974.00
Planetary Encoder Geared Motor	2	210.00	210.00
Asus FX553VD	1	1000.00	1000.00
Xbox 360 Wireless Controller	1	35.00	35.00
Lead Acid Battery	3	92.40	92.40
Arduino Nano	1	3.74	3.74
Miscellaneous Circuit Elements	NA	50.00	50.00
Rubber Wheels	2	20	20
Building Materials & Fabrication	NA	150	150
TOTAL		9985.14	7385.14