UNMESH MASHRUWALA
Innovation Cell
IIT BOMBAY

BUDGET PROPOSAL

A report of the technical aspect & budget estimate for the Intelligent Ground Vehicle Competition [IGVC] 2023-24 and SeDriCa

1 - Details of Project

1.1 Intelligent Ground Vehicle Competition

1.1.1 Overview

The IGVC offers a design experience that is at the very cutting edge of engineering education. It is multidisciplinary, theory-based, hands-on, team-implemented, outcome assessed, and based on product realization. It encompasses the very latest technologies impacting industrial development and taps subjects of high interest to students. It comprises of 2 competitions: the Self Drive and the AutoNav challenge

- The purpose of Self-Drive is to develop student, faculty, and university skills and experience which encompasses those required to develop automotive smart driving cars or defense intelligent vehicle systems compatible with current roadways and future intelligent highway systems.
- The Auto-Nav challenge involves making a smaller bot to tackle an outdoor obstacle course. It provides the opportunity to learn and explore autonomy at a more basic level, which would help lay the foundations of future innovation in autonomous ground vehicles.

1.1.2 Self-Drive Challenge

1.1.2.2 Objectives of Problem Statement

A *fully autonomous unmanned ground robotic vehicle* must navigate around an outdoor obstacle course under a prescribed time while maintaining a minimum speed of 1 mph over a section and a maximum speed limit of 5 mph, remaining within the lane, and avoiding the obstacles on the course.

- Vehicles must be unmanned and autonomous.
- They must compete based on their ability to perceive the course environment and avoid obstacles.
- All computational power, sensing, and control equipment must be carried onboard the vehicle.
- No base stations allowed for positioning accuracy are allowed.
- Teams are encouraged to map the course and use that information to improve their performance on the course.

1.1.2.3 Capabilities

- Camera Vision systems
- Lane following, Obstacle and Pedestrian avoidance
- Roadway and Parking driving maneuvers
- Road network navigation, Road signs and other Traffic functions

1.1.2.4 Design Constraints & General Rules

- The Self-Drive competition is designed for FMVSS-500/EU Quadricycle type electrical vehicles (EV) equipped with automotive drive-by-wire systems.
- The vehicle must be Side by Side 2-person four-wheel ground electrical vehicle with Maximum dimension specifications as (115x60x75 in) and a maximum weight of 1500 lbs. Speed is limited to 5 mph in 2022 as safety features of the Self-Drive course are developed.

- There should be Mechanical E-stops located inside the cabin, as well as outside on the sides and rear of the vehicle which on activation should bring the vehicle to a quick complete stop. There should be a wireless E-Stop that must be effective for a minimum of 100 feet and strobe light must be mounted on the roof and activated when the vehicle is under robotic control.
- Vehicles must be unmanned and autonomous. They must compete based on their ability to perceive the course environment and avoid obstacles. Vehicles cannot be remotely controlled by a human operator during competition. All computational power, sensing and control equipment must be carried on board.
- The maximum speed of the vehicle in front should not exceed 5 mph and the minimum is 1 mph and Maximum reverse speed direction should not exceed 2 mph.

1.1.2.5 Qualification Tasks

- 4 manual E-stop buttons for manually stopping the car in case something goes wrong.
- 2 wireless E-stop buttons should work in the range of 100 feet.
- Lane-keeping - intended to evaluate if the vehicle can stay within lane boundaries, without wheels crossing the line or driving on the line.
- Left & Right turn - intended to evaluate if a vehicle can make a turn across the traffic, merge into the expected lane and drive within this lane until an obstacle is detected.

1.1.2.6 Functional Tasks

These tests are intended to evaluate if a vehicle can follow a lane, change lanes, detect and avoid obstacles, detect signs, merge into a loop and park at the specified locations.

- White Line Detection - intended to evaluate the detection of white lines using traditional Machine Vision algorithms. A GUI interface with extracted white lines MUST be present during a run.
- Static Pedestrian Detection - intended to evaluate the detection of a mannequin using traditional Machine Vision algorithms. A mannequin wears an ORANGE construction vest.
- Tire Detection - intended to evaluate the detection of a small item present in a current lane using traditional Machine Vision algorithms.
- Stop Sign Detection - intended to evaluate Stop Sign classification detection and accuracy. Before the test, a RANDOM picture might be put on top of a STOP sign. A forgery sign could be red with different letters, be a different color with the same letters or be a different picture. A GUI interface shell displays a relevant classification as “Stop Sign” or “Unknown”.

- ❖ Lane Keeping - intended to evaluate if the vehicle can maneuver within lane boundaries, without wheels crossing the line or driving on the line while driving XX road drive.
- ❖ Intersection Testing. Left and Right Turn - intended to evaluate if a vehicle can stop at the 'Stop' traffic sign, make a turn across the traffic, merge into the expected lane and drive within this lane until an obstacle is detected.
- ❖ Pull Out Parking - intended to evaluate if a vehicle can reverse out (or pull out) of the representative parking space.
- ❖ Pull In Parking - intended to evaluate if a vehicle can pull into a representative parking space.
- ❖ Parallel Parking- intended to evaluate if a vehicle can parallel park into the representative parking space.
- ❖ Unobstructed Static pedestrian detection - evaluates the ability of a vehicle to stop if a pedestrian is detected within the boundaries of a current lane.
- ❖ Obstructed Dynamic pedestrian detection - evaluates the ability of an Ego vehicle to stop if an obstructed by barrel pedestrian (mannequin) suddenly starts crossing an Ego's vehicle lane.
- ❖ Static Pedestrian Detection & Lane Changing - imitates a situation of a broken vehicle in a current lane with a Static pedestrian standing behind a barrel and in front of Ego's vehicle. The vehicle must slow down and safely change into an adjacent lane.
- ❖ Obstacle detection & Lane Changing - evaluates the Ego vehicle's ability to safely change lane if a stationary object is present within a current lane.
- ❖ Curved Road Evaluation & Lane Keeping - intended to evaluate the Ego vehicle's ability to stay in the lane on a curved road and be able to stop at the obstacle within a current lane. This test consists of 4 possible case scenarios: driving in the right lane on the left curve, driving in the left lane on the left curve, driving in the right lane on the right curve and driving in the left lane on the right curve. Any of the above scenarios could be chosen as this year's test.
- ❖ Curved Road Evaluation & Lane Changing - intended to evaluate if a vehicle can perform a lane change on the curved road if obstacles are detected. This test consists of 4 possible case scenarios: changing the right lane on the left curve, changing the left lane on the left curve, changing the right lane on the right curve and changing the left lane on the right curve. Any of the above scenarios could be chosen as this year's test.
- ❖ Pothole Detection - intended to evaluate the Ego vehicle's ability to detect a pothole and safely change lanes.
- ❖ Merging - intended to evaluate if a vehicle can perform a merge onto a representative highway.

1.1.2.7 Final Operational Arena

1. Roadway Type: local road with 2 lanes – 10 ft per lane, 10 ft minimum turning radius

2. Geographic Area: Asphalt parking lot, eg. P37 at Oakland University

1.1.3 Auto-Nav Challenge

1.1.3.1 Objectives of Problem Statement

A *fully autonomous unmanned ground robotic vehicle* must negotiate around an outdoor obstacle course under a prescribed time while maintaining a minimum speed of 1 mph over a section and a maximum speed limit of 5 mph, remaining within the lane, and avoiding the obstacles on the course.

- Vehicles must be unmanned and autonomous.
- They must compete based on their ability to perceive the course environment and avoid obstacles.
- All computational power, sensing, and control equipment must be carried onboard the vehicle.
- No base stations allowed for positioning accuracy are allowed.
- Teams are encouraged to map the course and use that information to improve their performance on the course.

1.1.3.2 Course

- The Auto-Nav Challenge is on asphalt pavement which will be approximately 450 feet long in an area of 120 ft wide and 100 feet deep
- Track width will vary from 10-20 feet wide with a turning radius not less than 5 feet
- Outer boundaries will be designated by continuous or dashed white lines approximately 3 inches wide, taped on the asphalt
- Competitors should expect natural or artificial inclines (ramps) with gradients not to exceed 15% and randomly placed obstacles along the course
- Obstacles on the course will consist of various colors (white, orange, brown, green, black, etc.) of construction barrels/drums that are used on roadways and highways
- Natural obstacles such as trees or shrubs and man made obstacles such as light posts or street signs could also appear on the course
- The placement of the obstacles may be randomized from left, right, and center placements prior to every run
- Simulated potholes of 2 feet diameter solid white circles may be inserted. These simulated potholes must be avoided or an end of the run will occur
- If the obstacle is in the middle of the course then on either side of the obstacle will be a minimum of 5 feet of driving space or if the obstacle is closer to one side of the lane then the other side of the obstacle must have at least 5 feet of driving space for the vehicles
- The Course will be primarily sinusoidal curves with series of repetitive barrel obstacles
- Two waypoint pairs for the course will be provided prior to competition. One waypoint pair will be the entrance and exit of the course in No Man’s Land. The two additional waypoints in No-Man’s Land will guide the vehicles to the ramp entrance in either direction
- Five (5) minutes will be allowed for course negotiation

1.1.3.3 Competition rules and procedures

- Each qualified team will have up to two runs (time permitting) in each of three heats. Starting order will be based on order of qualification. Teams will set up on-deck in that order.
- At the designated on-deck time, the competing team will be asked to prepare their vehicle for an attempt. On-deck teams start in the order they arrive in the starting area unless they give way to another team.
- A team will have one minute in the starting point to prep the vehicle at the starting line and point out to the Competition Judges the buttons to start and stop the vehicle

- The Competition Judge will start the vehicle by a one touch motion; i.e. pushing a remote control button, hitting the enter key of a keyboard, a left mouse click, lifting the e-stop up, flipping a toggle switch, etc. The Competition Judge will also carry the E-Stop.
- An attempt will be declared valid when the Competition Judge initiates the start signal at the designated competing time. An attempt will continue until one of the following occurs:
    - The vehicle finishes the course.
    - The vehicle was E-Stopped by a judge's call.
    - The team E-Stops the vehicle.
    - Six minutes have passed after the vehicle run has started for the Auto-Nav Course.
    - The vehicle has not started after one minute after moving to the start line or at the judges' discretion.
- Based on the above allowable run times, if the vehicle has not completed the course in the 5 minute time period, the attempt will be ended by a judge's choice E-stop, with no additional penalty for that run.

1.1.3.4 How the competition will be judged

- A team of judges and officials will determine compliance with all rules
- Designated competition judges will determine the official times, distances and ticket deductions of each entry. At the end of the competition, those vehicles crossing the finish line will be scored on the time taken to complete the course minus any ticket deductions. Ticket values will be assessed in seconds (one foot = one second) if the vehicle completes the course within the run time.
- The team with the adjusted shortest time will be declared the winner.
- In the event that no vehicle completes the course, the score will be based on the distance traveled by the vehicle minus the ticket deductions. The team with the adjusted longest distance will be declared the winner.
- For standard award money consideration, entry must exhibit a sufficient degree of autonomous mobility by completing the Auto-Nav course. If a tie is declared between entries, the award money will be split between them.
- If your vehicle is overtaken by a faster vehicle you will be commanded to stop and your time will be recorded and allowed to be restarted with remaining time after the faster vehicle passes. Total distance will be assessed at the 5 minute mark of runtime.

1.1.4 Other Details

- Registration
    - Feb 2023

- Final event
  - June 2023

1.2 SeDriCa Research Project

1.2.1 Overview

SeDriCa research project aims to develop India's first self-driving car, trying to achieve Level 5 autonomy, capable of autonomous driving under Indian road conditions. The current testing is being done on Mahindra e2o and the plans include further and more precise improvements in the software stack as well as the hardware design.

1.2.2 Objectives

One of our current goals is to have an autonomous vehicle that can successfully be run inside a geofenced arena like our institute. This would include proper lane following, staying within safe velocity bounds, proper braking, efficiently handling intersections and tackling other road obstacles.

1.2.3 Capabilities

We are aiming for level 5 autonomous driving within the institute campus. The vehicle should be able to take in the destination input and autonomously navigate to it. For this we plan to work on building these capabilities into our vehicle.

- Dynamic Planning:
  - We plan to implement a dynamic motion planner so that the car can function properly in a completely unknown and dynamic environment. The software stack is almost ready but yet to be tested on hardware.
- Institute Mapping:
  - One of our capabilities is an Institute level map built using 3D LiDARs. We have tested some packages in the past and have received good results.
- Fully Autonomous:
  - A finite state machine is being implemented which will take care of all the “human-like” decisions like the interactions with traffic signs, lane merging, lane keeping, etc.
- Computer Vision Classification:
  - We have implemented state-of-the-art deep learning models for tasks such as object detection and semantic segmentation to recognize nearby objects and detect the drivable area

1.3 Motivation

1.3.1 Research & Development

Scientific knowledge on autonomous driving technology is expanding at a faster-than-ever pace. For an autonomous car to navigate effectively, technologies from multiple disciplines need to be combined. These disciplines broadly include computer science, electrical engineering, and mechanical engineering. In IGVC, tasks like lane-keeping, lane-changing, parking, pedestrian avoidance, etc requires us to come up with real-life driving solutions for a successful, robust and reliable Self Driving Car.

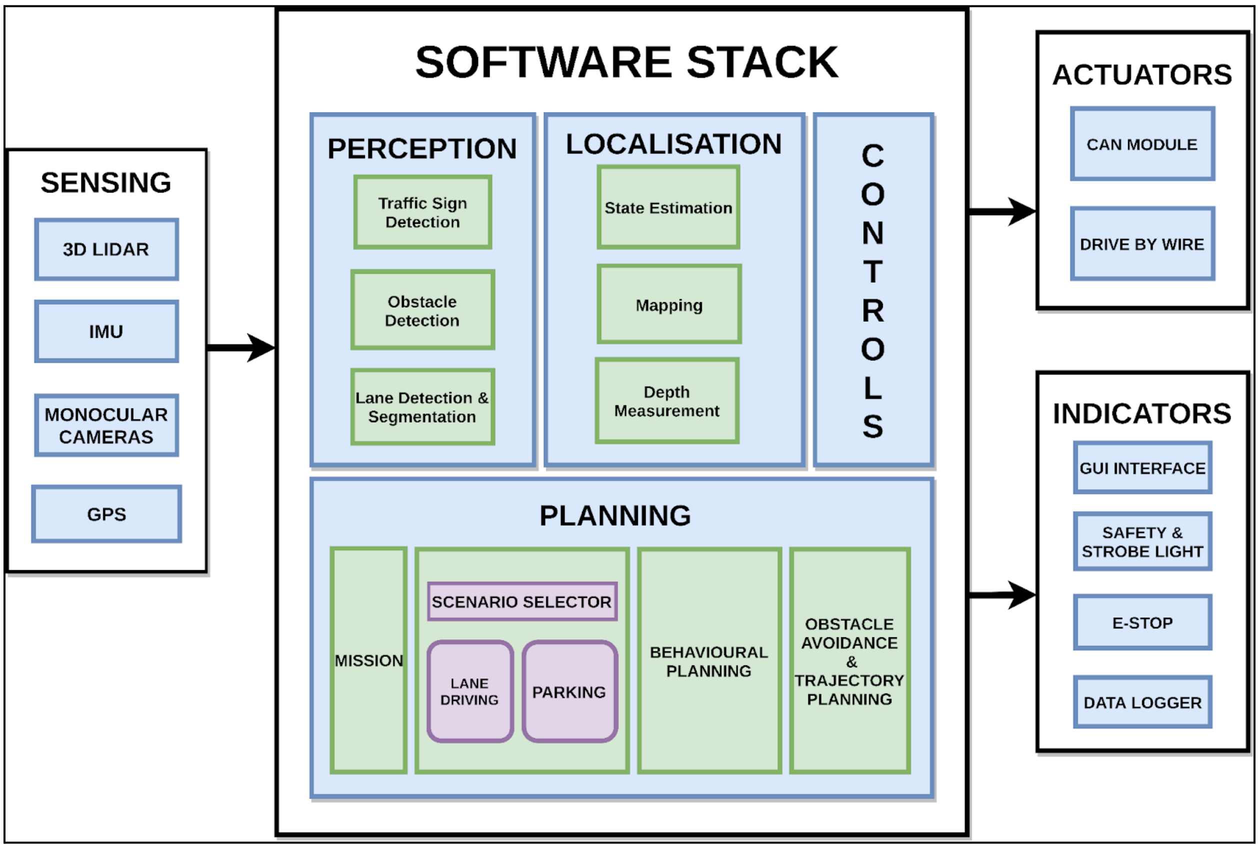
1.3.2 Social Benefit

According to the World Health Organization’s report on road traffic injuries (February 2020), there are approximately *1.35 million deaths per year*, caused by road crashes. Most of these crashes can be attributed to human error. These errors can be caused by over-speeding, driving under the influence of alcohol or distractions during driving (such as usage of mobile phones). These statistics advocate the need for broad adoption and advancement of autonomous vehicle technology. This has multiple benefits, such as a drastic reduction in the number of collisions, higher reliability, better flow of traffic and reduction in traffic congestion.

1.3.3 Innovation

One of our team’s main motivations to take up such problem statements and challenges is to explore the areas unknown to us, an opportunity where we can scratch our brains and come up with some new solutions or algorithms which not only help us solve the current challenge but also act as a starting point for many other advancements in the field.

2 - Technical Approach



2.1 Localization

Finding the vehicle's position and orientation in relation to its surroundings is the responsibility of localization. Because GPS is often accurate to 1m, it is unreliable for motion planning activities. When a car is driven on curving roads and through intersections, the wheel encoders (odometer) provide noisy readings. Magnetic interference from other devices affects the measurements made by the IMU sensor. To correct the errors in the measurements made by each sensor, we intend to combine data from several sensors like IMU, LiDAR, Cameras, and GPS.

2.1.1 Sensor Requirements

3D LiDARs, RGB-D camera, GPS, IMU, Wheel encoder

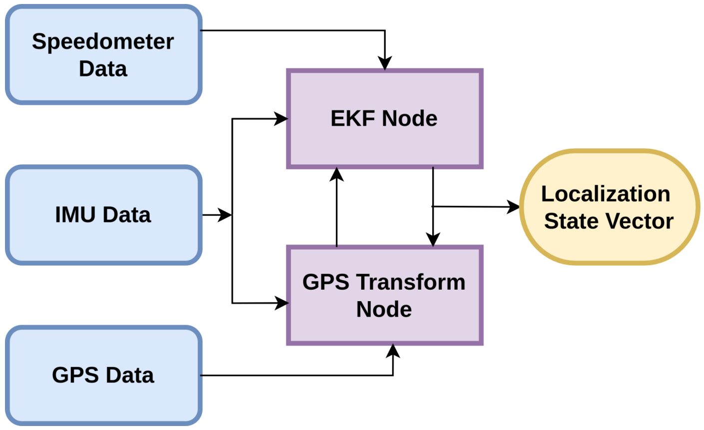
2.1.2 Tasks

1. **Implementing Extended Kalman Filter to fuse GPS, IMU, and Wheel Encoder:**

EKF is an Algorithm with a technique adapted from calculus, namely multivariate Taylor Series expansions, for nonlinear motion systems to linearize them about a working point. It serves to merge data from different sensors for localization. It takes data inputs from multiple sources and estimates unknown values more accurately than individual predictions using single measurement methods.

In a simple explanation to understand the algorithm, let's look at two equations:
One that predicts the state of something and continuously updates that prediction. The predict equation uses the previous forecast of the state (the range of possible state values calculated from the last round of predict-update equations) along with the motion model to predict the current state. This prediction is then updated (via the update equation) by combining the sensory input with the measurement model. We end up with a new range of possible state values, which turns into input for the new predicted equation — and again, we calculate the subsequent measurement to update the prediction. This process allows us to use sensory input to predict where that thing is and where it will be in the next increment.

Fusing GPS, IMU, and wheel encoder data with an Extended Kalman Filter (EKF) can improve the accuracy and robustness of the estimated state of a system, such as the position, velocity, and orientation of a vehicle. Each of these sensors provides information about the system's state vector with maximum accuracy.



Flow chart of working of EKF Package

2. **Simultaneous Localization And Mapping (SLAM):**

An accurate map of the environment is required for precise robot movement. SLAM (simultaneous localization and mapping) is a technique for autonomous cars that enables simultaneous map construction and vehicle localization. Depending upon the sensors used, there are two main types of SLAM:

3. **Visual SLAM:**

Visual SLAM, also known as VSLAM, uses images captured by cameras and other image sensors. Visual SLAM can use simple cameras (wide angle, fish-eye, and spherical cameras), compound eye cameras (stereo and multi cameras), and RGB-D cameras (depth camera). Visual SLAM can be implemented at a minimal cost using low-cost cameras. It is challenging to quantify depth using monocular SLAM because it only uses one camera as a sensor. This issue can be resolved in one of two ways: either by locating AR markers, checkerboards, or other well-known objects in the image for localization or by combining camera data with data from other sensors like inertial measurement units (IMUs), which can measure quantities like velocity and orientation.

4. **LiDAR SLAM:**

The technique known as light detection and ranging (lidar) primarily utilizes a laser sensor (or distance sensor). Lidars are employed in applications with fast-moving vehicles like drones and self-driving cars because they are substantially more precise. Laser sensors often provide point clouds in 2D (x, y) or 3D (x, y, z). The laser sensor point cloud offers highly accurate distance measurements to create maps using SLAM. In general, movement is calculated by sequentially matching the point clouds. 2D or 3D point cloud maps can be represented as a grid map or submap (in the case of Google Cartographer).

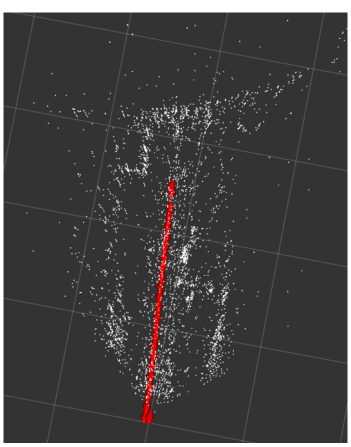
Density-wise, point clouds are less finely defined than pictures and only sometimes offer enough features for matching. For instance, aligning the point clouds in areas with few obstacles might be challenging, which increases the risk of losing track of the vehicle's location. Additionally, since point cloud matching typically demands a lot of processing power, optimizing the processes to increase efficiency is vital. Due to these difficulties, localization for autonomous cars may need to combine the findings of other measurements, such as wheel odometry, GPS, and IMU data.

2.1.3 AutoNav

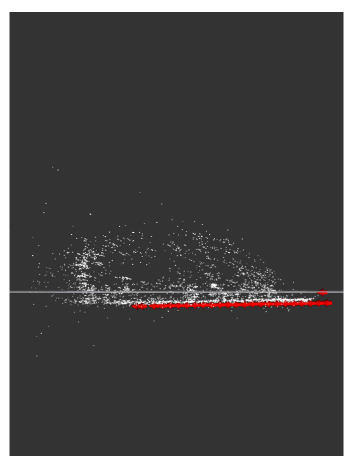
1. **Utilizing V-SLAM ORB SLAM 2 for mapping:**

ORB-SLAM2 is a real-time SLAM library for Monocular, Stereo, and RGB-D cameras that compute the camera trajectory and a sparse 3D reconstruction (in the stereo and RGB-D case with true scale). It can detect loops and relocalize the camera in real time. It provides examples of running the SLAM system in the KITTI dataset as stereo or monocular, in the TUM dataset as RGB-D or monocular, and in the EuRoC dataset as stereo or monocular. It also provides a ROS node to process live monocular, stereo, or RGB-D streams. The library can be compiled without ROS. ORB-SLAM2 provides a GUI to change between a *SLAM Mode* and *Localization Mode*.
We have been working with LiDAR SLAM packages till now. This is a visual SLAM package that we are now working on. It uses cameras as input sources instead of LiDAR SLAM packages that use LiDAR. We use a ROS implementation of the ORB_SLAM2_ROS, as it is integrated with ROS, unlike the original package.
We plan on using an Intel Realsense D455 depth camera. The same testing was done on the given dataset. Explanation of the dataset.

Results:



Top view



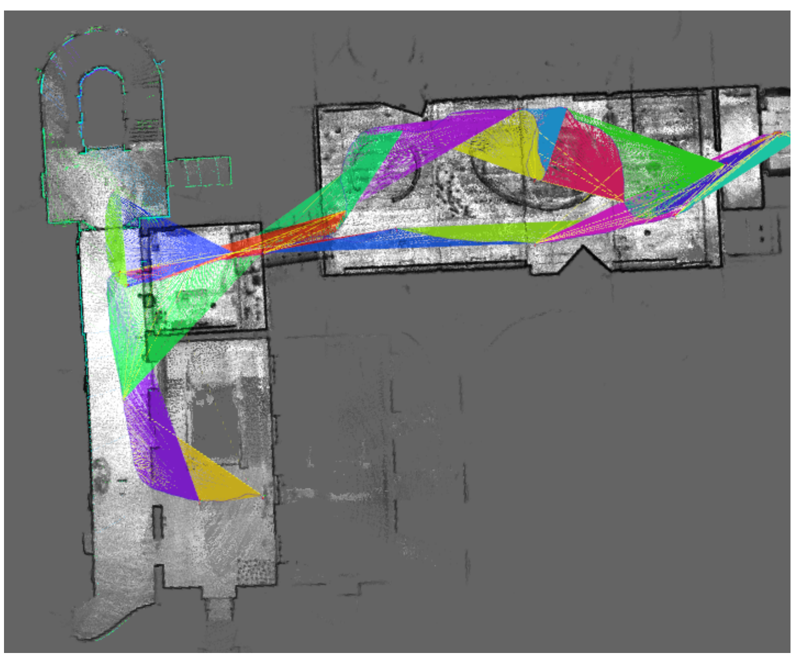
Side View

The point cloud was relatively sparse, as expected, given that it is a Visual SLAM package. However, the map was decently accurate, and its relocalization turned out to be accurate as well. This map was initially saved then. The bot was relocalized after loading the map and using localization-only mode. The red line is the trajectory traversed.
We are now testing the package in our Gazebo simulation to find out how the package behaves when the bot is manoeuvered in specific ways. After this, we will move on to hardware testing using the Realsense D455 camera.

2.1.4 SeDriCa Project

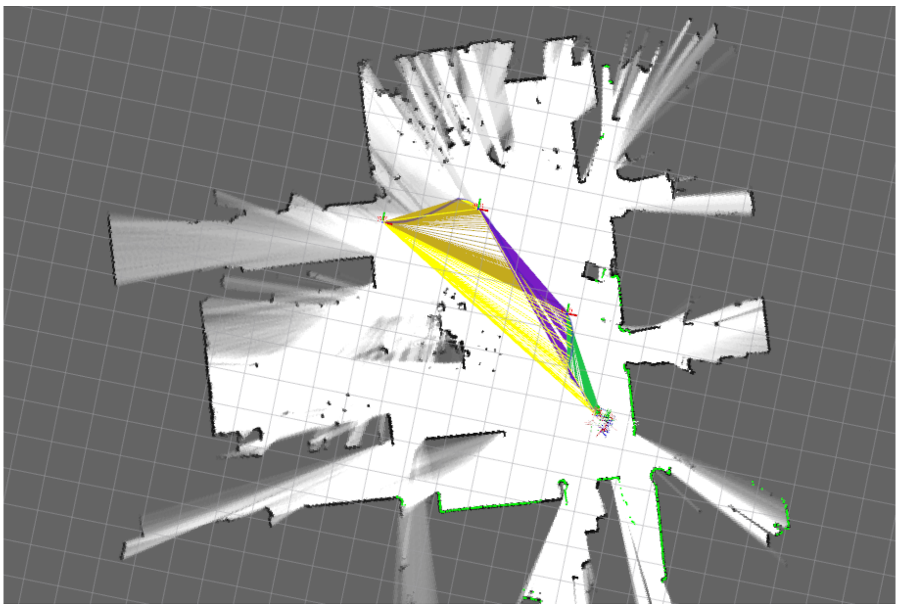
1. **Tuning the Lidar SLAM Google Cartographer for mapping our environment:**

Cartographer by Google provides real-time mapping of the surroundings. This is accomplished by putting the laser scans into a submap at the best-predicted position, presumed to be precise enough for short periods. Because scan matching occurs against a recent submap, it only relies on recent scans, and the error of pose estimations in the world frame grows. Pose optimization is used to address the accumulated error.
The inaccuracy accumulates over time, significantly deviating from the actual statistics. It may also cause map data to collapse or distort, making further searches difficult. The robot's starting and ending positions become misaligned as the faults accumulate. This is referred to as a loop closure problem.



2D map of Deutsches museum

All completed submaps and scans are considered for loop closure automatically. A scan matcher attempts to locate the scan in the submap if they are close enough based on current pose estimates. If a sufficiently good match is identified in a search window around the currently estimated pose, it is added to the optimization problem as a loop-closing constraint.
It is observed that loops are closed instantly when a location is revisited by finishing the optimization every few seconds. This results in the soft real-time constraint that loop closing scan matching must occur faster than new scans are added, or it will lag substantially behind. Using a branch-and-bound technique and numerous precomputed grids per finished submap accomplishes it.

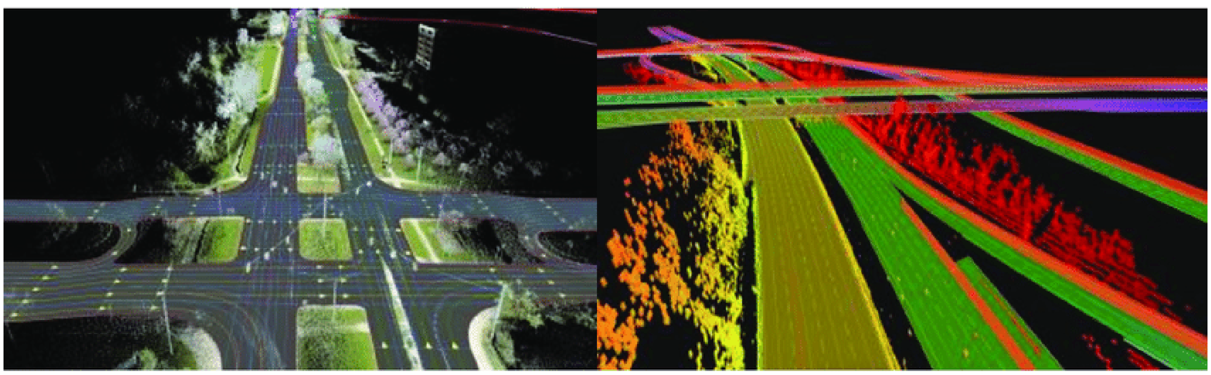


Cartographer generating submap of personal bagfile

We have implemented Cartographer on test bagfiles by making a urdf file containing the links and joints of the sensors used in the robot. We have also tweaked several parameters of the package and observed the changes in the constructed map.
In the future, we will be tuning the package on our vehicle's recorded bag file data and mapping in real-time. We will define the links and joints of the sensors placed on our vehicle in the final design and provide precise transforms among the different frames of reference. Based on the richness of the features available in the IGVC arena and the quality of the sensors used, we will tune the parameters of the package in the Lua file to obtain the best possible results. Cartographer also publishes accurate odometric data of the vehicle as one of its outputs. We can also utilize this data for more accurate sensor fusion. The real-time mapping of the Cartographer also accommodates the case of dynamic obstacles like the other participant vehicles that will run on the arena during the competition.

2. Researching Autoware software for generating HD maps

High Definition (HD) maps have recently become a prominent research topic. They provide navigation details that are exact to the sub-centimeter level, resulting in extremely high precision. HD maps are divided into three layers: semantic, geometric, and location. It allows for further vertical division by the level of detail and horizontal division by geography to improve implementation efficiency.
Autoware is an open-source software stack based on the Robot Operating System (ROS) for self-driving vehicles. It encompasses all the capabilities required to operate an autonomous vehicle, from localization and object detection to route planning and control. It was designed to allow as many individuals and organizations as possible to contribute to open advances in autonomous driving technology.
Autoware's pipeline design enables the creation of self-driving systems. Autoware's pipeline design is made up of components that are comparable to the three-layer architecture mentioned before. They also run in parallel. The Core and the Universe are the two main modules. These modules' components are intended to be expendable and reusable. It is termed micro autonomy architecture.



Reference:
https://www.researchgate.net/publication/327737058_Simultaneous_Localization_and_Map_Change_Update_for_the_High_Definition_Map-Based_Autonomous_Driving_Car/figures?lo=1

The Autoware software will be used to create the HD map of our institute. To accomplish this, we must carefully adjust and test the program to build a very accurate map of the environment. The resulting map will then be annotated with fixed obstacles, landmarks, indicators, and signs based on additional information, such as the institute's satellite map. The final HD map will save significant computational resources that would otherwise be used in other subsystems. The usage of HD maps can also efficiently address sensor errors. With increased use, autonomous driving will become more durable and safe. Following extensive testing, the pipeline can also be used to power autonomous EVs at the institute.

2.2 Computer Vision

The camera forms one of the primary sources of input to autonomous bodies. It proves useful in differentiating between objects where the use of the LiDAR may be cumbersome. Isolation of lanes guides the other subsystems and can only be performed using the camera's input. Obstacle detection and occlusion handling remain one of the more important elements of the CV subsystem.

2.2.1 Sensor Requirements

1. Velodyne VLP-16 LiDAR (to be mounted on top of vehicle)
2. Single camera (mounting style to be decided)

2.2.2 Ongoing Tasks

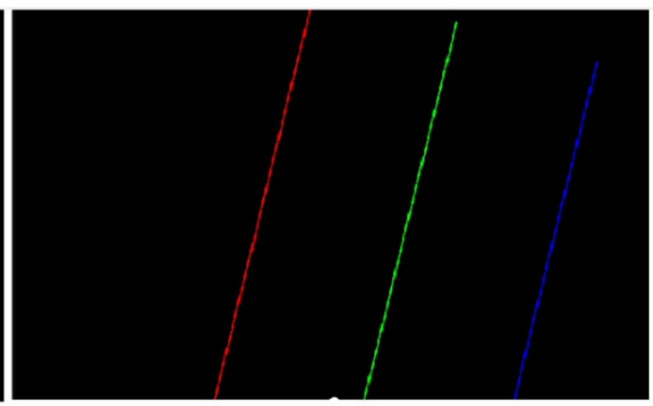
1. **Lidar Camera Fusion**

The LiDAR and camera form two primary sources of input to the car. Point Cloud data provided by LiDAR is different from the frame of 3 cameras installed on the vehicle. Once an entity or object is detected in the frame of the camera, we wish to map its position in the real world map using the LiDAR point cloud data. On a rudimentary level this would involve finding the static transform between the camera and LiDAR frame. This is especially powerful because then we can calculate the exact distance each point on the camera feed from the car. Camera data from appropriate frames can be overlaid over point cloud data to eventually form HD Maps. LiDAR and camera fusion approaches for distance estimation will be crucial in both IGVC & SeDriCa to separate nearer objects from farther, especially when both appear side by side in a frame. Especially at chicanes, where rows of barrels cover the screen to form a wall of sorts leaving the system in a daze.

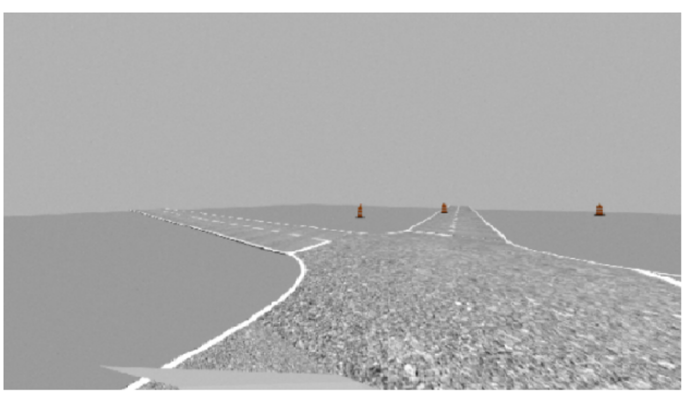
2. Lane Detection

We use image processing techniques powered by the OpenCV library to extract and estimate lanes from a given frame that the bot can see. Our methods involve intelligent use of all 3 camera images depending on what case we are handling and which camera shows a better view of the lanes. We use calculated IPM homographies along with image processing techniques as thresholding, Canny Edge detection and image traversal algorithms to detect relevant lane points. We use these lane points to make a final occupancy grid(in bird's eye view) using the given facts about minimum turning radius and lane widths in IGVC. Currently, each case is handled separately using a flag from the decision making subsystem as different cases demand for different lane detection algorithms. Cases relevant are, curved lanes, straight lanes and intersections.

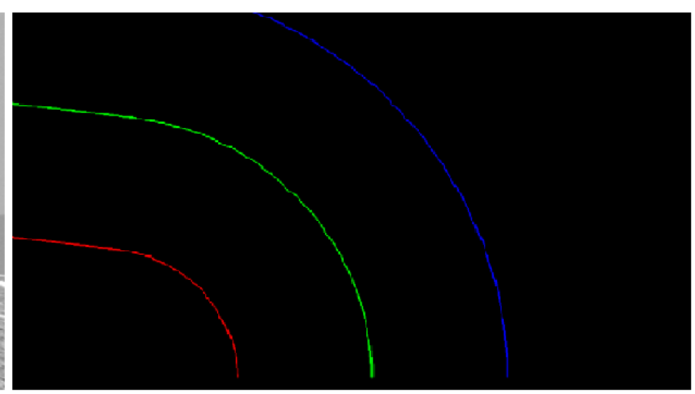
IPM over right camera



Final Lanes



Left Camera Feed at Intersection



Output Lanes for Intersection

2.2.3 IGVC-Autonav

Vehicle would use white lanes on the road to guide itself across the course. We plan on detecting obstacles on the way using Lidar and Camera fusion based data and this would make sure we complete the course while avoiding the obstacles. However, since the bot is supposed to be much smaller and obstacles about the size of vehicle, occlusion proves a major hurdle to the task of lane detection. To deal with occlusion, possible options discussed for placing the camera involve placing it on a rotating camera mount and a larger mount height. Apart from this, during a part of the course, we just have obstacles to guide us along the path and no lane markings. During these points we plan on using midpoints of nearest detected obstacles as the next goal point. We are also given the global goal point position too so we would use that too to guide us. Extrapolation techniques will be employed to reconstruct and complete lanes for optimal functioning of the bot. The problem statement for Autonav involves an aspect to navigation without any lanes, i.e by using obstacles as guidance systems. Covered under the broad umbrella of lane estimation, this task takes autonomous driving to new heights as we remove the cars dependence on lanes for navigation. The path to be followed will have obstacles on both sides and using these obstacles as reference points, we construct pseudo-lanes and give goals to the car.

2.2.4 SeDriCa Research Project

1. **Multi-Task Learning**

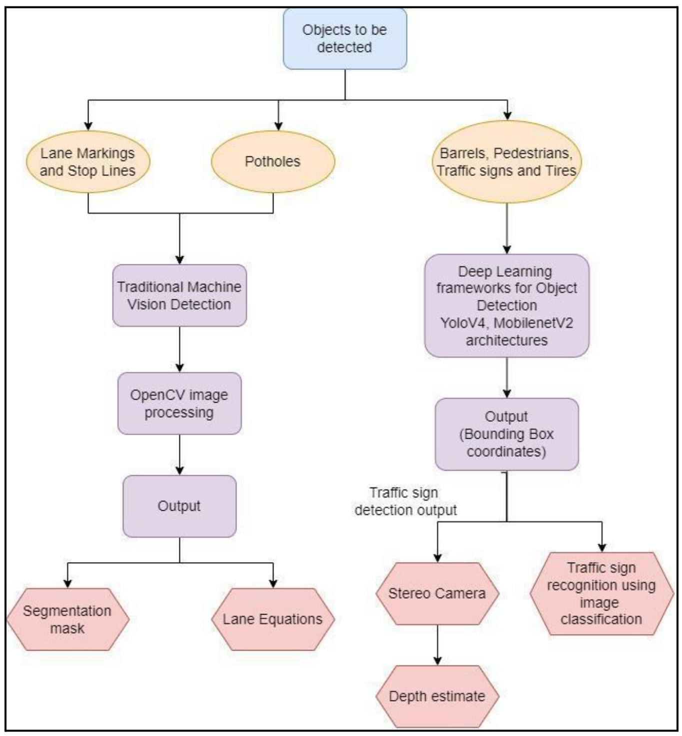
The holy grail for object detection models is to have a single model that is able to differentiate between all different classes of objects and perform semantic segmentation. The benefits of such a model performing multiple tasks are that shared representations reduce overfitting and the auxiliary information can be leveraged to increase learning speed and efficiency. Having the single model perform multiple tasks would also optimize runtime greatly as compared to two or three models running simultaneously.

2. **Multiple Object Tracking**

Dynamic Object Tracking is primarily to locate multiple objects, maintain their identities, and yield their trajectories. The goal of this task is to locate multiple objects, maintain their identities, and yield their trajectories. This can be further extended to high-level tasks such as pose estimation, action recognition, and behavior analysis. This is beneficial as predicting trajectories of dynamic obstacles in the vicinity can help to plan a collision-free path. A Kalman filter may be used to aid the model in 3D pose estimation to increase the tracking performance.

3. Interaction with Other Subsystems

From the computer vision subsystem, the output data is used by the motion planning and the decision-making subsystem. The output of the lane detection model is used by the motion planning subsystem to generate a local map that ensures that the vehicle follows the path marked by the lanes. The output of the object detection model in the form of bounding box coordinates and depth estimate (using the stereo camera) is given to the decision-making subsystem for obstacle avoidance and other functional tasks.



Flowchart of CV pipeline

2.3 Motion Planning

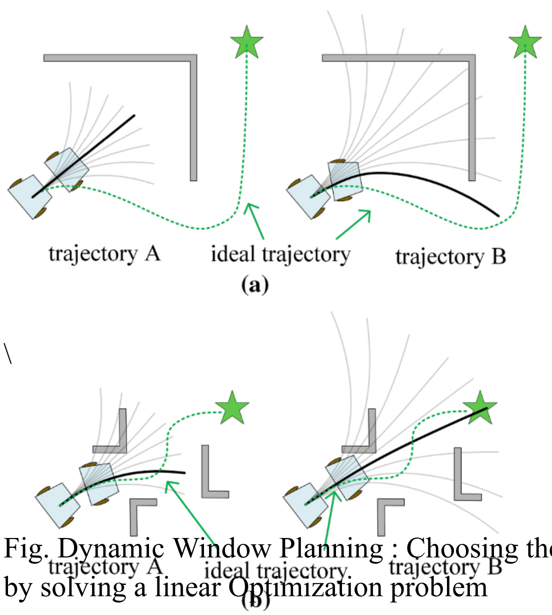
Our previous endeavor was an improvement to the Hybrid A* algorithm for fast and "holonomically accurate" path planning along with the allowance of reversepaths and fail safes in case of errors in obstacle probability fields and occupancy grids.
Our current and future plans for Autonav include planning algorithms that have polynomial computational speed and are capable of providing local paths at successive frames using previous path data and a general idea of the direction of the goal utilizing the obstacle positions. The idea is to implement algorithms that can not only provide quick paths to the goal but also predict the direction in which to reach the destination using the positioning of the obstacles as is appropriate for the problem statement.

Following are the two approaches we follow:

1. **Dynamic Window Approach**

We are planning to use the Dynamic window approach for real time obstacle avoidance in both static and dynamic fields.

- Through this approach, we can essentially control the path in accordance with the restrictions due to the parameters like maximum velocity, acceleration, and steering rate to avoid obstacles and finally reach the goal.
- In this, an objective function with parameters like linear, we create an objective function which has parameters like linear and rotation velocity, a, and acceleration.
- This function makes a trade off between how fast we want to reach our goal and its desire to ship around the object.



\

Fig. Dynamic Window Planning : Choosing the most feasible path by solving a linear Optimization problem

2. **Vector Field Histograms**

We are also doing research on vector field histograms (VFH). The VFH method subsequently employs a two-stage data-Reduction process in order to compute the desired control commands for the vehicle.

- In the first stage the histogram grid is Reduced to a one-dimensional polar histogram that is constructed around the robot's momentary location.
- In the second stage, the algorithm selects the most suitable sector from among all polar Histogram sectors with a low polar obstacle density, and the Steering of the robot is aligned with that direction.

3. **Modified Vector Field Histogram with Neural Networks**

A possible future research topic, this method relies on the foundations of the previous one.

❖ A Modified Vector Field Histogram (MVFH) is developed to improve path planning and obstacle avoidance for a wheeled driven mobile robot. It permits the detection of unknown obstacle to avoid collisions by simultaneously a steering the mobile robot toward the target; a regular grid map representation for a work space environment is carried out

❖ A Neural Network (NN) model is used to learn many critical situations of the environment during robot navigation among obstacles using MVFH. Also, digital filters are utilized for improving the robustness of obstacle avoidance trajectory of mobile robot

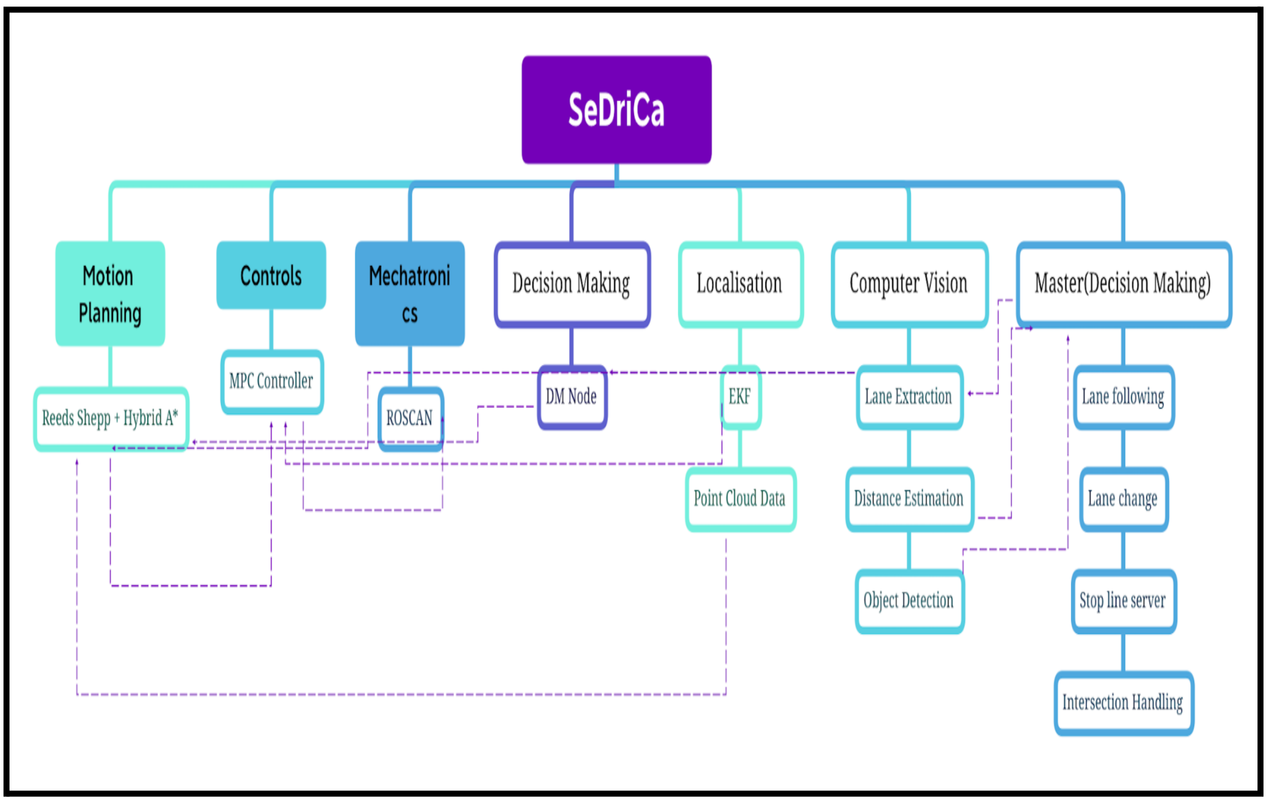
2.4 Decision Making

2.4.1 Tasks

1. **Integration:**

   Need to combine the following nodes for implementation of a functional pipeline for a Self Driving Car :

- ❖ Integrate object detection and classification nodes from the Computer VIsion subsystem as a part of the pipeline for feedback to clients.
- ❖ Combine UKF/EKF packages with parameters tuned and calibrated for IGVC vehicles from the localisation subsystem into the pipeline.
- ❖ Incorporate the Reeds Shepp - Hybrid A* package from Motion planning subsystem for better path planning results.
- ❖ Incorporate solution store method in MPC from controls subsystem into the working pipeline.



2. **Servers Testing:**

- ❖ Need to test Stop sign server with the new MPC node from Controls subsystem and live feedback from Distance estimation nodes of Computer Vision

- ❖ Must test integration server with the corrected lanes coming from Computer Vision subsystem
- ❖ Need to test lane change server with new current lane and corrected lane extraction nodes

3. Master Node Completion:

- ❖ Need to implement a new master node to integrate the stop sign, lane change and lane change clients so as to implement a master client.
- ❖ Following this, the master client needs to be tested with the integrated pipeline so as to test the possible scenarios of the IGVC-23 problem statement

4. Parking and Reverse:

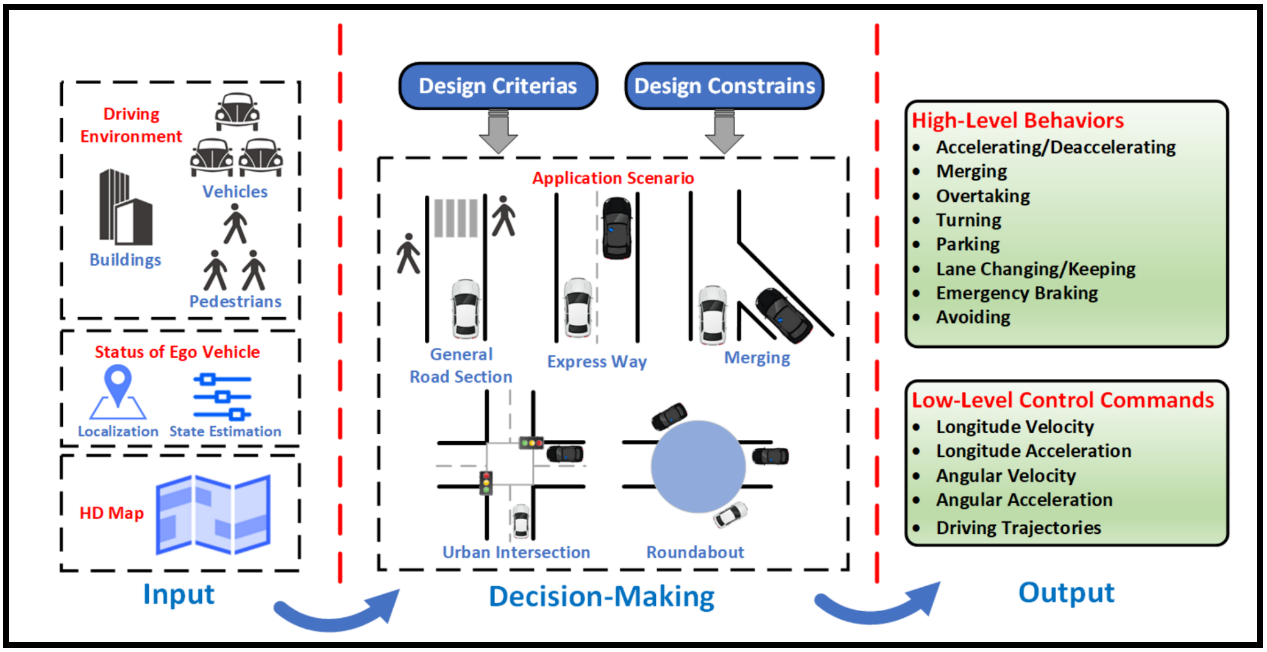
- ❖ Will implement 1 reverse parking and 3 parking scenarios of Pullout, Forward and sideways parking.
- ❖ Need to Integrate use of reverse camera into working pipeline for reverse parking.
- ❖ Will implement a state in the Master FSM to undertake the task of Reverse Driving in Motion Planning, Controls and Computer Vision.

5. Fail Safe Implementation:

- ❖ Will implement a separate state in Master Node to handle Fail safe situations
- ❖ Need to Ideate and fail safe handling scenarios for various possible failure situations
- ❖ Need to ideate implementation of manual mode drive or e-stop or keep the car on neutral during failsafe scenarios.

2.4.2 SeDriCa

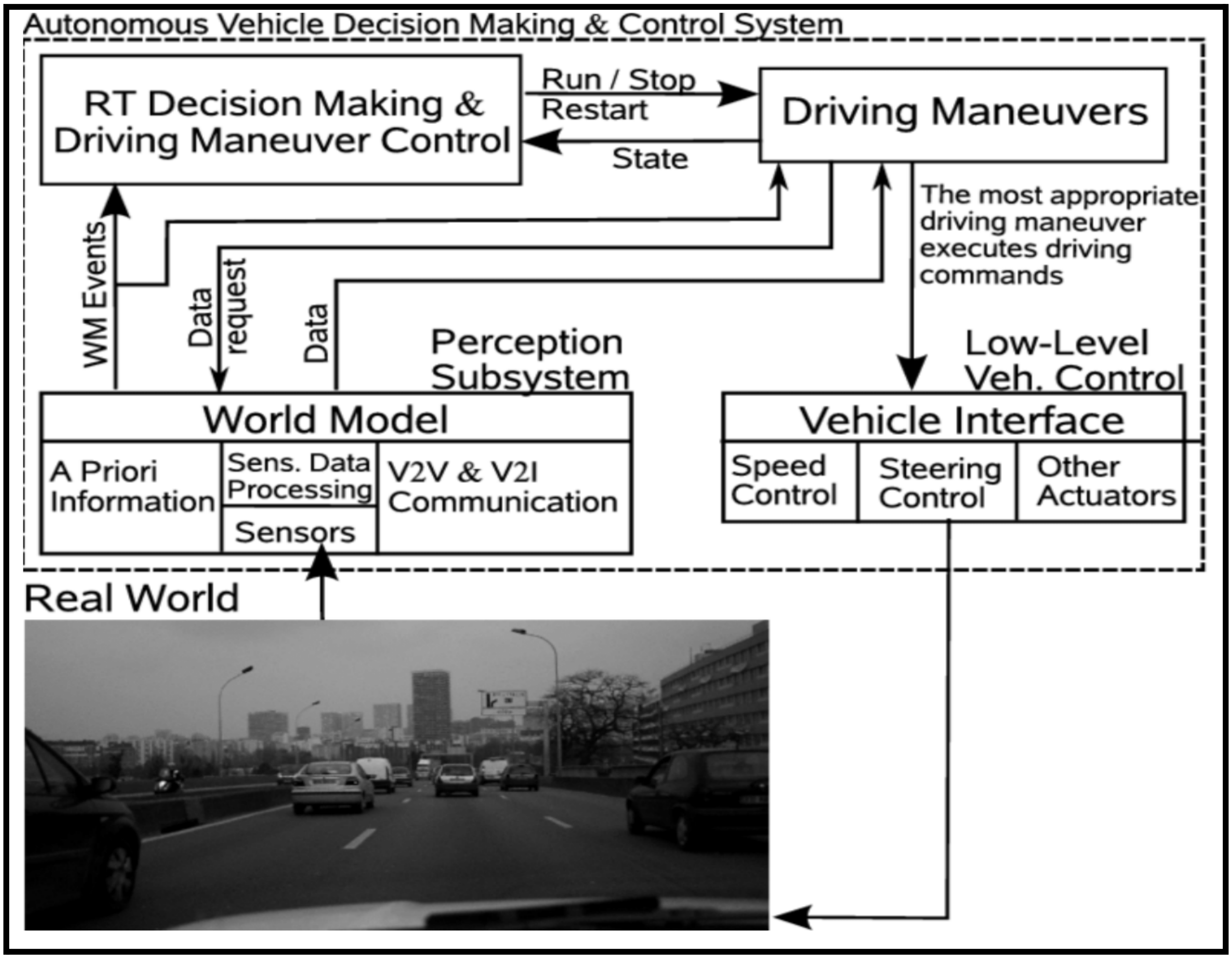
1. **Reinforcement Learning:**



This subsystem of SeDriCa aims to implement a state of the art reinforcement learning model capable of making intelligent decisions to deal with the task of Autonomous Driving while dealing with the problem at hand in the following steps:

1. Autonomous driving on empty roads with the system using feedback from the environment to determine its course of action without having to implement a deterministic Finite State Machine.
2. Autonomous driving on roads with moving obstacles on ideal roads with all traffic signs and signals appropriately marked and an ideal environment where rules of driving are obeyed
3. Eventually we aim to arrive at a level 5 autonomy car capable of navigating through Indian Roads without any human intervention.

The process of arriving at such a model will be to implement several models capable of making rational decisions in scenarios of obstacle avoidance, civilian obstruction, Lane change scenarios with other vehicles intercepting its path, and most importantly to mark local goals in current frame of the car to reduce distance to the global goal over time.

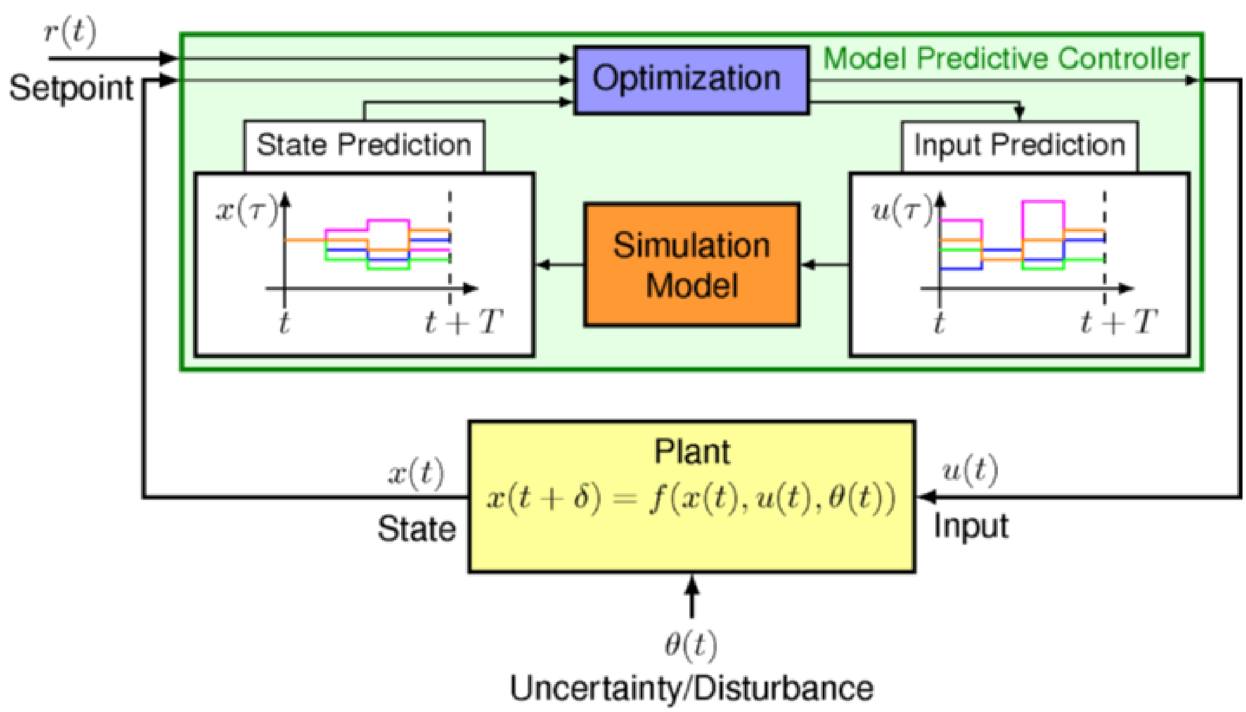


The algorithms that will be tried out for these implementations shall be:

1. Support Vector Machines
2. Optimization Based Control systems without implementation of an FSM
3. Hybrid Flow Diagrams
4. Possibly implement an in - house mechanism for reinforcement learning based decision making processes

2.5 Controls

The control subsystem is responsible for controlling the vehicle dynamics. The task will be accomplished by utilizing the instantaneous planar coordinates, velocity parameters, and other state parameters to calculate required acceleration and steering angle in a given time frame to ensure that the vehicle follows the path calculated by the Path Planning and Decision Making subsystems. For this purpose, we are using the Model Predictive Control (MPC) algorithm to calculate the control inputs.



(Source: University of Stuttgart, Institute of Systems Theory and Automatic Control)

2.5.1 IGVC

1. **Vehicle Dynamics Control**

Vehicle Dynamics:
The MPC algorithm requires a model of the vehicle dynamics as one of the inputs to the algorithm. Currently, a kinematic bicycle model (used for ease of computation, and gives reasonable results) has been used as a model for vehicle dynamics.

$$x_{t+1} = x_t + v_t * cos(\psi_t) * dt$$

$$y_{t+1} = y_t + v_t * sin(\psi_t) * dt$$

$$\psi_{t+1} = \psi_t + \frac{v_t}{L_f} * \delta * dt$$

(L_f is the distance between axles of front and rear wheels)

$$v_{t+1} = v_t + a_t * dt$$

$$cte_{t+1} = cte_t + v_t * sin(e\psi_t) * dt$$
$$e\psi_{t+1} = e\psi_t + \frac{v_t}{L_f} * \delta_t * dt$$

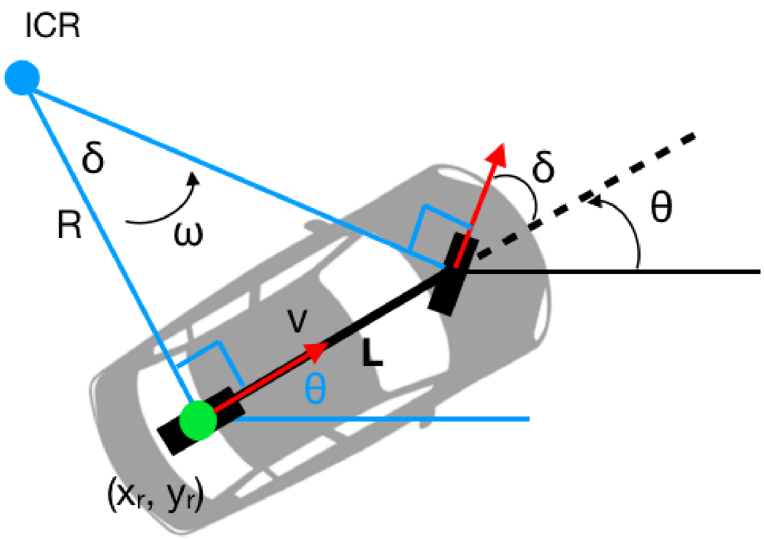
This model has the following state variables and control outputs:
State variables:

- X position
- Y position
- Longitudinal velocity
- Yaw angle
- Yaw angular velocity
- Steering angle

Control outputs:

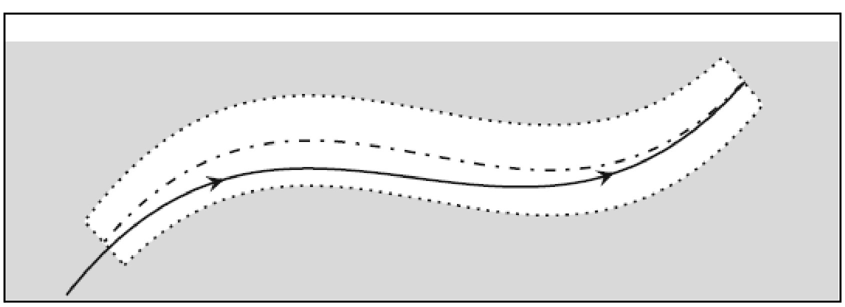
- Longitudinal acceleration
- Steering rate

  These variables are coupled via a set of difference equations that govern the model's physics in the center of mass reference frame.



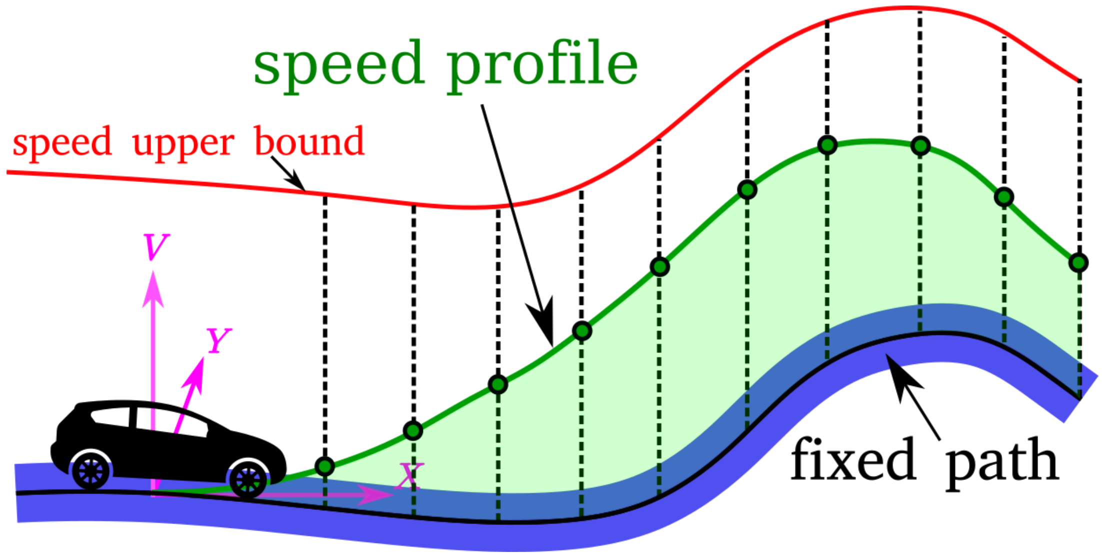
2. Path Following

For path tracking(and velocity tracking discussed in the following subsection), we will use the Model Predictive Control (MPC) algorithm to calculate control inputs at each control interval using model-based prediction and constrained optimization. The reference path plan and the current state is fed into the algorithm as input. The reference path is parameterized by a path parameter whose variation with time is considered an additional degree of freedom. This controls the velocity of the vehicle which is discussed in the section below. This reference is then passed into an appropriate objective function for the Optimal Control Problem along with the design and vehicle dynamics constraints. The MPC algorithm looks N-steps ahead using the vehicle model provided and calculates the best possible inputs for the system by optimizing the objective function. The control outputs thus generated are then applied to the vehicle, and feedback is received for the next step, where this process is repeated to complete the Closed-Loop Control.



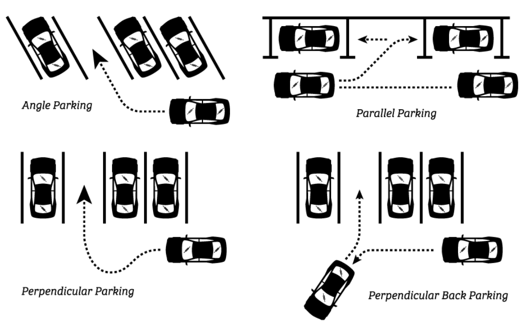
3. Velocity Tracking

As discussed in the previous section, the path parameter and thus the velocity of the vehicle is another degree of freedom in this problem which can be captured using the evolution of path parameters with time. Path following also allows to pre-assign a velocity profile. For this task, the objective function is slightly modified, and a velocity tracking term is added to it while extra velocity constraints are added. This modification will adjust the optimal control problem and result in velocity profile tracking along with path following.



4. **Parking Module**

The parking task requires the vehicle to get inside a specified spot, either parallel or perpendicular to the road and then come to a complete stop at the specified location in a specified orientation.



(Source: Acko Parallel Parking Guide)

5. **Fine accuracy and robustness**

Parking a car requires precision and accuracy compared to regular driving. Hence the controller design should be able to accommodate such behavior. This can be achieved by having sufficiently narrow bounds on the main path.

6. Orientation control

The vehicle needs to traverse in a way that its final orientation when parked within the parking zone. Currently, the approach identified for the same is modifying the objective function which is optimized in MPC to accommodate controlling the orientation too. This would be done by adding specific constraints on the terminal state orientation as well as heavy penalizing of orientation deviation.

7. Reverse parking

In reverse parking, also called parallel parking task, the car is placed parallel to the parking spot, and then it is driven in reverse into the designated parking spot. The control inputs are based on the velocity plan and the path planning. For reverse driving, having a velocity plan with negative values indicating the vehicle to drive reverse can be done. However, special attention to spatial orientation along with fine control on velocities has to be maintained.

2.5.2 AutoNav

The design finalized by the team is a 4 wheel differential drive rover. We wish to implement a controller that would best explore the maneuverability of this vehicle. Finally, we came up with an adapted version of the conventional PID control. A PID control which would control the differential torque on the motor over some ambient torque initially provided to it.

The advantage of such a design is the rover is kept running so that there is no starting torque required within the course of track. The vehicle is kept running as per the rules of the competition. We can achieve greater speed compared to its counterpart(The simple PID control).

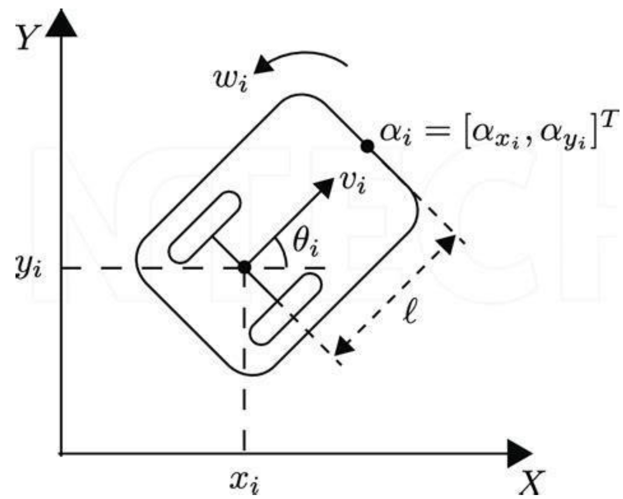


Fig. Two wheel differential drive which we will be modifying to achieve our goal.

The vehicle model we wish to follow is ‘Differential Drive Kinematic Model’. This model was chosen on the fact that we wish to achieve our problem statement with minimal computing power so that we can save up on space and weight our computing device.

Pure-pursuit controller is another way of approaching a quick solution of waypoint based navigation. This algorithm is better tested with steering based vehicles however it can be modified to suffice for a differential drive too.

This control works on the principle of achieving a proper curvature to minimize the cross-track error in further course of track.

2.5.3 SeDriCa

1. **Improving the current algorithm**

a. Model Complexity

The current algorithm uses a model to predict the behavior of the vehicle in the near Future and optimizes the control accordingly. Thus, a model which describes the physical system more realistically allows for better performance. Hence we plan to improve the models currently being used for the Tyre Dynamics, Vehicle Dynamics while adding some models for engine and braking dynamics for better control.

b. Obtaining Better constraints

NMPC solves an optimal control problem using non-linear optimization techniques, which perform better as compared to linear ones. More accurate and better constraints on acceleration and velocity could be essential in further cutting down on processing power. A Literature review for existing methods would be done. Apart from it, we plan to search for novel ways to get physically reasonable constraints to aid the optimisation process.

2. Using Hybrid Control and Planning Algorithms

Integrating Motion Planning and Control allows for better trajectories hence optimizing the control objectives on these trajectories is much more efficient and faster. A single loop for optimisation for planning and control can be used. However, planning algorithms do not require a sample rate as high as the control systems. Hence a nested loop approach can be used where the control loop samples at a much higher rate than the planning loop allowing for a reduced computation budget.

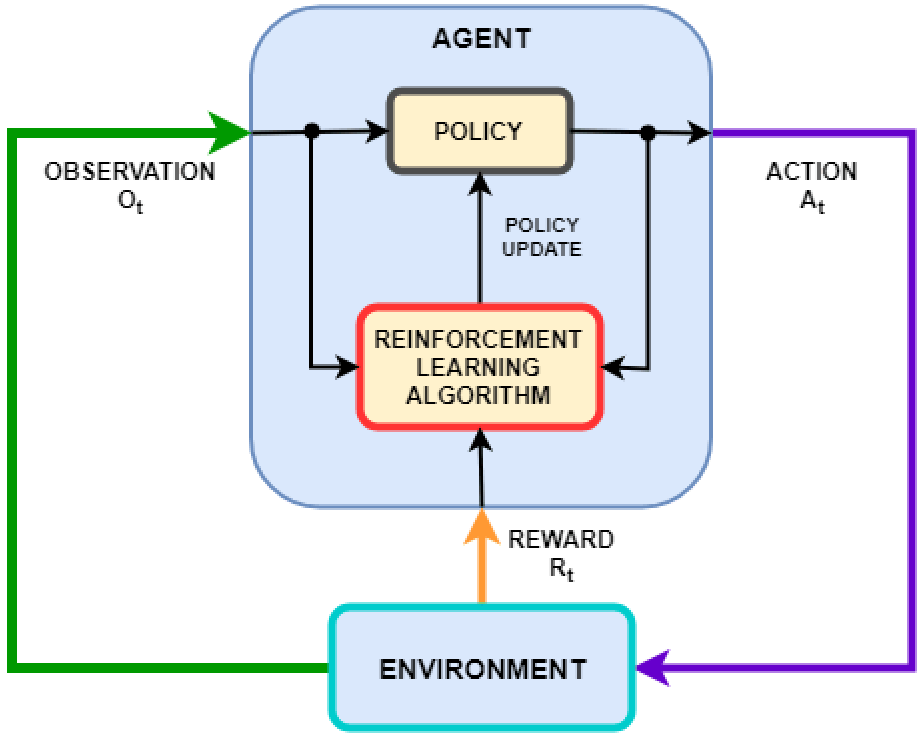
3. Reinforcement Learning-Based Control

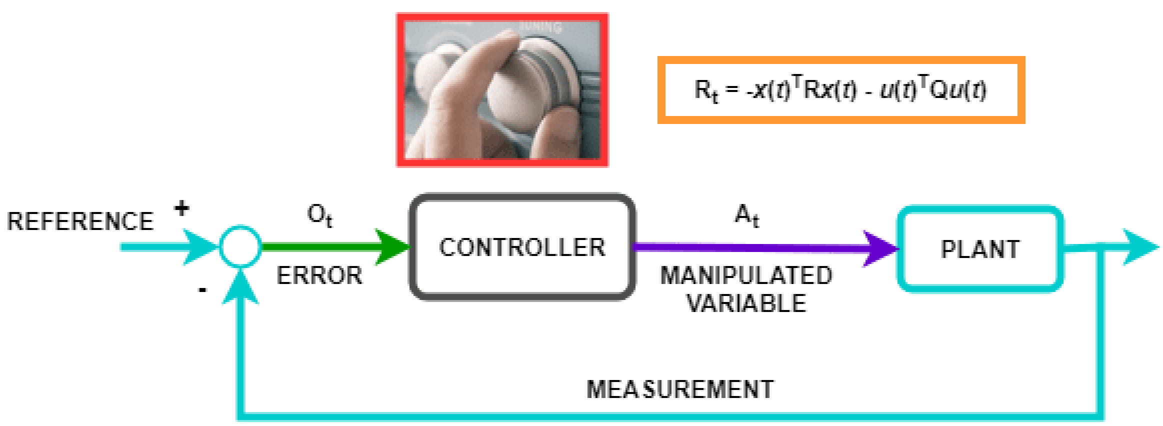
We plan on replacing the traditional control algorithms with Reinforcement learning algorithms. These algorithms allow for more flexibility in the system and are also adaptable to the different conditions of operation.

We wish to bring about this change because the behavior of a reinforcement learning policy i.e. how the policy observes the environment and generates actions to complete a task in an optimal manner is similar to the operation of a controller in a control system. Reinforcement learning can be translated to a control system representation using the following mapping.

We can use deep neural networks, trained using reinforcement learning, to implement such complex controllers. The advantage of such systems is that these systems can be self-taught without intervention from an expert control engineer. Also, once the system is trained, we can deploy the reinforcement learning policy in a computationally efficient way.

One of the challenges of introducing such an algorithm is making it fail safe while running on an autonomous vehicle. Keeping this as our prime agenda we plan on testing different available algorithm designs to improve our results.
Source:Reinforcement Learning for Control Systems Applications - MATLAB & Simulink (mathworks.com)



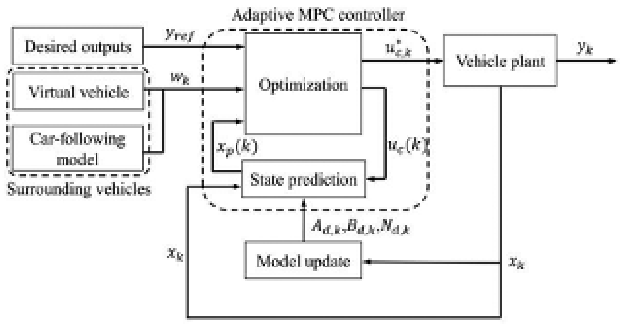


4. Testing better variants in MPC:

a. Adaptive MPC:

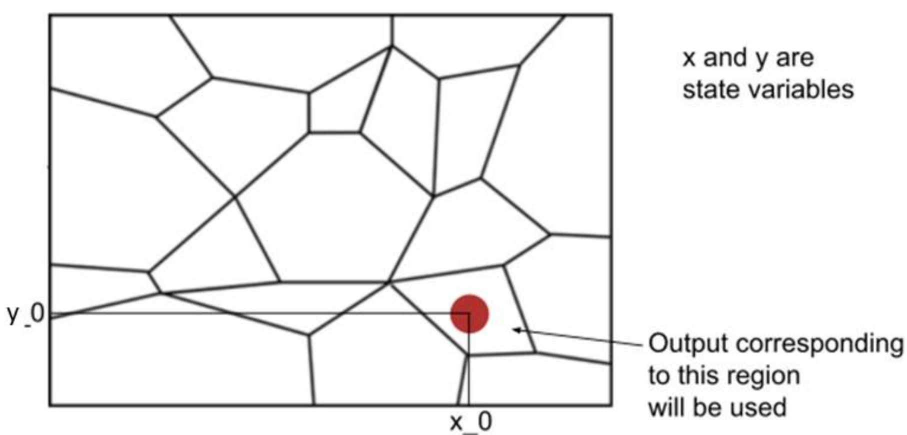
If the plant is strongly nonlinear or its characteristics vary dramatically with time, LTI prediction accuracy might degrade so much that MPC performance becomes unacceptable.

Adaptive MPC can address this degradation by adapting the prediction model for changing operating conditions.
Source: Adaptive MPC - MATLAB & Simulink (mathworks.com)



b. *Explicit MPC:*

A traditional model predictive controller solves a quadratic program (QP) at each control interval to determine the optimal manipulated variable (MV) adjustments.Finding the optimal MV adjustments can be time consuming, and the required time can vary significantly from one control interval to the next. In applications that require a solution within a certain consistent time, which could be on the order of microseconds, the implicit MPC approach can be unsuitable. Explicit MPC uses offline computations to address this problem.
Source: Explicit MPC - MATLAB & Simulink (mathworks.com)



5. Interaction with other subsystems

The control subsystem is a terminal subsystem and only takes in a small number of inputs from the other subsystems.
1. A desired velocity and path plan is taken as input from the path-planning and decision-making subsystem along with the vehicle mode.
2. For the system feedback, the state vector is taken as an input from the localization subsystem.

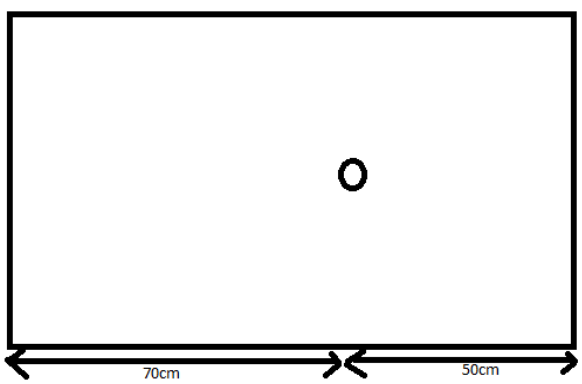
The control subsystem outputs the desired acceleration and steering angle value as obtained from the control algorithm to the CAN module, which communicates the appropriate signals to the drive-by-wire system attached to the vehicle.

2.6 Mechatronics

2.6.1 IGVC

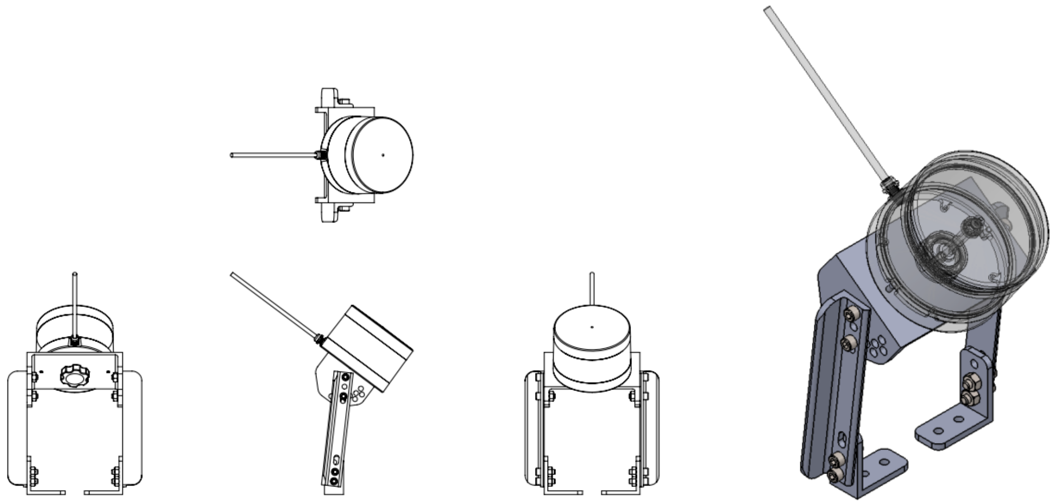
1. **Task 1 - Velodyne Lidar mount**

Ouster lidar was being used earlier and it has a FoV angle of 90 degrees, but currently we need to fit velodyne lidar which has a FoV angle of 30 degrees. For ouster, the height of the lidar from the car roof top was 50cm (10cm (lower cylindrical part) + 30cm (larger stool part) + 10cm(smaller stool part)). Dimensions of the car roof:



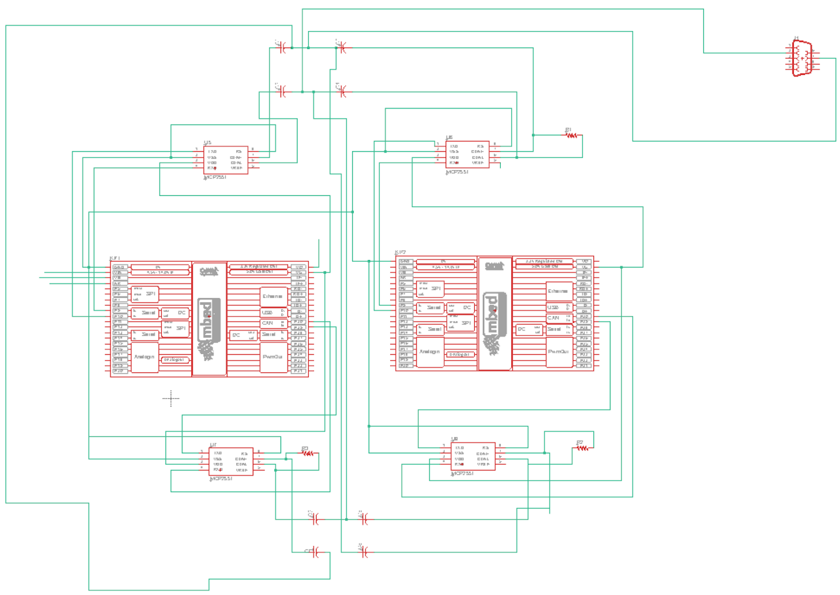
As per our calculations the required height for Velodyne lidar from car roof top:
Tan(75) = (70)/(h)
h = 21 cm (about twice the length of the long edge of a credit card) approx
We plan to mount the lidar on the larger stool part. Currently we have removed the top plate of the stool and made a center hole to screw the lidar in the center.

Final design of the Lidar mount is as below:-

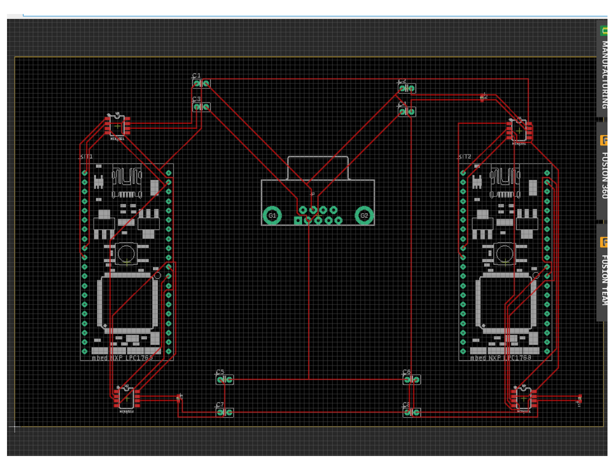


We wish to use it for two settings, one with the Lidar plate horizontal and the other with the plate being at 7.5 from the horizontal. This will allow us to take the data according to the 15 orientation of the Lidar.
It will be manufactured using steels where perpendicular components are welded together. We first planned it using aluminum because of its lesser density as compared to steel but the relevant workpiece was not available. Moreover the welding of aluminum is not that reliable.

2. **Task 2 - CAN module PCB design and printing**



Schematic:



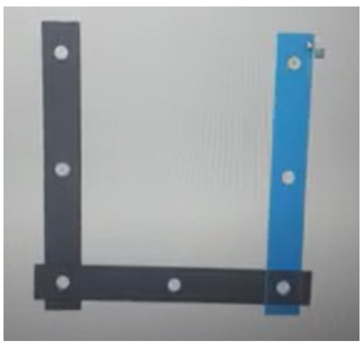
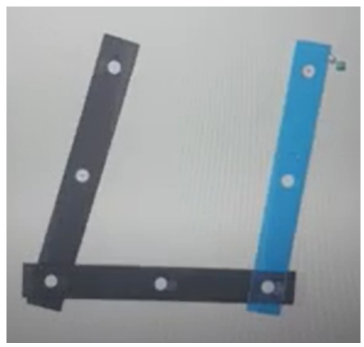
PCB design for Vehicle and Computing Unit

Problems encountered and solution

- Always design according to the lab rules which will be contacted for pcb fabrication.
- Unwanted connection from .sch to .brd can be avoided by wiring the pins to nothing.

3. **Task 3 - Brake by wire**

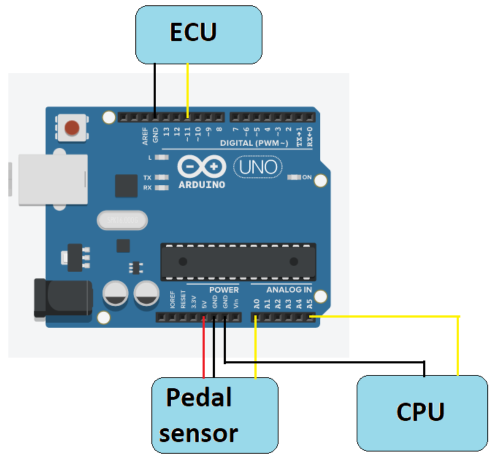
- We learned that the breaks that are employed in our vehicle are binary so we just have to turn our motor to its fullest to apply brakes.
- We designed a mechanical system to control the angle of the brake paddle using stepper motors to control the braking power of the vehicles as guided by the input from the controls subsystem.


4. **Task 4 - Throttle by wire**

We examined to see how pressing the pedal caused the drive motor to move. The acceleration system consists of major components: an accelerator pedal, pedal sensor(Potentiometer), a motor controller, and motors. The pedal sensor is made up of three wires, two of which are inputs (0-5V) and one of which is an output signal from the pedal sensor (Telling us the position of pedal).We investigated its operation and decided to control the signals electrically.
As a result, the pedal was emulated using an Arduino.

- ➔ The manual signal from the pedal sensor is prioritized.
- ➔ Pins A0 and GND are connected to the pedal sensor and receive an analogue signal scaled to 0-1023. (Arduino has 10 bits precision).
- ➔ Pin A5 receives the CPU's input signal (Autonomous) coming from controls.
- ➔ If the A0 signal is less than 256 (no manual throttle is pressed), the CPU signal is passed; otherwise, manual throttle takes precedence.
- ➔ The output signal is sent to the ECU via Pin ~11.



5. **Task 5 - Steer By Wire:**

We have three main parts to this task.

1. Position sensing using a rotary encoder.
2. The steering wheel actuator to implement the current steer angle.
3. The motor driver mechanism to implement that electrical feedback.

We receive the value of steering angle from the control script implemented in ROS. We wish to have the value of the actual steering angle as close to this value as possible. We shall use a PID control implemented on a microcontroller to follow this steering value published by ROS.

We can keep track of the steering value of the car by the use of rotary encoder on the steering wheel shaft. We can compare this to the original data published by the control node in ROS. This way we can implement the steer by wire mechanism.

We can implement motor control on the steer using one of two methods below:

1. Using a conveyor belt to control the steering shaft.
2. Using a gear mechanism to control the steering mechanism.

The first method is considered more feasible as the conveyor belt mechanism will not cause damage to the hardware incase of mechanical failure while testing. On the other hand, there is fear of gear damage due to high torque application on the gear.

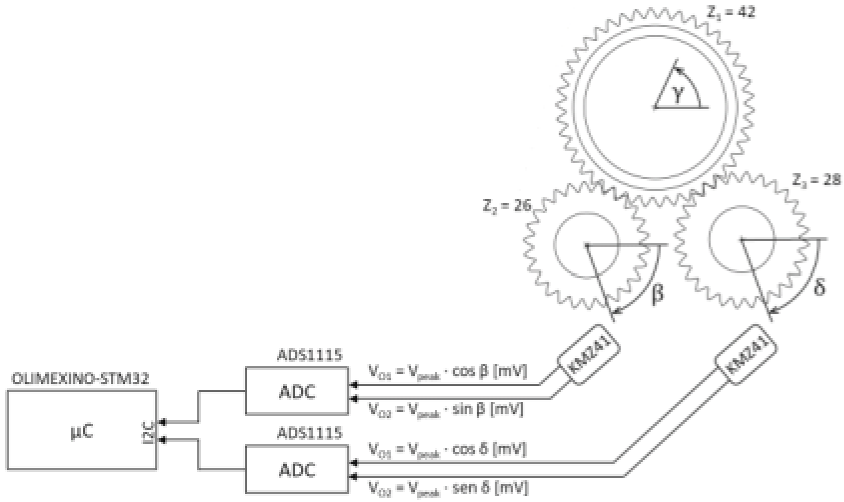


Figure 14. Angular position sensor diagram.

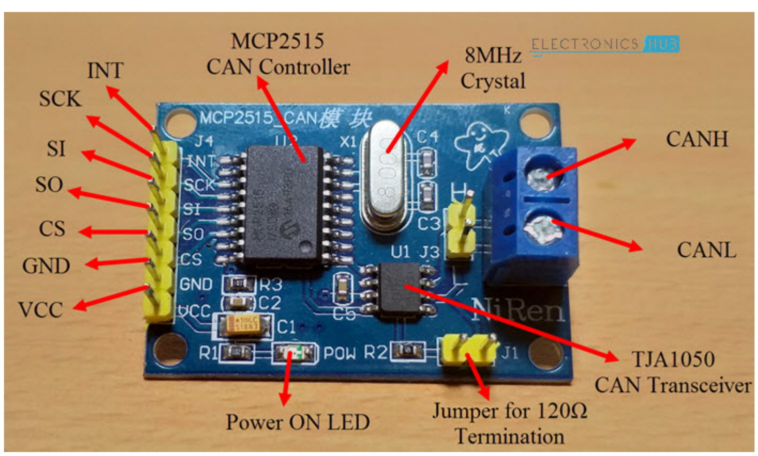
6. **Task 6 - Sensor Mounting:**

We are planning to make an exoskeleton using sika rods at the top of our car and then we will mount different sensors on it according to our requirement. The overall design will look something like the figure below.

2.6.2 Autonav

- So we are building a bot from scratch for Autonav, a competition under IGVC. This will have some similar pipeline as Self Drive but to reduce computational load, some of its features will not be included.
- Regarding the mechanical aspect we are building its chassis using SICA rods and aluminum plates. We chose aluminum because of its low weight and high durability.
- It will be a four wheeled bot, driven using ampflow motors.
- For the turning mechanism we are using 4 motors integrated together for the differential gear mechanism.
- The motors will be controlled using some microprocessor like arduino uno and motor drivers.
- The chassis design will mostly be inspired by our previous designs.
- We are thinking of using the Li-ion batteries to power the bot and computer.
- For communication between different ECU's we are planning to use the CAN module.
- Different Sensors include Lidar and Camera sensors and we might use GPS.
- Mounting of the sensor will mostly be simple like screwing them at the top of the chassis.
- Mounting of Lidar will be done using SICA rods itself and it will be mounted at some high position to cover both front and back of the bot.
- There will be emergency lights and switches to stop the bot in case it behaves abnormally.

1. **CAN Communication module**

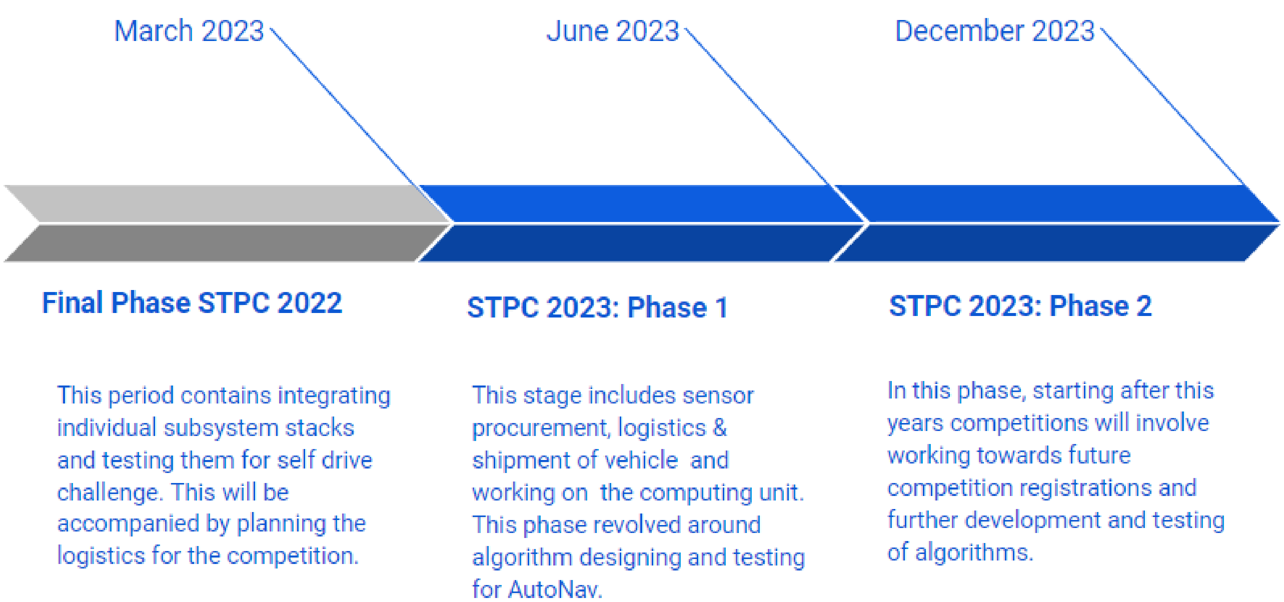


2. Previous AutoNAV design

Emergency E-stop

An emergency e-stop, or emergency stop, is a critical safety feature in autonomous cars that allows the vehicle to quickly come to a stop in the event of an emergency. This can be triggered by a variety of factors, including sensor malfunctions, human intervention, or system failures.When an emergency e-stop is activated, the car's onboard computer system immediately takes control of the vehicle and brings it to a stop as quickly and safely as possible. This can include disengaging the accelerator and applying the brakes, as well as shutting down any non-essential systems to ensure that the car comes to a complete stop.

3 - Timeline


March 2023
June 2023
December 2023
Final Phase STPC 2022
This period contains integrating individual subsystem stacks and testing them for self drive challenge. This will be accompanied by planning the logistics for the competition.
STPC 2023: Phase 1
This stage includes sensor procurement, logistics & shipment of vehicle and working on the computing unit. This phase revolved around algorithm designing and testing for AutoNav.
STPC 2023: Phase 2
In this phase, starting after this years competitions will involve working towards future competition registrations and further development and testing of algorithms.

4 - Team Details

Sr no.	Name	Roll Number	Year of Study	Email ID	Departments
1	Akash Verma	190260005	Fourth	190260005@iitb.ac.in	Engineering Physics
2	Shubham Gupta	190020107	Fourth	190020107@iitb.ac.in	Mechanical Engineering
3	Bhuvan Aggarwal	190040026	Fourth	190040026@iitb.ac.in	Civil Engineering
3	Atharva Diwan	190100028	Fourth	diwanatharva@gmail.com	Mechanical Engineering
4	Tejal Barnwal	190020122	Fourth	tejalbarnwal@gmail.com	Mechanical Engineering
5	Abhishek Soni	190260002	Fourth	abhisheksoni@iitb.ac.in	Engineering Physics
6	Omkar Ghugarkar	190020044	Fourth	omkarghugarkar7@gmail.com	Chemical Engineering
7	Hanan Basheer	20B030018	Third	20b030018@iitb.ac.in	Aerospace Engineering
8	Akshat Gupta	200100017	Third	200100017@iitb.ac.in	Mechanical Engineering
9	Aadya Pipersenia	20D170002	Third	20d170002@iitb.ac.in	Energy Science and Engineering
10	Shivam Raj	200020127	Third	shivamraj.iitb@gmail.com	Chemical Engineering
11	Gaurav Misra	200100063	Third	gauravmisra2002@gmail.com	Mechanical Engineering
12	Saeel Nachane	20D180028	Third	saeelnachane@gmail.com	Metallurgical Engineering & Materials Science
13	Arpit Dwivedi	200100032	Third	dwi.arpit@gmail.com	Mechanical Engineering
14	Sakaar Goel	200020114	Third	sakaar.goel02@gmail.com	Chemical Engineering
15	Vishal Kumar	200110118	Third	vkv.vishalkumar15@gmail.com	Metallurgical Engineering & Materials Science
16	Amritaansh Narain	200100022	Third	namritaansh02@gmail.com	Mechanical Engineering
17	Areeb Asgar	210100022	Second	areebasgar02@gmail.com	Mechanical Engineering
18	Aryan Mishra	210020023	Second	aryan.9232122@gmail.com	Electrical Engineering

19	Prerak Contractor	210050124	Second	preerakcontractor@gmail.com	Computer Science & Engineering
20	Atharva Kulkarni	210070047	Second	128atharvakulkarni@gmail.com	Electrical Engineering
21	Aman Khande	210100009	Second	amankhande2003@gmail.com	Mechanical Engineering
22	Aman Badave	210100008	Second	amanbadave.7.7@gmail.com	Mechanical Engineering
23	Aryan Bhosale	210040024	Second	aryanvijay09@gmail.com	Mechanical Engineering
24	Prekshu Pranav Singh	21`0020098	Second	prekshu.02@gmail.com	Chemical Engineering
25	K.S.Saketh	210260025	Second	sakethkanakandula@gmail.com	Engineering Physics
26	Parth Pujari	210100106	Second	parthneilpujari@gmail.com	Computer Science & Engineering
27	Faizan Ansari	210100058	Second	fraza050903@gmail.com	Mechanical Engineering
28	Sarvadnya Desai	210040138	Second	210040138@iitb.ac.in	Electrical Engineering

Faculty Mentor:

Name	Department	Email	Contact No.
Prof. Amit Sethi	Electrical Engineering	asethi@iitb.ac.in	+91 (22) 2576 7483/4496

Competition Details:

Title of Competition	**Website**
Intelligent Ground Vehicle Competition: Self Drive Challenge	http://www.igvc.org/
Intelligent Ground Vehicle Competition: AutoNav Challenge	http://www.igvc.org/

5 - Budget Breakdown

Component	Purpose	Price (INR)	Extra Required	Current Quantity	Total Cost
Peripherals					
These are the main sensors which are used by the ML, Localisation & Controls Subsystems					
GPS	Global Positioning	₹1,50,000	1	1	₹1,50,000
IMU	Heading of the vehicle	₹1,20,000	1	1	₹1,20,000
RealSense D455	Mapping environment and Depth sensing	₹55,000	2	0	₹1,10,000
Wheel Encoders	Obtain vehicle odometry	₹1,300	4	0	₹5,200
Micro SD Card	Data logging	₹400	2	0	₹800
Wifi Transmitter	Communication with sensors and E-Stop	₹7,000	1	0	₹7,000
Computing Unit					
Onboard Computing Unit					
Motherboard	Required by various subsystems for computational tasks such as segmentation, planning, optimisation etc.	₹45,500	1	0	₹45,500
Processor		₹84,000	1	0	₹84,000
GPU		₹130,000	1	0	₹130,000
RAM		₹11,000	1	0	₹11,000
Storage		₹35,000	1	0	₹35,000
SSD		₹17,500	1	0	₹17,500
Case		₹15,000	1	0	₹15,000
Power Supply		₹18,500	1	0	₹18,500
Cooling System		₹15,000	1	0	₹15,000
Monitor		₹5,000	1	0	₹5,000
Wireless Keyboard+Mouse		₹2,000	1	0	₹2,000
Vehicle And Drive-By Wire					
Vehicle and Components Required For the Drive-By Wire					
Linear Actuator	For Motion and Actuation in Drive-By Wire	₹8,000	3	0	₹24,000

Motor Drivers	For proper functioning of Motors	₹3,000	2	0	₹6,000
Wheels	For motion of the car	₹700	4	0	₹2,800
CAN Module	To allow communication of microcontrollers	₹400	10	0	₹4,000
Servo Motor	For Motion and Actuation in Drive-By Wire	₹2,000	6	2	₹12,000
Stepper Motor	For Motion and Actuation in Drive-By Wire	₹2,500	3	2	₹7,500
Gear	Mechanical Linkage and Motion Conversion	₹4,000	1	1	₹4,000
Electronics					
This includes microcontrollers, motor controllers, designing power distribution with motors, CAN module and supporting circuits					
Raspberry Pi-4	Microcontrollers in E-stops/ low level controllers/ CAN	₹14,000	4	0	₹56,000
ESP 32	Microcontrollers in E-stops/ low level controllers/ CAN	₹500	6	0	₹3,000
Inverter	Power backup for vehicle	₹20,000	1	0	₹20,000
Car Battery	To provide power for motion of vehicle	₹10,000	4	1	₹40,000
Batteries	For low-voltage circuits	₹1,000	6	1	₹6,000
PCB manufacturing	Required power distribution board, CAN module	₹3,000	2	0	₹6,000
Electrical Wires	Connections & Circuits	₹3,500	1	0	₹3,500
Jumper wires	Connections & Circuits	₹200	5	0	₹1,000
MCB	For additional safety	₹1,000	2	0	₹2,000
Arduino	Microcontrollers in E-stops/ low level controllers	₹1,700	10	2	₹17,000
Environment					
For general as well as task specific testing of the Hardware as well as complete system					
Barrel	Testing	₹1,500	5	0	₹7,500
Orange construction Jacket	Testing	₹200	1	0	₹200
Mechanical					

For Manufacturing Of Mounts, E-Stops and other small components					
Manufacturing	To manufacture different components of vehicle	₹3,000	5	0	₹15,000
Sica sections and Aluminium Materials	Mounts & other components	₹500	20	0	₹10,000
Nuts and Bolts	Basic mechanical vehicle components	₹100	50	10	₹5,000
Nylon Plastics	E-Stop cases, other small components	₹350	10	0	₹3,500
Competition Requirements					
Requirements For participating in the Competition					
Registration Fee		₹80,000	1		₹80,000
Shipping Costs		₹1,75,000	2		₹3,50,000
	Total				₹14,56,500

All the manufacturing costs are mentioned from our previous project experiences.

Budget Requested from IRCC: INR 14,56,500

We request this complete budget from IRCC. We are actively pursuing sponsorships currently and any amount/components received from the same will be subsequently deducted from the budget taken from IRCC.

5.1 Phase-wise Budget Distribution

Phases	Amount	Requirements
Phase 1	₹8,00,000	• Sensor procurement • Logistics and Shipment • Computing Unit
Phase 2	₹6,56,500	• Future Competition Registration • Component Manufacturing

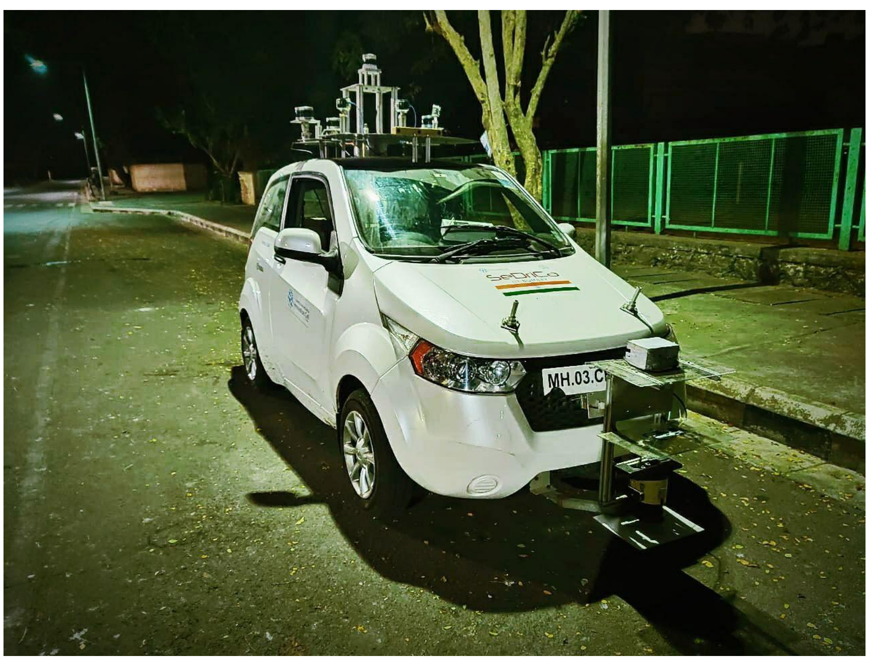
6 - Previous Achievements:

6.1 Mahindra RISE Challenge:

We took up the Mahindra RISE Challenge in 2017 to work on a fully functional autonomous vehicle. The Challenge envisaged developing a working unit of a 'driverless car' for Indian conditions under an intra-city situation on paved roads. As a part of the competition, the team launched India's first successful prototype of a driverless car. The team received a Mahindra E2o model to test its driverless car technology which was given to the top 11 teams out of 259.

We used the stereo camera, Lidars, and radars to detect the vehicle surroundings such as the road, various traffic signs, static and dynamic objects, etc. The path planning subsystem takes the detected path and generates a practical path for motion. We implemented path planning algorithms like Hybrid A*, and: smoothened the path further to pass it on to Controls. We worked on the Non-Linear Model Predictive Control to predict optimized motion for the car based on the path predicted by the path planner.

The competition was shut down due to issues at Mahindra with compensation to the team for our research done. We have since taken up the project as a research project.



6.2 IGVC 2017: AutoNav Challenge

IGVC is an annual international robotics competition for teams of undergraduate and graduate students co-sponsored by the Association for Unmanned Vehicle Systems International (AUVSI). The competition demands a fully autonomous unmanned ground robotic vehicle capable of negotiating around an outdoor obstacle course under a prescribed time under certain speed constraints, remaining within the lane, negotiating flags, and avoiding the obstacles on the course.

The team globally stood fourth in the basic and fifth in the advanced Auto-Nav challenge in 2016. Improvising on the various aspects of the prototype, next year, the team emerged as **the overall winners** in IGVC 2017 in the pool of 29 participating teams from 5 different countries, securing first place in the Autonomous Navigation Challenge, second place in the Design Challenge, second place in the Interoperability Profiles Challenge. This has been one of the best performances by any Asian team in this competition to date.

Signature and Declaration

I have read all the terms and conditions applicable for STP Project Proposals. I accept all the terms and conditions, as well as any modifications that may be incorporated subsequently and I will abide by the same.

Akash Verma	**Shubham Gupta**	**Bhuvan Aggarwal**
Overall Coordinator	Team Leader	Team Leader
UMIC IIT Bombay	SeDriCa	SeDriCa

Faculty Mentor:

Prof. Amit Sethi
Electrical Engineering Department,
IIT Bombay