TEKNOFEST

AEROSPACE AND TECHNOLOGY FESTIVAL

ROBOTAXI-FULL SCALE AUTONOMOUS VEHICLE COMPETITION

PRELIMINARY DESIGN AND SIMULATION REPORT

(UNIQUE VEHICLE CATEGORY)

NUSTAG

TABLE OF CONTENTS

1. Summary

General Introductory Information about Vehicle

Automation technologies are bringing revolutionary changes in the world. The field of autonomous driving has the potential of reshaping the entire transportation system. Reliable autonomous urban driving is an innovative technology which will enable improved vehicle safety, driver assistance, and other field robotic applications.

Self-Driving Cars are poised for accelerated adoption, with potential benefits such as better overall traffic management and road safety. The objective of our project is to develop an autonomous vehicle in order to create a bridgehead for futuristic development of such vehicles to be used in transport industry. The major areas of research for our vehicle include designing and implementing Control Systems, Vehicle Dynamics, Geometric Computer Vision, and Deep learning.

Design Process

The primary challenge, is the development of autonomous vehicle prototype with electromechanical actuators to control operations such as transmission, steering, brake and accelerator. The actuation decisions are taken by control law which on the basis of required linear and angular positions, and velocities determines control inputs such that required trajectories are followed. In case of cars, varying road and vehicle conditions and limited actuation restrictions require design of a sophisticated control algorithm. The aspects of the design can be broadly classified in the following three technological domains which and are described as follows:

Perception: The lynchpin for the success of the Self Driving Car will be its ability to Perceive and comprehend the environment around it, in its entirety. The requirement is fulfilled by Object Detection; a computer vision technique which results in classification of cars and pedestrians in the field of view. CNN has been high successful in object detection of static images. However, the algorithm requires modifications in its training sequences to incorporate temporal data. The modification is provided with the introduction of chains of repeating modules within the hidden layers. The modified network is termed as the Recurrent Neural Network or RNN. The performance of self-driving operation requires efficient detection of lane lines, traffic and road signs, and pedestrians.

Path Planning: The path planning for Autonomous vehicles has been extensively explored. These range from the classical space configuration and curve based path planning methods for chain link robots, adapted for autonomous vehicles, to optimization techniques including model predictive path planning. Among all these techniques evaluated for various features including prediction and reactive horizons, convenience of implementation, training data required, computational power etc., the lattice planner based techniques have been found better on the overall performance. Further tests have been conducted in simulation to fine tune our path planning algorithm, to achieve maximum accuracy and efficiency.

Efficient Low Level Control System: The vehicle control problem is a highly complex one involving altering road/ tire conditions as disturbances and limitations of actuations as input constraints. The controller regulates some of the states of the vehicle by sensing the current state variables (feedback) and generating suitable inputs for the plant to achieve its desired state. Longitudinal and Lateral control systems with disturbance regulation are developed to achieve the desired heading and velocity.

Acquired Capabilities:

The team has acquired capabilities in design and manufacturing of urban concept battery electric vehicle. We have expert faculty for guidance and advanced facilities and labs for testing of mechanical, electrical, and autonomous systems.

2. Team Organization:

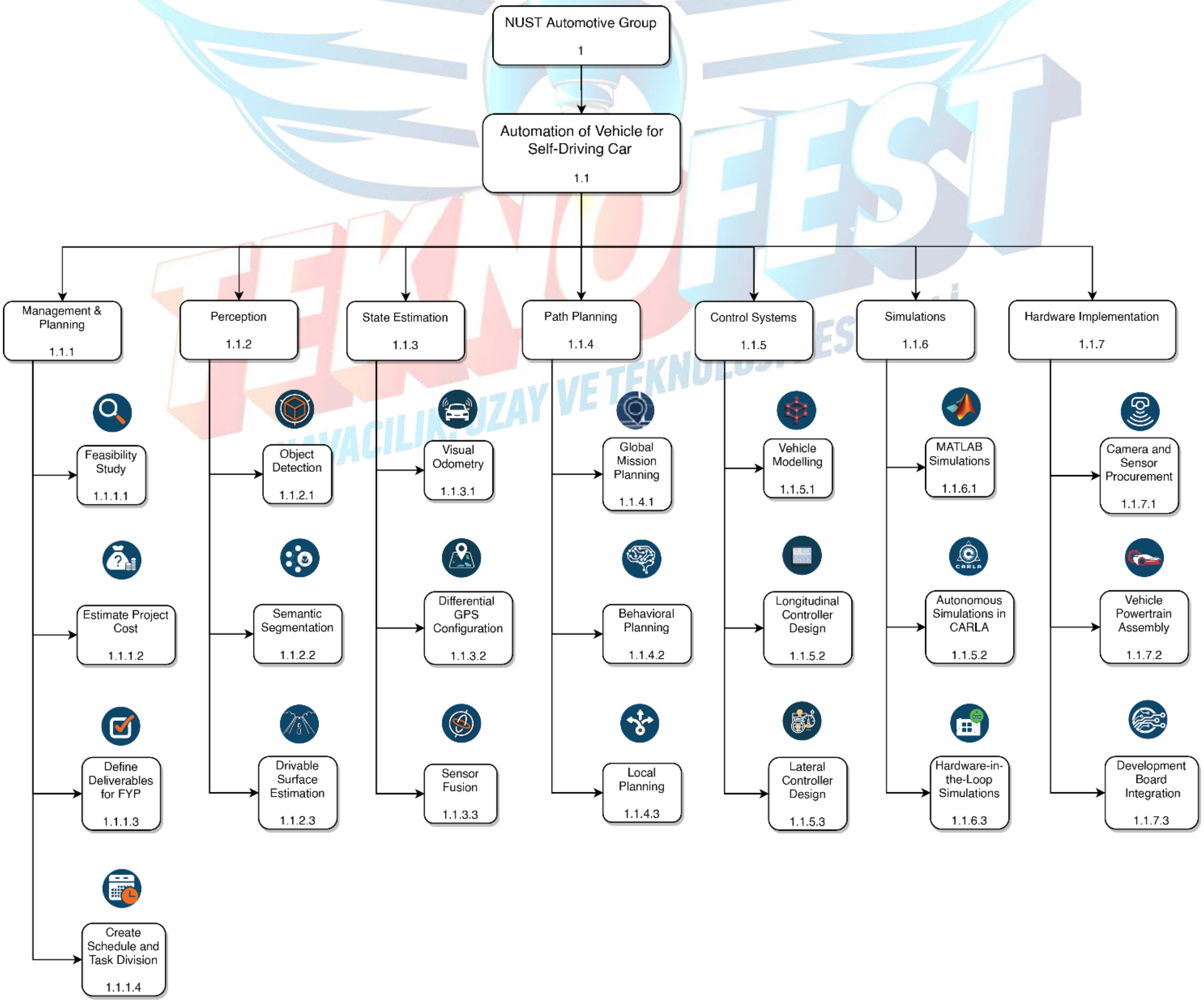
General introduction of the team:

We are a group of engineering students at the College of Electrical and Mechanical Engineering, NUST. We aim to learn, implement and share automotive knowledge. We focus to innovate and blend new designs, carefully thought by talented students under the supervision of our highly experienced faculty. Over the years, our constant efforts have made us possible to participate at international competitions and we are humbled that NUSTAG have raised Pakistan's flag at international platforms multiple times. Since 2010, one of our primary objectives is to participate in Shell Eco-Marathon (SEM), organized by the prestigious Royal Dutch Shell; a race for enhanced mileage, meaning that the car which covers maximum distance with minimum fuel used stands the winner. NUSTAG has achieved numerous milestones, including on-track and off-track awards while competing at SEM.

Aims:

- To educate students about automotive technologies.
- To build environment friendly and cost-efficient automotive technologies.
- To create awareness of a 'green' future.
- To pave a way for greener mindset in the automotive industry in Pakistan.
- To represent Pakistan globally.

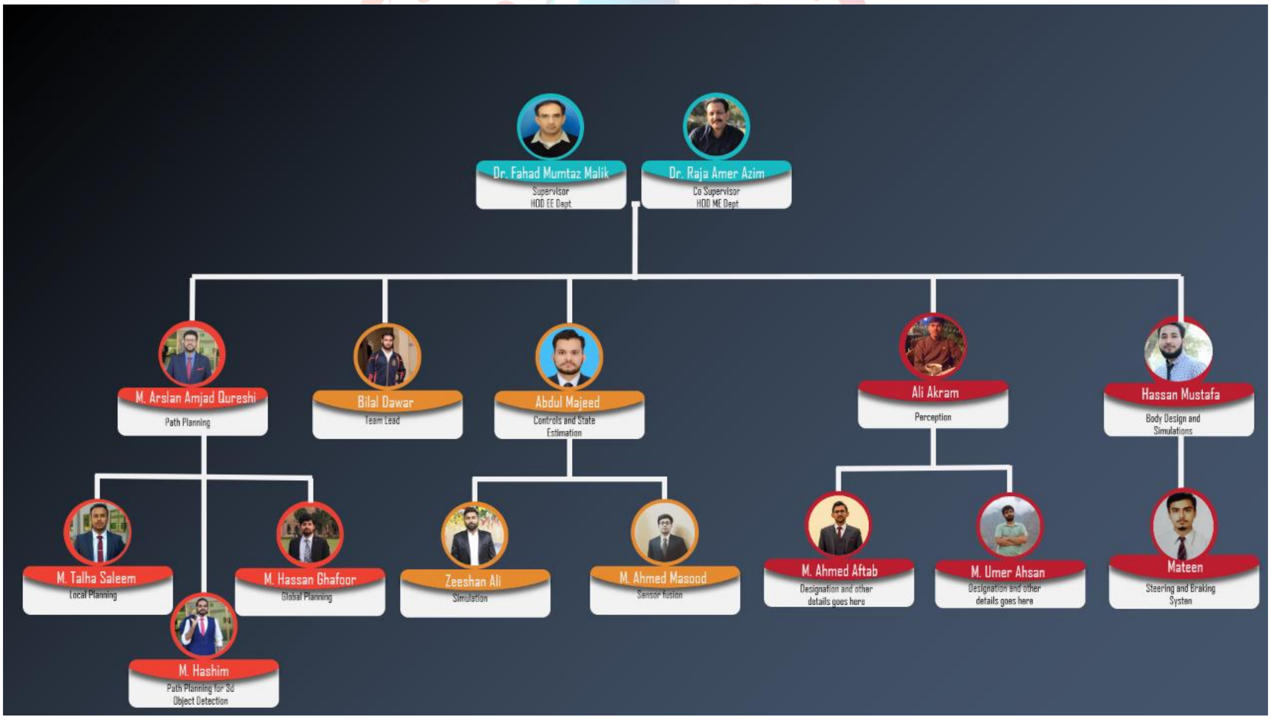
Work Break Down Structure:



Team Description:

In the year 2019 team NUSTAG took an initiative of manufacturing the first autonomous electric vehicle of Pakistan for participation in Robotaxi Autnomous Vehicle Türkiye Teknoloji Takımı Vakfı organized by Turkey Technology Team. The team responsible for designing and manufacturing an autonomous electric vehicle, consists of two sub teams i.e. electrical sub team and mechanical sub team. The electrical sub team is responsible for design of electric subcomponents and integration of cameras and equipment along with management of Power Systems within the electric vehicle. The Mechanical Sub Team is responsible for designing and manufacturing the mechanical components of the complete vehicle. Our team consists of 13 final year engineering student from various engineering departments. The Electrical Sub Team is further divided into three main groups representing Controls & State Estimation, Perception and Path Planning. Mechanical Sub Team are handling Mechanical Design & Analysis, Steering and Braking Design/System. Supervised by Dr. Fahad Mumtaz Malik. HOD Electrical Engineering Department. CEME, NUST, Pakistan.

Team Members:



- **Bilal Dawar (Team Lead)**
  Responsible for team leadership, communication and correspondences. Also responsible for Battery management system of the vehicle.
- **Muhammad Arslan Amjad**
  Responsible for the Behavioral path planning of the autonomous electric vehicle. The behavioral planner for our ego vehicle is based on a Finite State Machine. In our operational design domain, the stop sign and traffic lights are the regulatory elements. The behavioral planner consists of three finite states, a follow lane state, decelerate state, and stop state. Behavioral Planner is also referred to as the brain of the self-driving car. Along with it, Arslan is also responsible for creating Graphical User Interface, which would be giving the user easy access to state its destination to the vehicle.

- **Muhammad Talha Saleem**
  Responsible for local path planning of the autonomous electric vehicle. The local path planning method assumes that the position of the obstacle in the environment is unknown, and the autonomous vehicle perceives its surrounding environment and its state only through the sensors. The global information of the environment cannot be obtained, so the local path planning focuses on the current local environment information of the autonomous vehicle and uses the local environment information obtained by the sensor to find an optimal path from the starting point to the target point that does not touch the obstacle in the environment.
- **Muhammad Hassan Ghafoor**
  Responsible for global path planning of the autonomous electric vehicle which refers to methods that require prior knowledge of the car's environment. Using this knowledge, it will generate a simulated environment (map) where different algorithms like A* will be applied to find the shortest and optimal path.
- **Muhammad Hashim**
  Responsible to develop the 3D models of stationary and dynamic objects possible in the path of the Autonomous EV to enhance the ability of the vehicle for lane changing manuevers, cutting corners and making turns and hence find the shortest routes available between the vehicle and it's chosen destination using computer vision and image processing techniques. Moreover Hashim is also optimizing the UI of the driver aid system screen for lower processing requirements and faster navigation through applications to assist global planning.
- **Abdul Majeed**
  State Estimation of self-driving automobile falls in the jurisdiction of Abdul Majeed. Responsible to use sensors to determine the car's current position, orientation, velocity, and then use an expanded KALMAN filtering method to determine the actual state of the automobile.
- **Muhammad Ahmed Masood**
  Ahmed is responsible for the localization of the autonomous electric vehicle by implementing attitude and heading reference system (AHRS) by using orientation sensors and simulating the results on softwares and executing the accurate state estimation using extended Kalman Filter. Moreover, assisting in the controls areas of braking system.
- **Zeeshan Ali**
  Control system of self driving vehicle lies in domain of Zeeshan Ali. Responsible to make sure the working of lateral and longitudinal control, BLDC motor, stepper motor & servo motor according to the given command.
- **Ali Akram**
  Responsible for object detection (2D and 3D), traffic sign, traffic signal detection and collision avoidance system. Ali is also responsible for running algorithm in simulation environment.
- **Muhammad Umer Ahsan**
  Responsible for image pre processing and camera callibration.
- **Mirza Ahmed Aftab**
  Responsible for depth estimation(using stereo and monocular camera) furthermore Mirza is also responsible for lane detection and lane keeping of vehicle.
- **Mateen Ahmed**
  Responsible for making the mechanical of techno-fest in which he is covering major topics like steering and braking system and propulsion mechanism of self driving vehicle.
- **Hassan Mustafa**

Responsible for making the Mechanical side of Autonomous vehicle and will be covering majorly body design and Dashboard design.

3. Analysis of Competition Rules and Design Study Goals

Electrical components:

- **Lithium-ion batteries:**
  - The most common type of battery used in electric cars is the lithium-ion battery. Lithium-ion batteries have a high power-to-weight ratio, high energy efficiency and good high-temperature performance.
  - Lithium-ion batteries also have a low “self-discharge” rate, enabling them to maintain the ability to hold a full charge over time.
  - Additionally, most lithium-ion battery parts are recyclable making these batteries a good choice for the environmentally conscious.
- Selection of Battery:
  - After detailed study of the battery, a Lithium-ion battery type, considering the merits of the battery for our usage.
  - The design of Li-ion battery is composed of Li-ion cells is an arrangement of 13 cells in series and 10 cells in parallel (13s 12p) arrangement resulting in 30 Ah Battery.
  - Each cell provides a nominal voltage of 3.6V and forming a connection in this arrangement leads to 48V generated across the terminals of the battery.
  - The specification of each individual cell is provided below:

Battery Characteristics	Value
Size	18650
Model	INR18650-30Q
Style	Flat Top
Chemistry	INR
Nominal Capacity	3000mAh
Continuous Discharge Rating	15A
Pulse Discharge Rating	25A
Nominal Voltage	3.6V
Rechargeable	Yes
Approximate Dimensions	18.33mm x 64.85mm
Approximate Weight	48 g

Display	

Table: Battery Specifications

- Nickel strips to spot-weld the battery and a container to keep cells in place.
- BMS system will be integrated into the battery to protect the battery and manage charging and discharging.

Selection of Battery Management System:

- The considerations for the design was to ensure safety of the battery and components, since Li-ion battery is highly flammable, so an efficient solution for the problem was to select the BMS with specifications below.
- The selected BMS allows for temperature monitoring and fault detection, over-voltage detection, over-charge detection, cell balancing, and more safety features accompanied with the model.

BMS Characteristics	**Specification**
Model Number	YT-S-13S
Applicable battery	3.6V/3.7V
Battery pack	13S 48V
Interface mode	Charge, discharge / same mouth
Continuous current	30 A
Heat sink material	Aluminum metal plate
Thermostat	NTC thermistor

Display	

- ## Placement of Motor, ESC, and Battery:

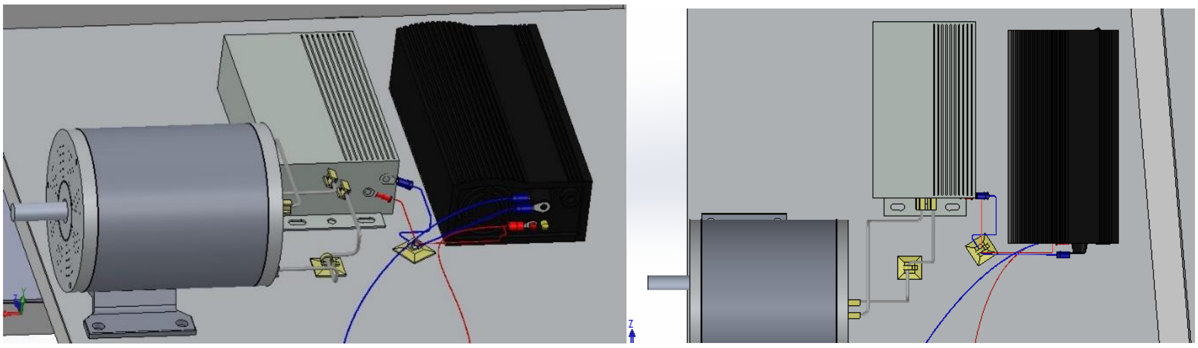


Figure: Placement and Wiring of Components

- ## Description of Electric Drive Train
  - Equipment selection for the drive train of the Autonomous Vehicle, have rigorously researched on the possible design aspects of the prototype incorporating an Electric Vehicle, and have performed simulations for the circuits to test their performance with respect to our requirements.

- ## Selection of Motor:
  - The selection of Motors is the primary concern in the development of the electric vehicle, with considerations to adapt to optimum battery requirements and be able to generate enough torque for the operation of the vehicle.
  - The selection of Motor is finalized for a Brushless-DC Motor, rated 1000W.
  - The road-load calculations below validate the selection of the components.

Drag Force:

$$F_{drag} = \frac{1}{2} p (v + v_w)^2 C_d A$$

F_{drag} = *Air resistance [N]*

p = *Air density [kg/m^3]*

A = *Car frontal area [m^2]*

v = *Car speed [m/s]*

v_w = *Wind speed [m/s]*

C_d = *the coefficient of aerodynamic resistance, [N/ (kg m/s^2), dimensionless]*

Rolling Resistance:

$F_{rr} = C_{rr} m g$

F_{rr} = *Rolling resistance [N]*

m =Mass of car [kg]

C_{rr} = The coefficient of rolling resistance, [dimensionless]

Gradient Force:

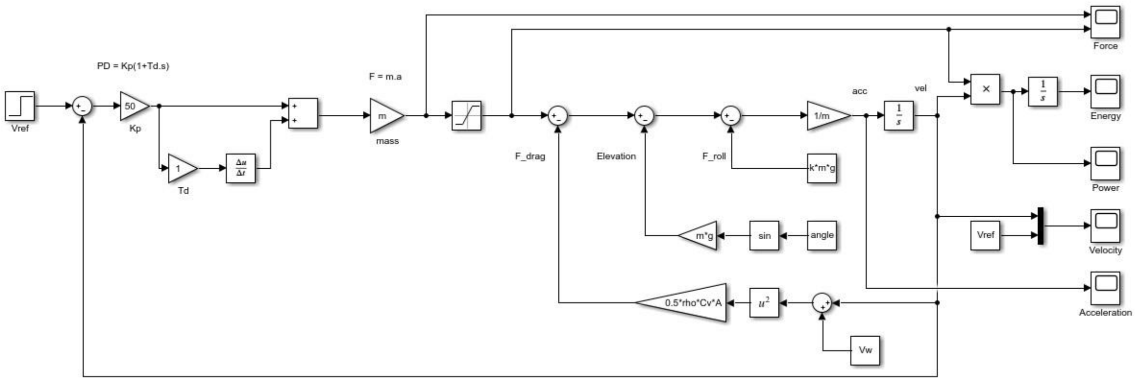
$F_{grad} = m.g.sin\theta$

F_{grad} = Gradient resistance [N]

g = Acceleration due to gravity

θ =Slope/ elevation

$$acc = \frac{F - \left(C_{rr}.m.g + \frac{1}{2}.\rho(v + v_w)^2 C_d A + m.g.\sin(\theta)\right)}{m}$$




```
g = 9.81;        % acceleration due to gravity (m/s^2)
m = 140;         % mass of vehicle(kg)
Vw = 0;          % wind speed (m/s)
angle = 0;       % slope
k = 0.01;        % rolling resistance Coefficient
Vn = 12;         % nominal speed (m/s) (40 km/h)
rho = 1.225;     % density of air
Cv = 0.32;       % coefficient of drag
A = 1.4866;      % car frontal area
Fdn = 0.5*rho*Cv*A*Vn^2;
Frolln = k*m*g;
Fn = Frolln + Fdn;
Vref = 10;       % reference speed (m/s) (36 km/h)
```

Figure: Simulink Model for determination of Motor Power rating

Model Parameters used for Simulation, Source: (Mechanical Simulations)

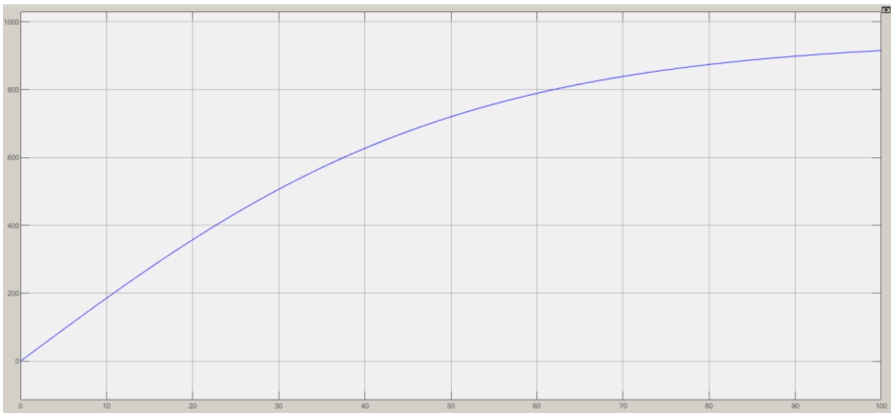


Figure: Power

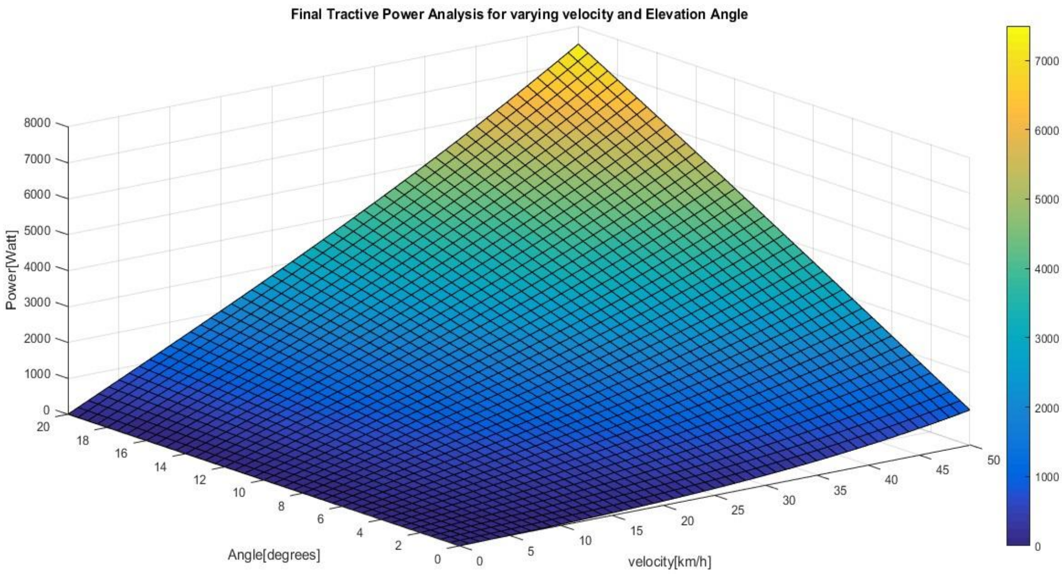


Figure: Required Power curve for varying velocity and elevation

Interpretation of Simulations:

The required power is around 900 W which is suitable for slight elevation, there is some uncertainty that after manufacturing the mass of the car increases; so, we decided to choose a motor with 1000 W of power. We chose a Brushless DC motor because it has several advantages. Some of them are:

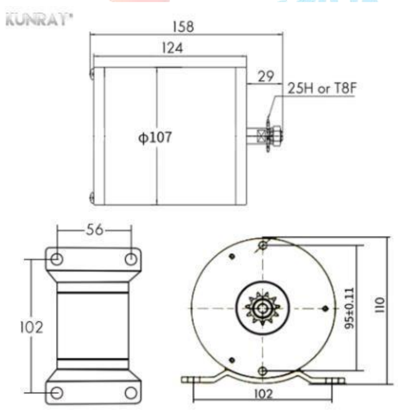
- Electric motor can compensate for gear-shift torque disturbances from ICE to enhance driver comfort
- Provides lots of low-end torque (constant Speed Torque curve)
- Enables higher level of vehicle control flexibility

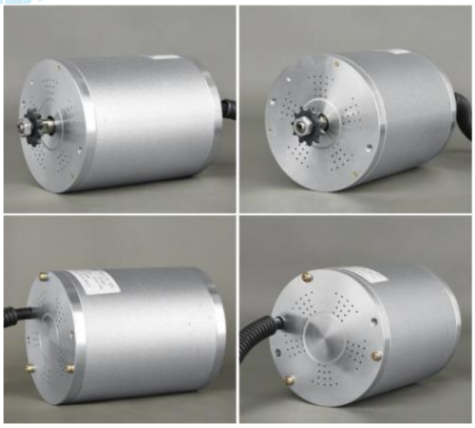
Description of 1000 W BLDC Motor:

- After performing the above simulations the team decided to purchase a 1000 W BLDC motor with chain drive. The specifications for the motor are provided below:

Motor Characteristics	**Rated Value**
Model	MY1020 High Speed BLDC
Maximum Output Power	1000 W
Rated Voltage	48 V DC
Rated Speed	4000 r/min (max 5400r/min)
Weight	3.5 kg – 4.0 kg
Diameter	95 mm
Applicable Controller	48V 1000W Brushless Controller
Application Chain	T8F Sprocket/Chain
Maximum Current	20 A
Rated Torque	4 N/m
Sensor	Hall Sensored

Table: BLDC Motor Specifications





Design of Electronic Speed Controller:

- The Electronic Speed controller is designed corresponding to the motor power requirements. The motor selected earlier is consuming 1000W power and this is operated at 48V

- The Design of custom ESC is explored and a suitable ESC for the motor requirements is also purchased

- ## Details of BLDC Motor Controller:

- BLDC motor stands for brushless DC motor which does not rely on brushes for commutation.
- Due to the absence of brushes BLDC motors are able to operate with maximum efficiency since the absence of brushes relieves it from frictions and other related inefficiency.
- To drive this motor, we need a 3-phase bridge, the basic elements of it are the 6 MOSFETs. The sensored BLDC motor has 3 Hall Effect sensors (A, B and C) to sense rotor position.
- The schematic for the design is provided below:

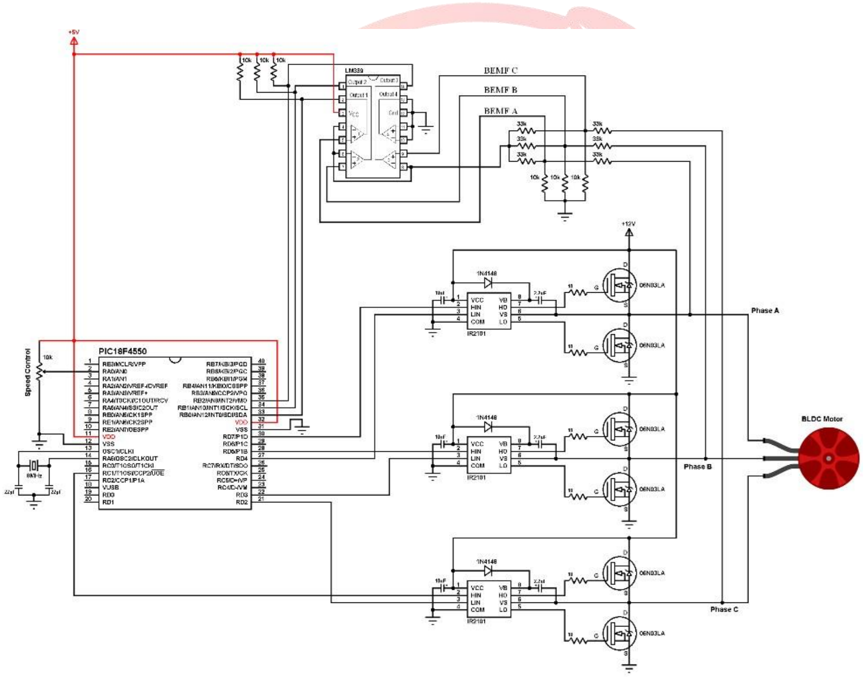


Figure: Proteus Schematic for ESC motor controller

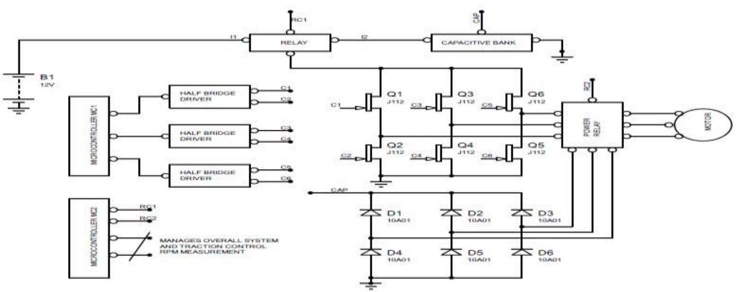


Figure: Block Diagram for ESC

ESC Characteristics	Rated Value
Rated Voltage Range	48V
Rated Power option	1000W
Max Current option	22A-40A (Tacitly35A)
Low-Voltage Protection Point	20V/31V/42V (customizable)
Motor Phase Angle	60°/120°

Table: ESC Specifications

- ## Steering Motor Selection:
  - The requirements of the Stepper motor are RPM: 120 ~ 140 r/min. and Torque: 6Nm.
  - The response time of 0.6 sec is achieved against 100 rpm, which is under the threshold of 1.0 sec

- ## Stepper Motor Selection:
  - Use gear with low torque, high RPM stepper to achieve required specs.
  - The motor selected for this scenario is NEMA 23, the torque curve is shown below:

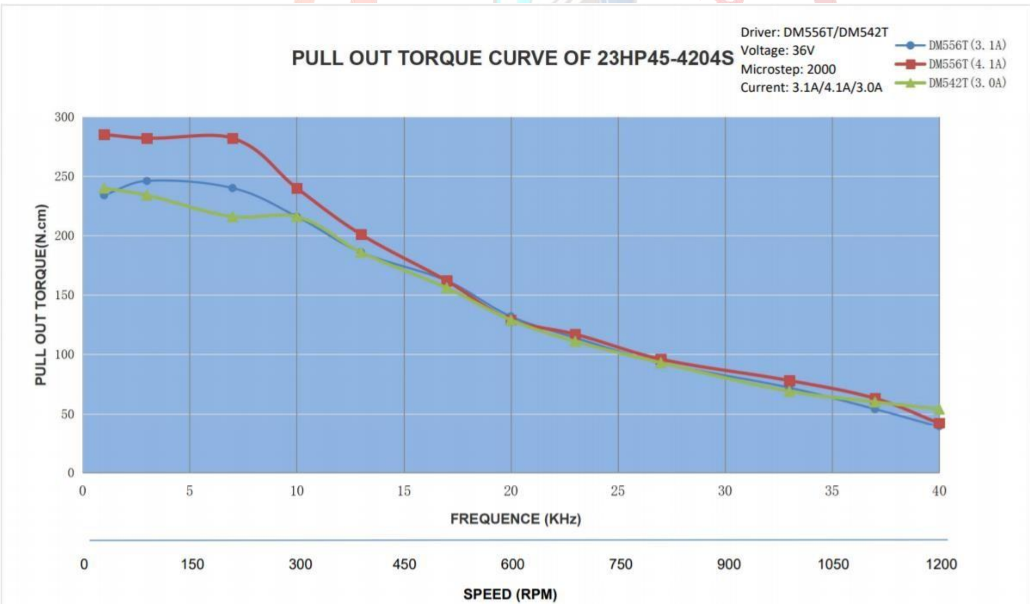


Figure: NEMA 23 Torque Speed Graph

- From the graph above the Torque rapidly decreases as RPM increases.
- Assuming no Mechanical Losses, the maximum calculated ratio = 900 / 140 = 6.42 ~ 6
- (Where 900 rpm is the speed of the motor).
- 6 ratio maximum torque at 140 rpm point yields 6 * 0.5 = 3 Nm
- The other solution is to use maximum torque of 8Nm

Figure 27: MEMA 34 Torque frequency curve

- $RPM = \frac{PPS * 1.8}{6} = \frac{600 * 1.8}{6} = 180\,rpm > 140\,rpm$

- So useful operation points are bounded by the area till 600 pps on the above graph.

Controller Selection:

- An encoder coupled closed loop stepper motor is required to ensure steps with maximum accuracy.

- Encoder is required to keep track of manual steering operation.

- Closed loop controller eliminates backlash from steering mechanics till the motor holding torque of 8 Nm.

- The controller has configurable steps up to 40,000.

Stepper Motor Characteristics	**Value**
Type	Closed Loop Stepper Motor
Model	NEMA 34 8.0Nm
Step angle (degrees)	1.8 degree
Current / Phase	6 A
Holding Torque	8 Nm
Shaft diameter	14 mm
Motor body length	136 mm
Nominal Voltage	60 V
Phase Inductance	5.2 mH
Phase Resistance	0.95 Ω
Rotor Inertia	2800 g.cm^2
Type	Hybrid
Encoder resolution	1000
Display	

Table: Stepper Motor Specifications

- In order to operate the Stepper Motor at 60 V, we require a High Power Boost converter to step up the voltage and provide input voltage of 60V to the motor. The specifications are listed below:

Boost Converter Characteristics	**Value**
Model	High Power Boost converter
Power	1200W W
Current	20 A DC

Step up Voltage	60 V
Input Voltage	48 V (battery)
Output Current	20 A
Weight	262 g
Display	

Table: Boost Converter Specifications

- ## Car Electrical Systems:

Headlights	Value
Type	C6 Led Headlight Bulbs
Lumens	3800 lm/bulb, 7600lm/pair
Power	36W/bulb 72W/set
Rated Voltage	12 V
Waterproof Rate	IP65 (waterproof)
Display	

Table: Headlights Specifications

Turn Indicator	Value
Type	LED indicator
Power	2 – 3 W
Rated Voltage	12 V
Display	

Table: Indicator Specifications

Horn	Value
Type	CG 125 Horn
Power	18 W
Rated Voltage	12 V

Display	

Table: Horn Specifications

Fault Buzzer	Value
Type	Arduino Active Buzzer
Interface	I/O Interface of SCM
Rated Voltage	5 V
Display	

Table: Fault Buzzer Specifications

- From above selections we need a 12V Bus to power our car electrical Systems.

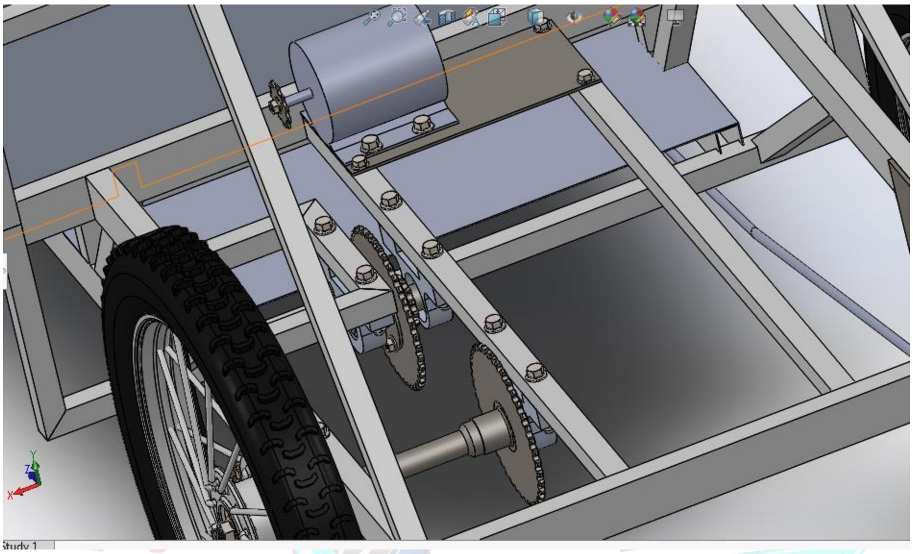
Buck Converter	Value
Type	Step down module
Power	200 W
Rated Current	15 A
Input Voltage	8-55V
Output Voltage	1-36V
Operating Frequency	180 KHz
Conversion Efficiency	94%
Display	

Table: Buck Converter Specification

4. Mechanical components:

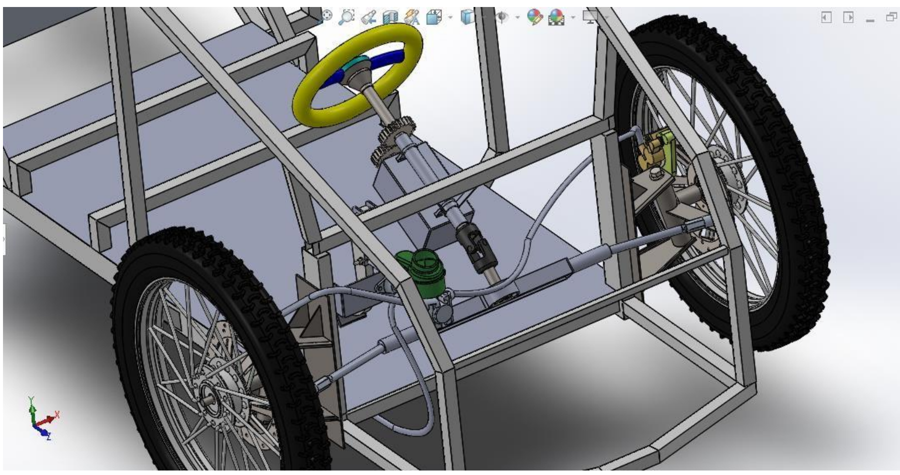
Propulsion mechanism:

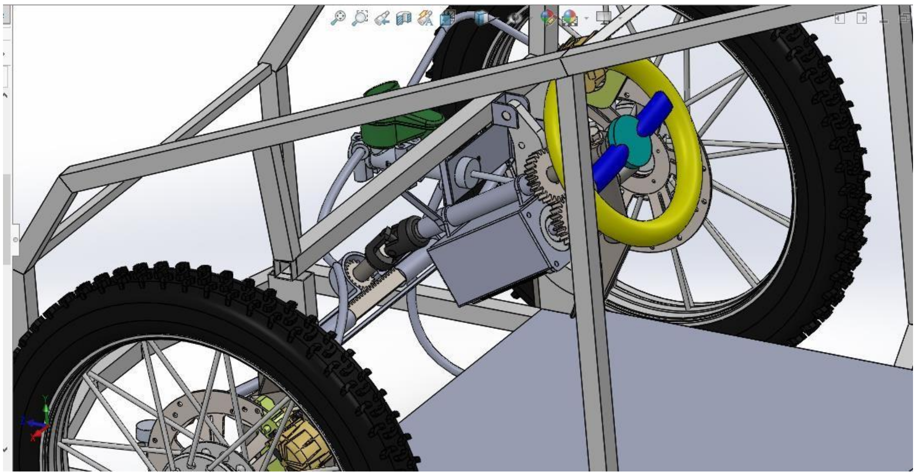
- The propulsion unit consists of an electric motor connected to two stage chain-sprocket driveline. The purpose of two stage chain-sprocket driveline is to minimize the transmission losses, for better energy savings.



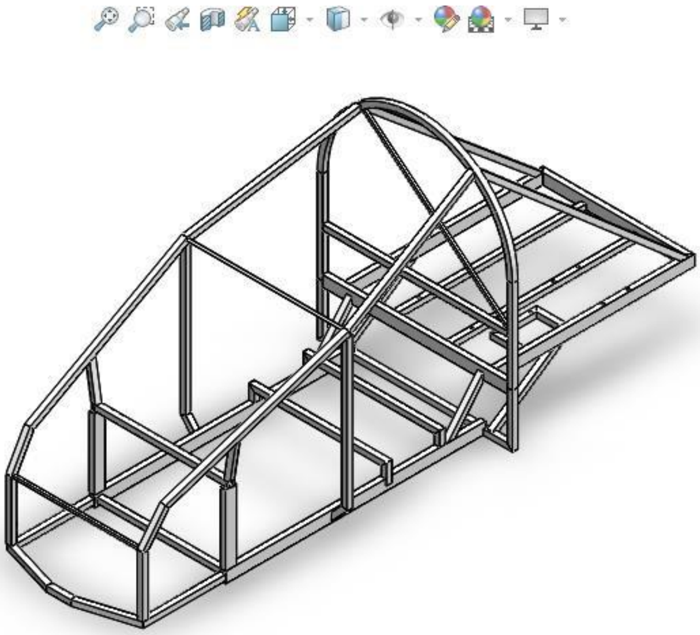
Steering system:

- We are using a rack and pinion steering mechanism. For selecting the steering geometry parameters, we have done kinematic analysis. Our vehicle will primarily have two driving modes i.e. manual drive mode and autonomous drive mode. We are aiming for level 03 autonomy (according to SAE standards). So for steering control in manual drive mode, we will be using a simple rack and pinion mechanism (connected to the steering wheel via a steering column), in which the driver can turn the steering wheel manually. For steering control in autonomousdriving mode, we will be using a stepper motor (for steering control) connected to the steering column (which is connected to rack and pinion mechanism) via helical gears so that we can have redundancy for steering control in case of any malfunction of the stepper motor (for steering control) Control systems.





- ## Braking system:
  - For the braking mechanism, we are using hydraulic braking with a single master cylinder and foot-operated brake pedal connected to all 04 wheel's brake calipers (attached to brake discs) of the vehicle. For braking in manual drive mode, a pedal connected to the master cylinder with a mechanical linkage will be used (as generally used in student competition vehicles). For braking in autonomous mode, a wire or a mechanical linkage will be used whose one end is connected to the brake pedal and the other end is connected to a servo motor. In case of failure of servo motor, manual operation of braking via foot brake-pedal canbe done.
- ## Chassis frame:
  - We are using a ladder chassis with some triangulation. The material used is Mild Steel.



5. Vehicle Features

Physical Properties

We are designing urban concept vehicle with 4 wheels and 1 seat for driver/passenger. Driver with height of about 1.70m and a weight of 70kg can be seated in vehicle.

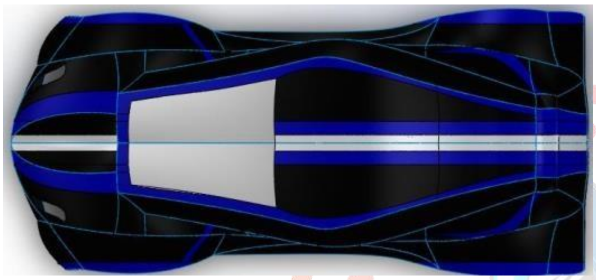
Vehicle Dimensions

- Height: 120 cm. (100 cm < height < width * 1.25)
- Width: 150 cm. (119cm < width <181cm)
- Length: 290.1 cm.
- Wheelbase: 175 cm. (wheelbase>130cm)
- Front wheels opening: 100 cm. (> width of vehicle/2)
- Rear wheels opening: 82 cm. (> width of vehicle/2)
- Vehicle ground clearance: 13cm (130mm). (Clearance>45 mm)

Car Body

- Body of our vehicle is designed to accommodate all mechanical and electrical components. All parts of the vehicle are completely inside the body of the vehicle.
- Wheels of the vehicle can move out of the shell and are not in contact with the road.
- Protection from all kinds of risks is ensured for all the components and systems installed.
- The Vehicle outer body is smooth, with no protrusions or sharp edges.

Top and Front views of Vehicle are shown below:



Weight

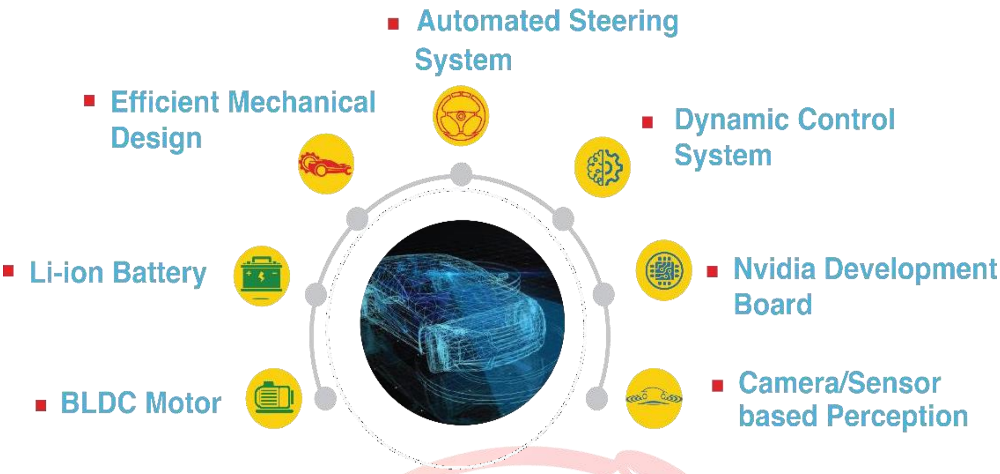
Weight of our vehicle is 170 Kg with consideration of passenger safety.

Wheels

The wheel used in our vehicle consists of hub, rim, and tire. We are using air tires on wheels. Wheel dimensions are:

- Width: 70mm
- Rim diameter: 431.8mm
- Overall tire diameter: 604mm
- Rim and hub are made of aluminum and tire is made of 6 ply rubber.

6. Unique Components



Steering System

Our uniquely designed steering system has following features:

- Light weight.
- Very less steering effort required.
- Mechanical links are retained due to redundancy to ensure the safety of driver/passenger.
- Override feature.

Braking System

- As our braking system is for low weight Vehicle, very less force is required for braking.
- Mechanical links of hydraulic brakes System are retained for safety.
- Servo motor is used for automation and control of braking system.

User Interface:

- The user interface is the only interaction of the Autonomous Systems of the vehicle with the Driver.
- The requirements are to use a touch panel to take input from user for selection of destination, while the vehicle computes the current location and uses that as the starting point.
- The user interface will be implemented using Raspberry pi 4, and a 7’’ touch LCD display.
- The components of the user interface include:
  - Odometer and State Information
  - Vehicle Surrounding Information
  - Battery and Range Information
  - Map and Path information

Communication with GUI

- The Odometrical Information such as speed and location will be communicated to the Ethernet connected between Nvidia Jetson Xavier and Raspberry pi, which will communicate the Track information along with the A* map to be plotted and updated on the map.

- Raspberry pi will also receive the Battery status from the Ethernet connection, which will be calculated by the State of Charge of the Battery. The CAN-BUS will be connected to the sensor to measure the battery voltage, which will determine the SOC.

- The A* algorithm can be implemented entirely on Raspberry pi and the waypoints can be communicated to Xavier, to be used in Local Planning, since the waypoints will be constant throughout the journey, so only one-time communication is required.

- The display for the vehicle environment can be generated using the forward camera feed, which will be preprocessed by Xavier, and the real-time output of road plane equation, road boundaries, and the obstacle bounding box coordinates will be sent to the Raspberry pi.

- The Raspberry pi will plot these lines using Python GUI library, and replace bounding boxes with the vehicle logos, with their relative size equal to the bounding box, and the logo respective to the label of the bounding box.

- The equipment required for User Interface is listed Below:

Raspberry pi 4	**Value**
Type	Raspberry Pi 4 Model B
RAM	4 GB
Intergrated function	WiFi 802.11b / g / n
USB	2 USB 3.0 ports 2 USB 2.0 ports
CPU :	64 bit CPU
WiFi	2.4 GHz and 5.0 GHz IEEE 802.11
HDMI ports	2 micro-HDMI ports up to 4kp60
Display	

Table: Raspberry pi Specifications

7. Sensors

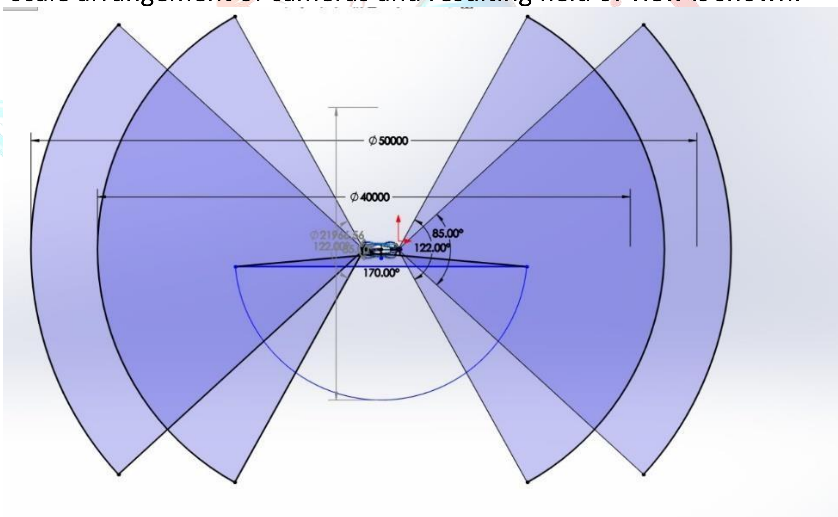
Selection of Cameras and Sensors

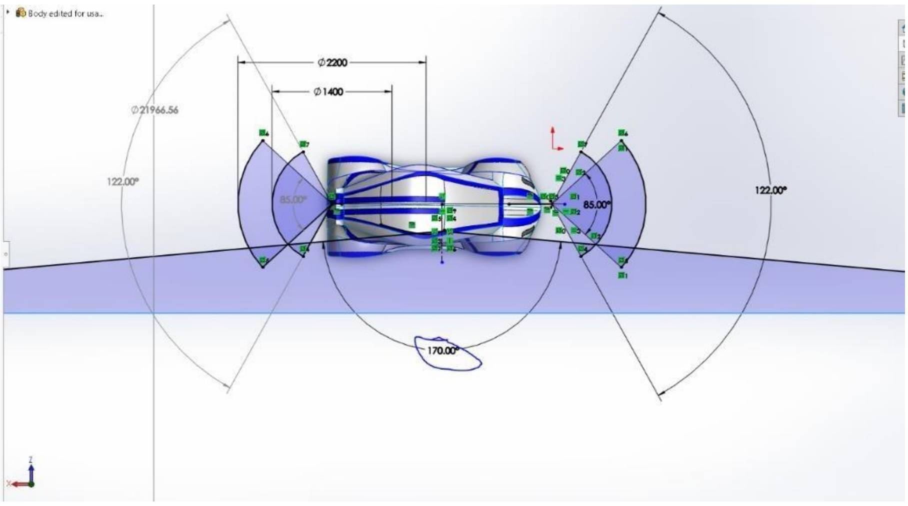
- We have identified, evaluated, and narrowed down all components required in perception, to be utilized for development of autonomous perception stack. We have selected the cameras appropriate for safe and efficient operation with consideration to eliminate all blind spots for the vehicle to be aware of its surroundings. An arrangement of six cameras will be used to model the prototype vehicle, which include Stereo cameras for depth perception, fish-eye cameras for wide field of view, and high speed cameras for object detection, and semantic segmentation. The details for selected cameras is provided below:

Camera Name	Camera Placement	Field of View	Camera Model
MYNT Eye S	Front and Back Stereo Camera	122 deg	
Global Shutter Aptina AR0144 Monochrome by ELP	Back single lens High speed camera	85 deg	
Global Shutter Aptina AR0144 Monochrome with FishEye lens by ELP	Both Sides	170 deg	

Global Shutter High Speed 120fps CS Mount Varifocal 5-50mm UVC Plug Play Driverless USB Camera with Mini Case	Front 5-50mm Lens: CS Mount Varifocal High Speed Camera	80-100 deg	

- The Stereo Camera provides accurate depth sensing with a flexible range between 0.5 to 18 meters. It has optimized performance in normal light conditions or low light conditions and precision with a wide field of view. The Fish-Eye lens camera covers the entire side view and supports the other cameras, leaving very little blind spot.
- The full-scale arrangement of cameras and resulting field of view is shown:





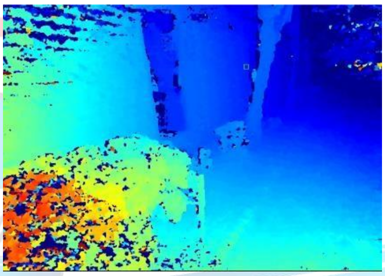
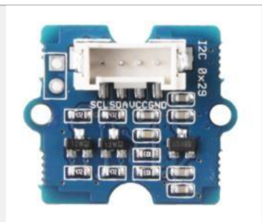


Figure: Arrangement of Cameras (zoomed in)

Figure: Benchmark Performance of MYNT Eye S camera (depth sensing)

- The stereo camera is coupled with a six axis IMU combined with frame synchronization which provide accuracy at less than one millisecond. Complete package with SDK is simple to integrate providing easy development and quick integration with the depth data created through the EYE S sensor.

Other Sensors:

Sensor Name	Sensor Type	Description	Sensor Model
VL53L0X Time-of-Flight Distance Sensor	Ranging Sensor	Short range (5m), high precision ranging sensor	

u-blox NEO-M8N	GNSS (GPS, GLONASS, BeiDou, QZSS, SBAS and Galileo-ready1	The NEO-M8 series of standalone concurrent GNSS modules is built on the exceptional performance of the u-blox engine in the industry proven NEO form factor.	

Table 17: Description of Sensors

Hardware Placement and Wiring (sensors):

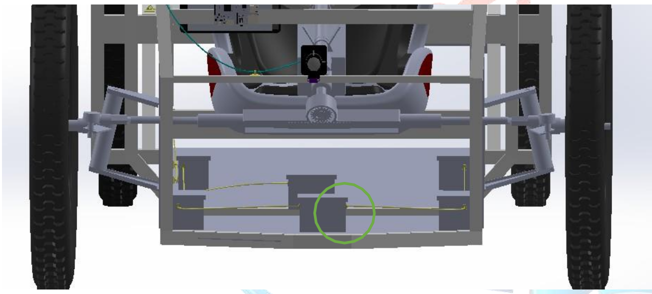


Figure: Placement of Arduino boxes and CAN-BUS wiring

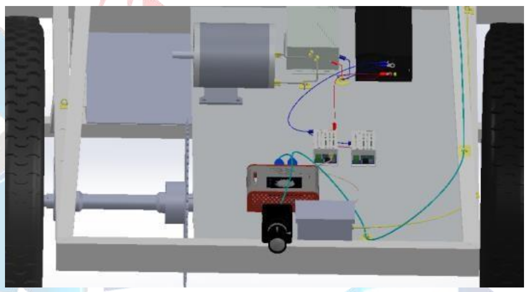


Figure: Placement of Electric Equipment, Xavier, and Back Camera

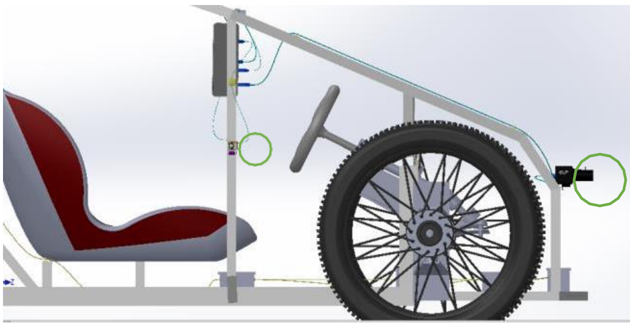


Figure: Placement of Cameras, and USB-hub extension

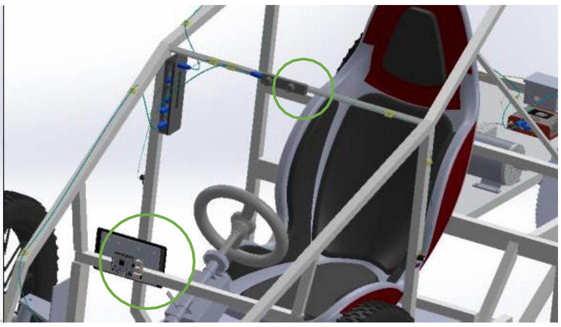
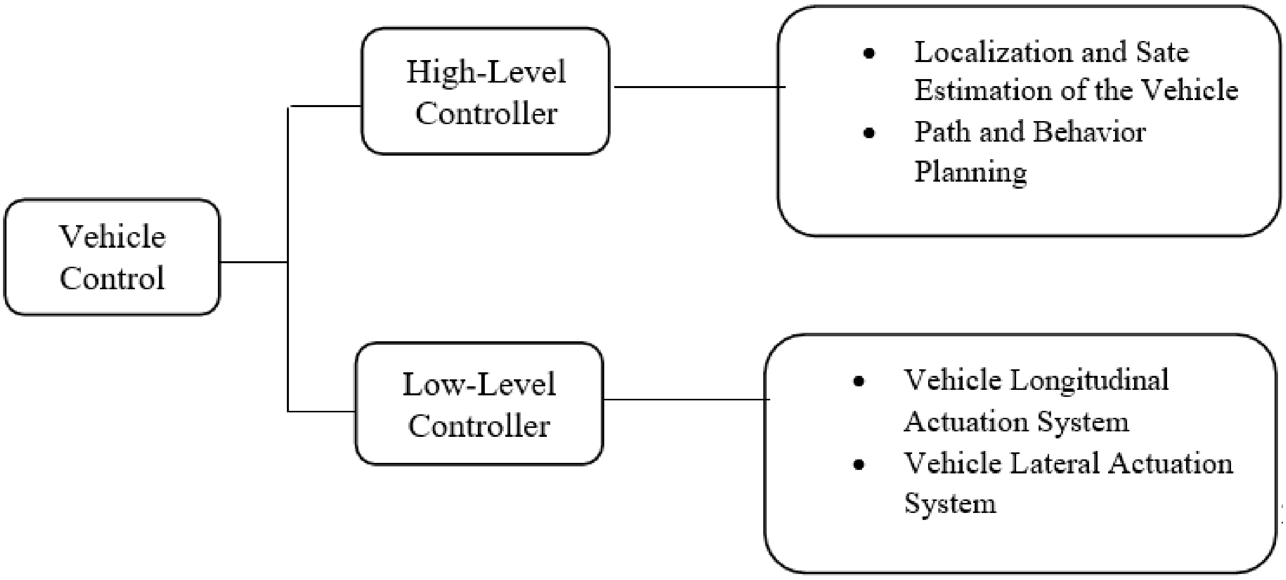


Figure: Placement of Stereo Camera, and LCD Display

8. Vehicle Control Unit

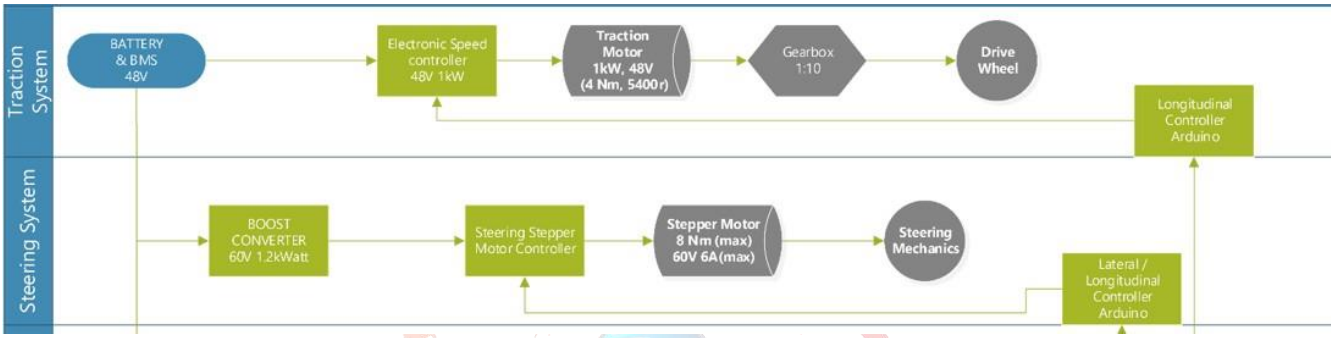
Information about the control unit intended to be used in the vehicle must be given. Features in your control software must be explained.
The vehicle control unit is responsible for carrying out the decision making of the self- driving vehicle. This unit consists of high-level planner and a low-level planner as shown in the figure. The high-level planner is responsible for determining the vehicle's location and generate a trajectory

towards the vehicle's destination. The output of the high-level controller are instantaneous desired forward velocities and steering angles that the vehicle must pursue. These desired quantities are fed into the low-level controller which controls the vehicle actuation systems allowing the vehicle to reach the desired state. The section below further elaborates on the Low-Level and the High-Level Controllers.



Low-Level Controller

The Low-Level Controller consists of two further units as indicated in Figure 1: a control system for vehicle's longitudinal motion and a control unit for the vehicle lateral motion. The longitudinal controller controls the vehicle's electric driveline or the powertrain of the car to ensure that the vehicle is moving at the correct velocities. Whereas, the lateral controller controls the steering system of the vehicle ensuring that the vehicle's heading is correct, and the vehicle is following the correct path. Figure 2 shows the design of the two subsystems of the Low-Level Controller and the respective actuation systems to be implemented in vehicle.



Vehicle Electric Drive Train Architecture Design

The vehicle is powered by a single 48V battery. The driveline of the vehicle consists of the wheel shaft coupled to a 1:10 gearbox connected to a 1KW BLDC motor. The motor speed is controlled by ESC combined with an Arduino ATMEGA. The Arduino ATMEGA implements a simple PID control algorithm and generates appropriate PWM signal for the ESC to control the motor speed. The PID constants of the actual vehicle are yet to be determined, however PID control for the longitudinal motion of the vehicle was tested on a sample vehicle body.

Vehicle Steering System Architecture Design

As shown in Figure above, the steering system of the vehicle consists of the Ackerman steering mechanics of the vehicle to a stepper motor which is controlled by an Arduino ATMEGA and powered by the main battery (passing through a boost converter). The Arduino implements the Stanley controller algorithm ensuring that the vehicle reaches desired steering angles.

Controller Algorithm Simulations

Following sections show the implementation of the controller algorithms through simulations on the vehicle body in AirSim environment.

PID Controller Design and Simulation on Airsim

The design phase of the controller (including both lateral and longitudinal controller) was explored using testing and simulations. The simulations were performed in Microsoft Airsim on Unreal Engine, with Ubuntu as the operating system. The simulation employed Robot-Operating System (ROS) environment, which was initially configured using an Airsim ROS-wrapper, allowing full access to the real-time feed from sensors. In this Airsim environment, the model of the vehicle was pre-selected. The plant function of this particular vehicle was determined, and the respective longitudinal and lateral controllers developed. The controllers were then tested and improved by simulating them in the Airsim ROS environment. The implementation of

ROS environment lead to efficient control and monitoring, while also allowing easier handling of sensor and odometry data. The simulation environment is shown below.

Vehicle Model Description

The parameters of the vehicle modelled for the controller design. These parameters were not similar to the actual NUSTAG vehicle due unavailability of some system parameters at the time. However, the modelling technique used for the system shall remain the same and modelling a different vehicle proves as a good practice for modelling the actual vehicle. The parameters for of the vehicle modelled are shown below.

- Vehicle mass = 170 kg
- Frontal Area = 1.45 m^2

Determination of PID gains for the Longitudinal Controller

After the plant transfer function is determined, the PID Auto-tuning app is used for the estimation of PID gains. PID gains which allow good settling time, remove overshoot, and allow better steady state response are selected. The initial estimation lead to the stabilization of the function, and then the best setline time was found. The parameters selected after tuning are given below:

- kp = 0.09535
- ki = 0.045757
- kd = 0.04968

The closed loop output response and the corresponding root locus plot is shown below:

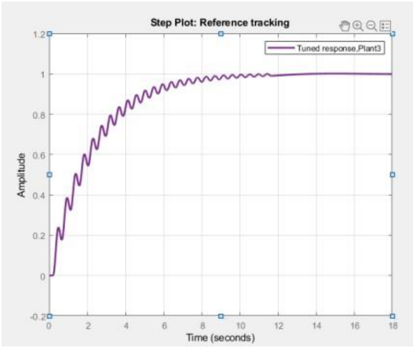


Figure: PID Tuning of Plant Transfer function

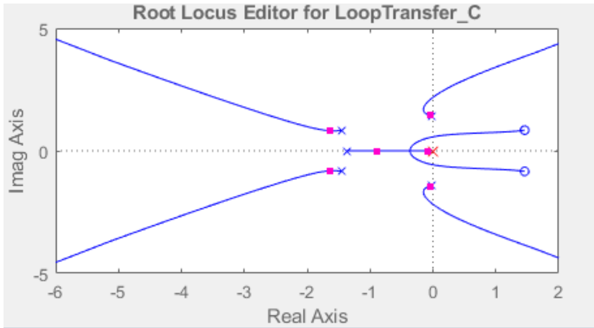


Figure: Root Locus system tuning

Longitudinal Controller Design Verification

With the implementation of longitudinal controller in ROS Airsim environment, the speed tracking is monitored. A reference trajectory is defined for the system and the response of the compensated system is observed. The results for the response of vehicle to input step response are plotted below highlighting good controller performance. Both the stability and the steady state characteristics of the system are improved with addition of PID controller in the design.

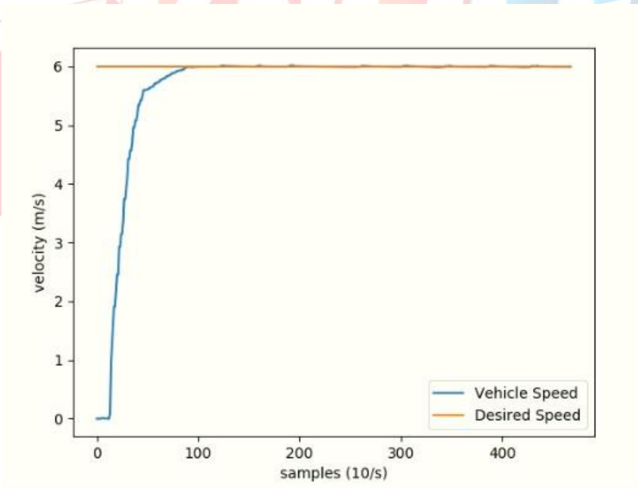


Figure: Longitudinal controller simulation response

Other parameters associated with the longitudinal controller are also tested in the simulation. The throttle of the vehicle is the output from the longitudinal system model. The relationship between the throttle and the speed tracking is highlighted. The plot below shows these characteristics showing high initial throttle to engage the car in motion, and then the value exponentially decays as the desired speed is approached. A steady state value of 0.55 is maintained to compensate for frictional forces on the car and tires, such that the desired speed is unchanged.

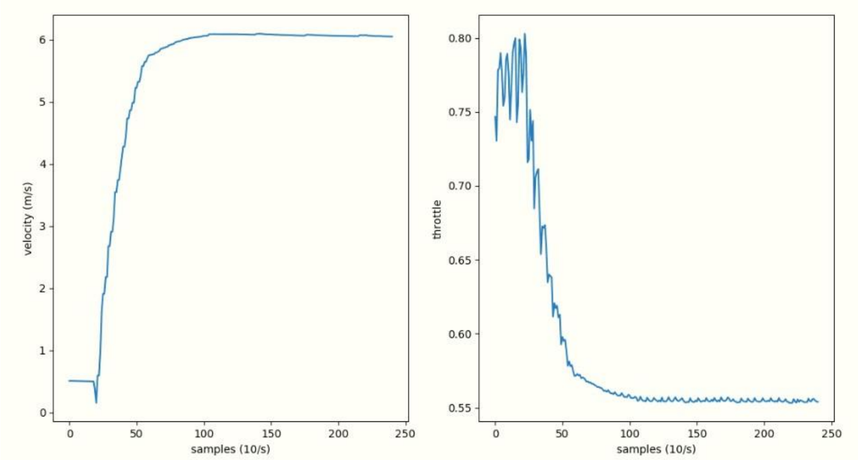


Figure: Relationship between vehicle speed and throttle

Vehicle Lateral Controller Design

For controlling the steering of the vehicle, Stanley controller is implemented. The Stanley controller outputs the steering angle the vehicle needs to make to track its desired path. It does so by observing the vehicle's current steering angle and heading.

The relevant parameter description used in the Stanley controller is shown in the figure below.

The gains for the Stanley controller are experimentally found. These are given below;

- ke = 0.5
- kv = 10

These perimeters were also tuned depending on efficient response time, and good tracking by the controller.

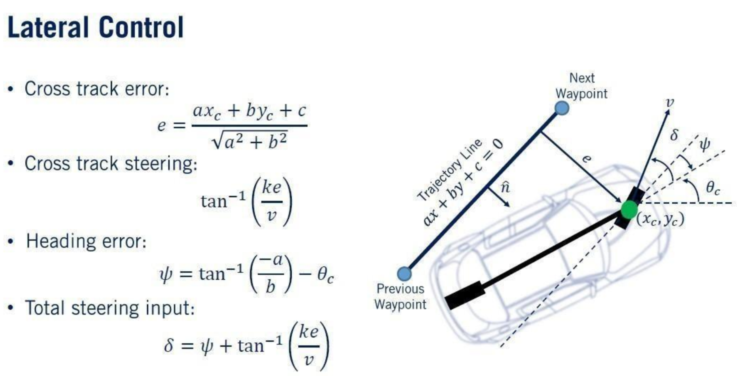


Figure: Description of the Stanley Controller

Implementation of the Stanley Controller

The algorithm made for the implementation of the Stanley controller consists of finding the parameters associated with the controller. These include: the cross-track error, *e*, the yaw/heading of the vehicle, θ_c, the heading error, ⍰ as show in Figure 13. The output steering is then evaluated by the formula in Figure 13.

Simulation and Results

The test scenario is shown below. The vehicle's heading is not aligned with the desired path. The vehicle needs to make a left turn to align itself yaw with the road heading. This is done by the implementation of the Stanley controller which evaluates the cross- track error of the vehicle to the desired path as well as the difference in headings and outputs the desired steering angles. The graph below displays this relationship between vehicle yaw and steering angle output.

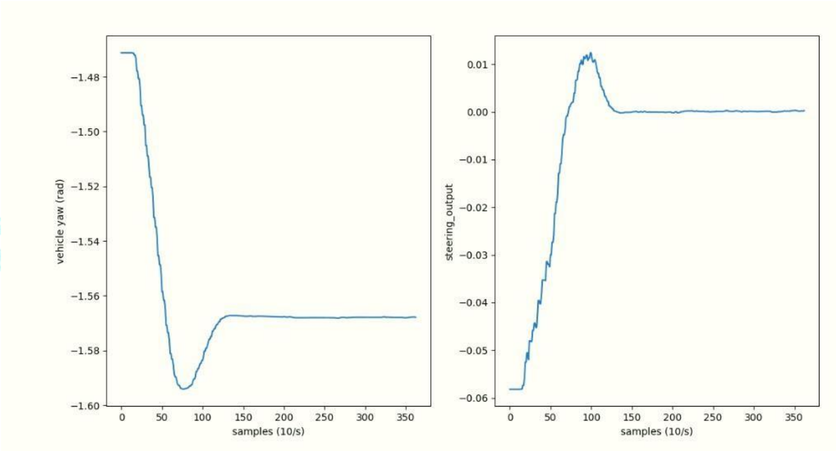


Figure: Relationship between vehicle yaw and steering angle

The same controller is tested for several turns and the graphs below highlight the initial run of the vehicle comprising of two complete left turns and two right turns. The steering angle is negative for left turns and positive for right turns. The relationship can be understood by relating to the graphs shown in figure, in which the start heading from the start location is – 90° (-π/2 rad) which is maintained for the first part of the road. The right turn causes the vehicle heading to change to -180° (-π rad). The scale limits angles from -180 to 180 degrees. The performance of the controller can be tested by the curvature of turns, and how smoothly the final heading is achieved i.e.

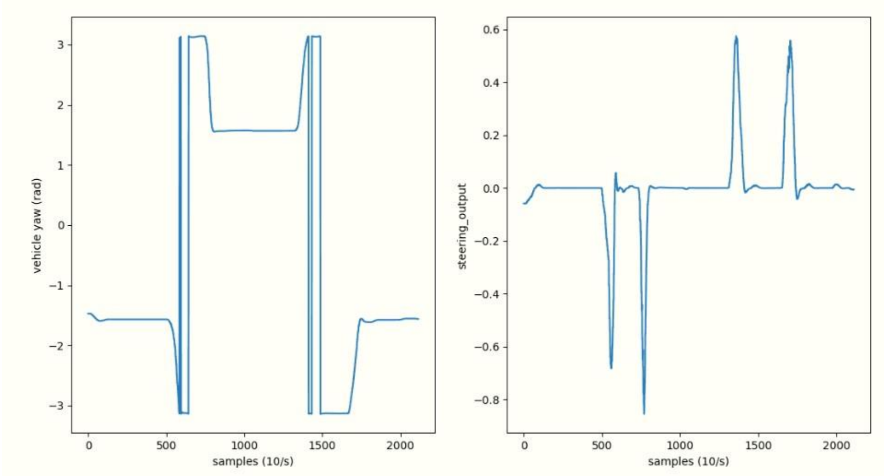


Figure: Relationship between vehicle yaw and steering angle (two left turns and two right turns)

Jetson AGX Xavier Developer Kit

- Jetson AGX Xavier is ideal for deploying advanced AI and computer vision to the edge, enabling robotic platforms in the field with workstation-level performance and the ability to operate fully autonomously without relying on human intervention and cloud connectivity. Intelligent machines powered by Jetson AGX Xavier have the freedom to interact and navigate safely in their environments, unencumbered by complex terrain and dynamic obstacles, accomplishing real-world tasks with complete autonomy. Jetson AGX Xavier's high- performance can handle visual odometry, sensor fusion, localization and mapping, obstacle detection, and path planning algorithms critical to next-generation robots.

	NVIDIA Jetson AGX Xavier Module
CPU	8-core NVIDIA Carmel 64-bit ARMv8.2 @ 2265MHz 8MB L2 + 4MB L3
GPU	512-core NVIDIA Volta @ 1377MHz with 64 Tensor Cores
RAM	16GB 256-bit LPDDR4x @ 2133MHz | 137GB/s
SECONDARY MEMORY	32GB eMMC 5.1
ENCODER	(4x) 4Kp60 | (8x) 4Kp30 | (16x) 1080p60 | (32x) 1080p30 Maximum throughput up to (2x) 1000MP/s – H.265 Main
DECODER	(2x) 8Kp30 | (6x) 4Kp60 | (12x) 4Kp30 | (26x) 1080p60 | (52x) 1080p30 Maximum throughput up to (2x) 1500MP/s – H.265 Main
CAMERA INTERFACE	(16x) MIPI CSI-2 lanes, (8x) SLVS-EC lanes; up to 6 active sensor streams and 36 virtual channels MIPI CSI-2, up to 40 Gbps in D-PHY V1.2 or 109 Gbps in CPHY v1.1 SLVS-EC, up to 18.4 Gbps
DISPLAY	(3x) eDP 1.4 / DP 1.2 / HDMI 2.0 @ 4Kp60
ETHERNET	10/100/1000 BASE-T Ethernet + MAC + RGMII interface
USB 3.0	(3x) USB 3.1 + (4x) USB 2.0
CAN BUS	Dual CAN bus controller
UART	UART, SPI, I2C, I2S, GPIOs
OPERATING TEMPRATURE	-25°C to 80°C
POWER CONSUMPTION	10W / 15W / 30W profiles, 9.0V-20VDC input

9. State Estimation

Implementation of Visual Inertial Odometry

Visual Inertial Odometry or VIO is implemented in the NUSTAG autonomous vehicle by using the VINS algorithm. VINS is a real-time localization framework based on multisensor approach for accurate state estimation of vehicle for autonomous application. It is based upon sliding window formulation and provides visual-inertial (VI) based odometry with high accuracy. Efficient IMU pre-integration, bias correction, and estimator initialization along with camera calibration, pose graph reuse and map merge makes VINS accurate and real-time. Moreover, it features loose coupling of GPS data with VI odometry output using Extended Kalman Filter to rectify the problem of accumulated drift over time.

System Sensor Overview

Sensors can be divided into two categories according to their frame of reference for measurements:

Local Sensors

These sensors are not globally referenced and hence an initialization for their reference is required for which generally first pose of robot is taken as origin. The estimation is than evolved from the set origin which causes the drift to accumulate over time. Camera, LiDAR, IMU etc. falls under this category.

Global Sensor

These are globally referenced and have a fixed frame usually earth frame. As the reference is fixed and known, the measurements can provide robots states globally but they are noisy. Moreover, the error is independent of the distance robot has travelled. GPS, Magnetometer, barometer etc. fall under this category.

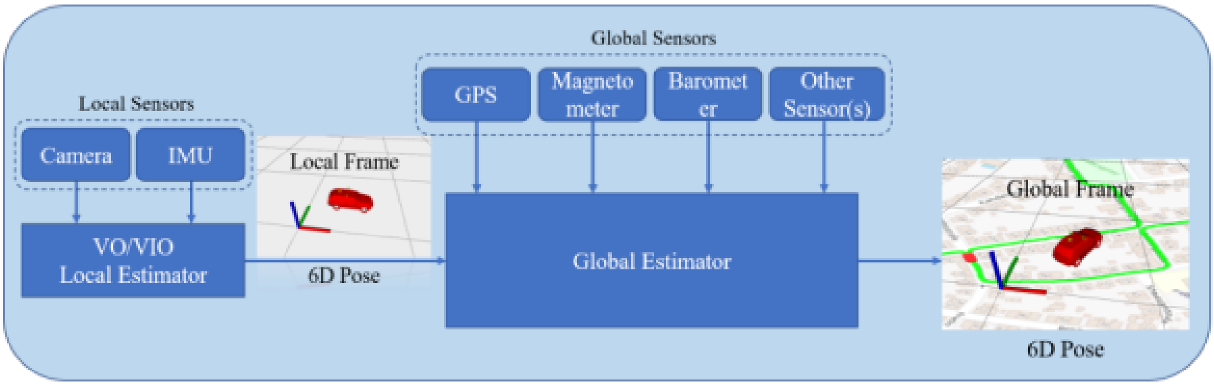


Figure 154. Representation of Global and Local Sensors

Local Pose Estimation

The VI odometry estimates poses of several IMU frames and features depth within a sliding window. The states are defined as follows:

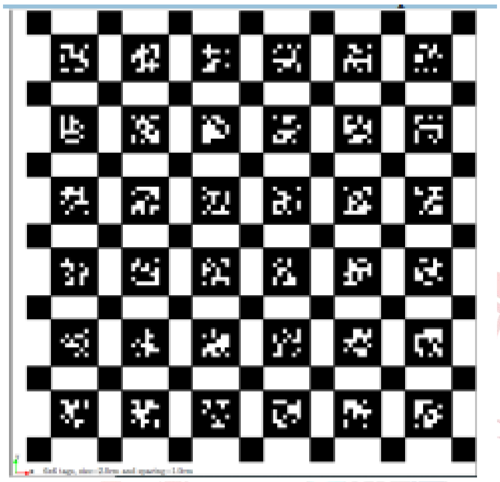
$$X^l = [x_0, x_1, \cdots x_n, \lambda_0, \lambda_1, \cdots \lambda_m]$$

$$x_k = [p^l_{b_k}, v^l_{b_k}, q^l_{b_k}, b_a, b_g], k \in [0, n],$$

where the kth IMU state (x_k) consists of the position ($p^l_{b_k}$), velocity ($v^l_{b_k}$), orientation ($q^l_{b_k}$) of IMU's center with respect to local reference frame l. We use quaternion to represent orientation. The first IMU pose is set as reference frame. b_a and b_g are accelerometer and gyroscope bias, respectively. Features are parameterized by their inverse depth λ when first observed in camera frame. The nonlinear least-squares are used to formulate the estimation problem.

Static Camera Calliberation

Kalibr tool is used for the static calibration of camera. Initially an april-grid checkboard is generated on an A4 sheet and printed and pasted on a hard surface such as cardboard. The grid is customizable based on the requirements and is displayed below:



A bag file is generated by subscribing to the left and right camera ROS topics, and the grid is moved in multiple directions while keeping the camera static. The results of calibration are shown below:

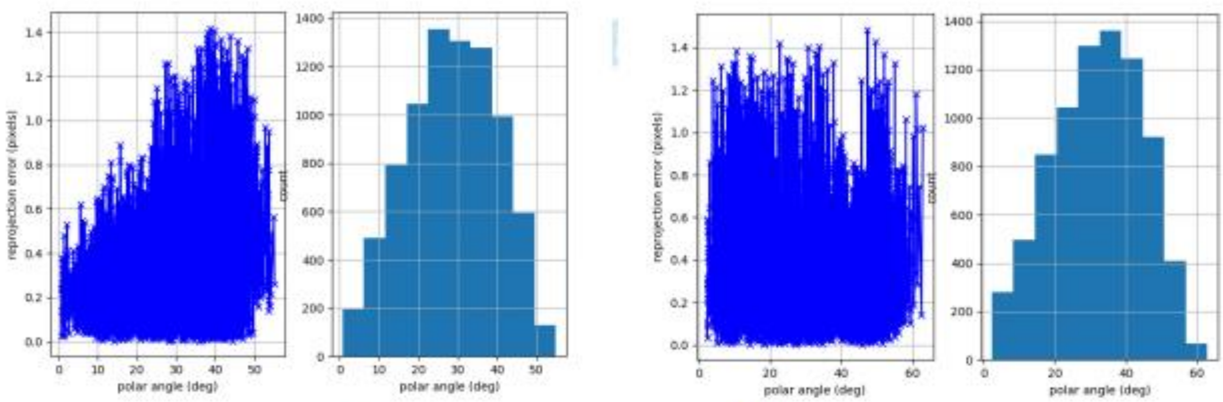


Figure 158. Camera-0 and Camera-1 Polar error

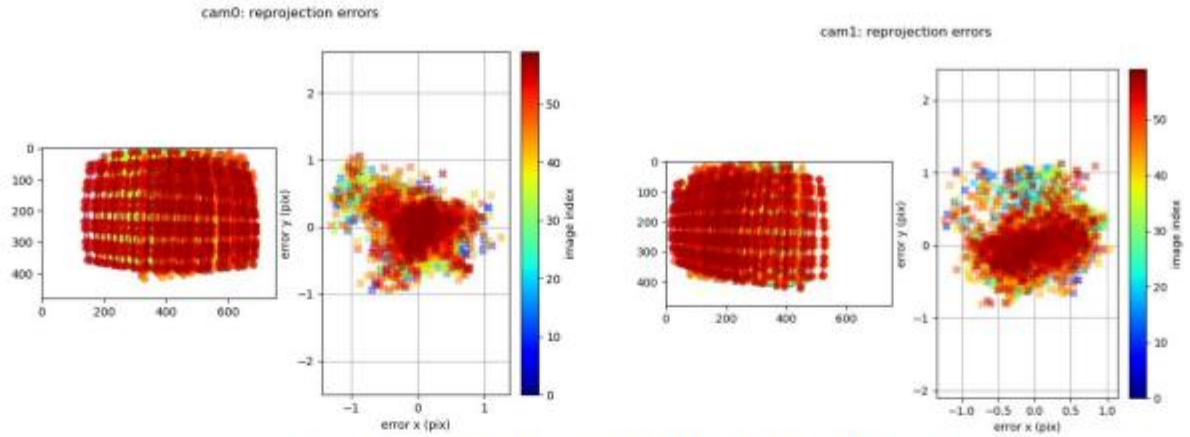
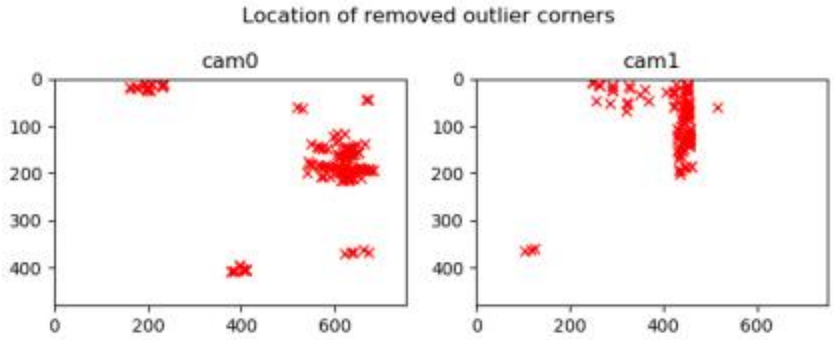


Figure 159. Figure 160. Camera-0 and Camera-1 Reprojection error



In the figure above, outliers are removed from the camera during calliberations.

VIO Implementation:

After successfully calibrating the cameras and IMU the VIO was implemented and the results were matched with the ground truths, yielding overall good accuracy of operation, and accurate pose estimation. The results of the VIO implementation have been shown below:

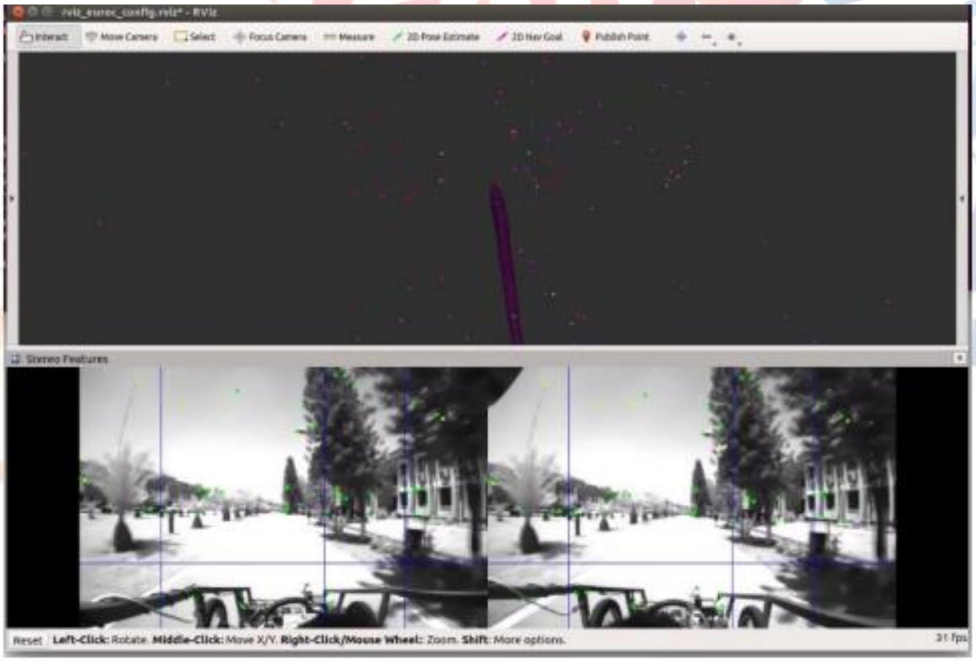
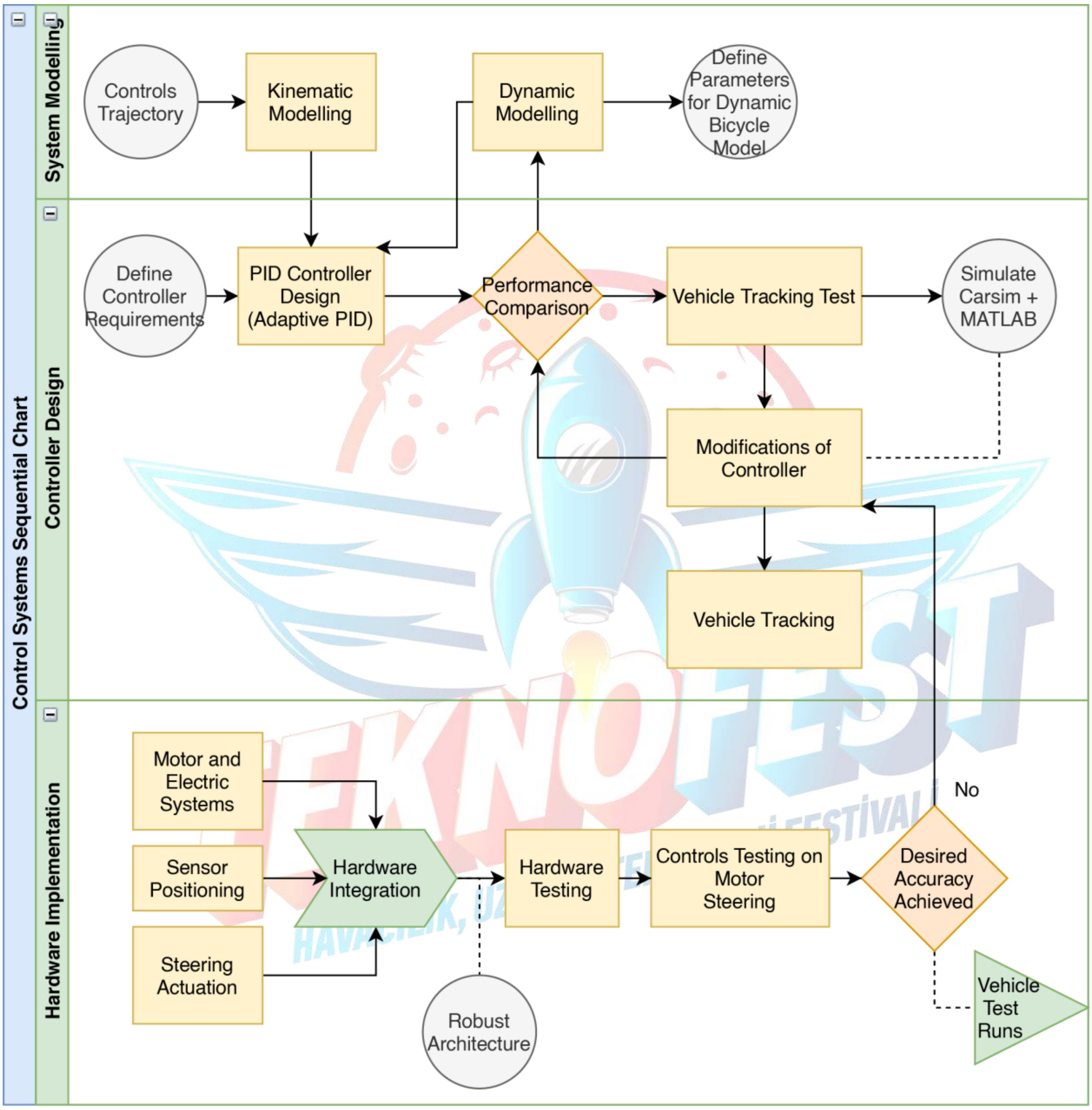


Figure 170. Output of VIO visualized on RVIZ on straight road

10. Autonomous Driving Algorithm

Controls System Algorithm Flowchart:



Perception stack.

Perception stack involves process to make our vehicle intelligent and give it ability to perceive environment around it, to drive autonomously. This is done by using camera as a sensor to perceive the environment and Jetson AGX Xavier as a brain to run the algorithms. The stereo camera is used for depth estimation and Global shutter high speed camera is used for object detection, segmentation, collision avoidance and lane keeping.

Camera Calibration:

Camera calibration is the process of computing the extrinsic and intrinsic parameters of a camera. This image information can be used to recover 3-D information from 2-D images.

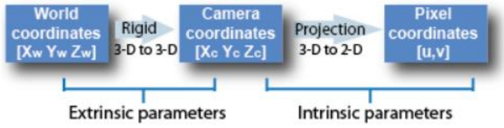


Figure: Extrinsic and Intrinsic Parameters

We can use this technique to undistort the fisheye lens camera (side cameras)

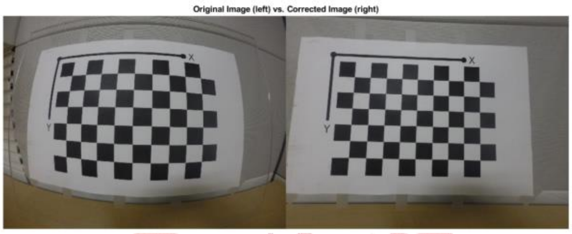


Figure: Calibration of Fish-eye lens

Stereo Camera:

The Stereo Camera has a variable range of 0.5 to 18 meters and enables excellent depth sensing. It provides optimal performance in both normal and low light circumstances, as well as precision and a wide field of view. The full-scale arrangement of cameras and resulting field of view is shown:

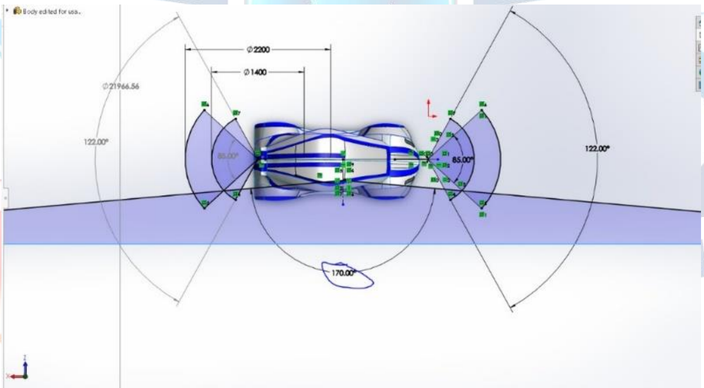


Figure: Arrangement of Cameras (zoomed in)

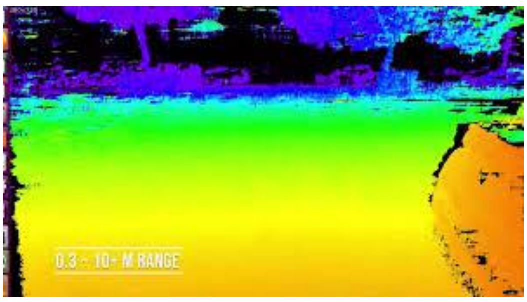


Figure: Benchmark Performance of MYNT Eye S camera (depth sensing)

Object Detection Algorithm:

The emphasis of scene recognition is on providing a global description of a scene, which is often summarized at a single scene category level. Object detection is concerned with locating object instances and categories inside a scene, which is commonly localized using bounding boxes.

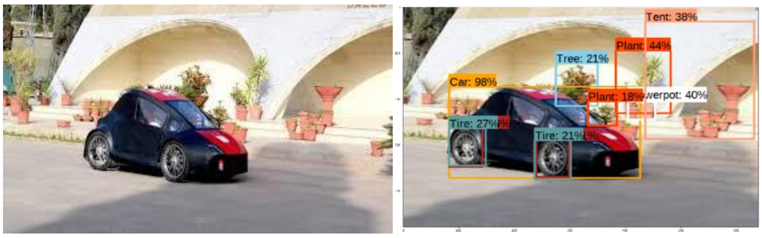


Figure: Output of Yolo v5 Network on NUSTAG vehicle

Traffic sign detection

YOLOv5 is used to to train on custom dataset of Turkish traffic sign.

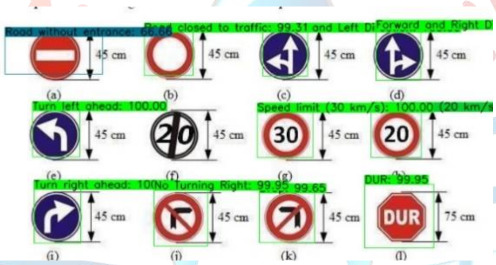


Figure: Output of Yolo v5 Network on Turkish traffic signs

Collision Avoidance system:

The pixel values of detected objects are obtained using an object detection method, and a region of interest is drawn on the road. When a pedestrian or another vehicle comes in front of our vehicle, the collision avoidance system applies the brakes.

Lane Detection:

Lane Detection is achieved using Computer Vision Algorithm. It uses Perspective Transform to project the lanes in Bird-eye-view. After application of edge detection, the algorithm measures the curvature and offset

Figure: Output of lane detection

Semantic segmentation Algorithm:

Semantic segmentation focuses on providing a finer-grained prediction of the semantic category to which each pixel belongs, whereas instance-specific semantic segmentation increases the difficulty of identifying the pixels that make up each object instance, combining semantic segmentation and fine-grained object detection. The encoder module uses a combination of convolution and pooling operations to extract DCNN features. The decoder module recovers the spatial details from the sub-resolution features, and predicts the object labels (i.e. the semantic segmentation)

Figure: Output of segmentation

The datasets which were used to train deep models for this autonomous driving include CamVid, the KITTI Vision Benchmark Suite, Leuven, Daimle Urban Segmentation and Cityscapes.

TensorRt:

TensorRt-based applications perform up to 36X faster than CPU-only platforms during inference, enabling to optimize neural network models trained on all major frameworks, calibrate for lower precision with high accuracy, and deploy to automotive product platforms. We deployed TensorRt inference for real time object detection. First, a network is trained using any framework. After a network is trained, the trained model is passed to the TensorRt optimizer, which outputs an optimized runtime also called a plan. The plan file is a serialized file format of the TensorRt engine. The plan file must be deserialized to run inference using the TensorRt runtime.

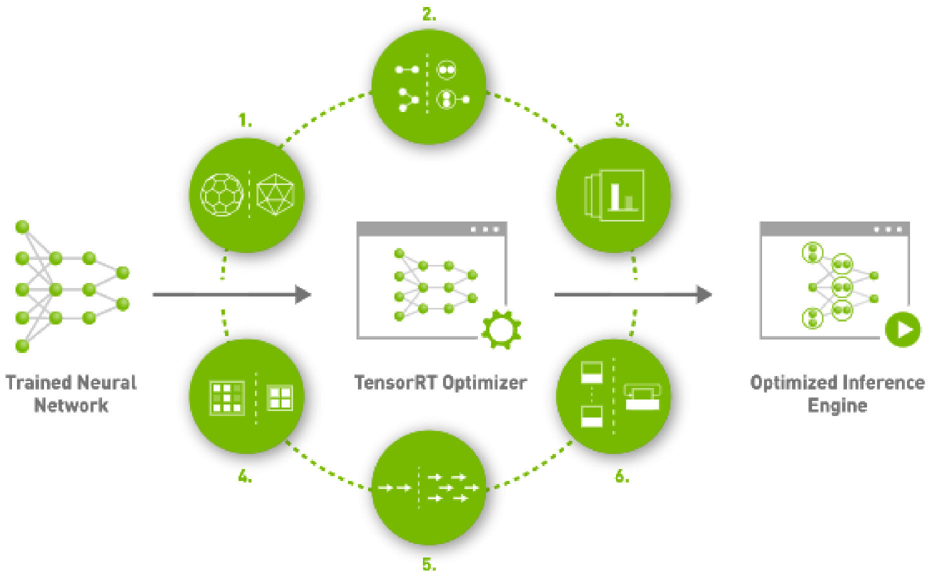
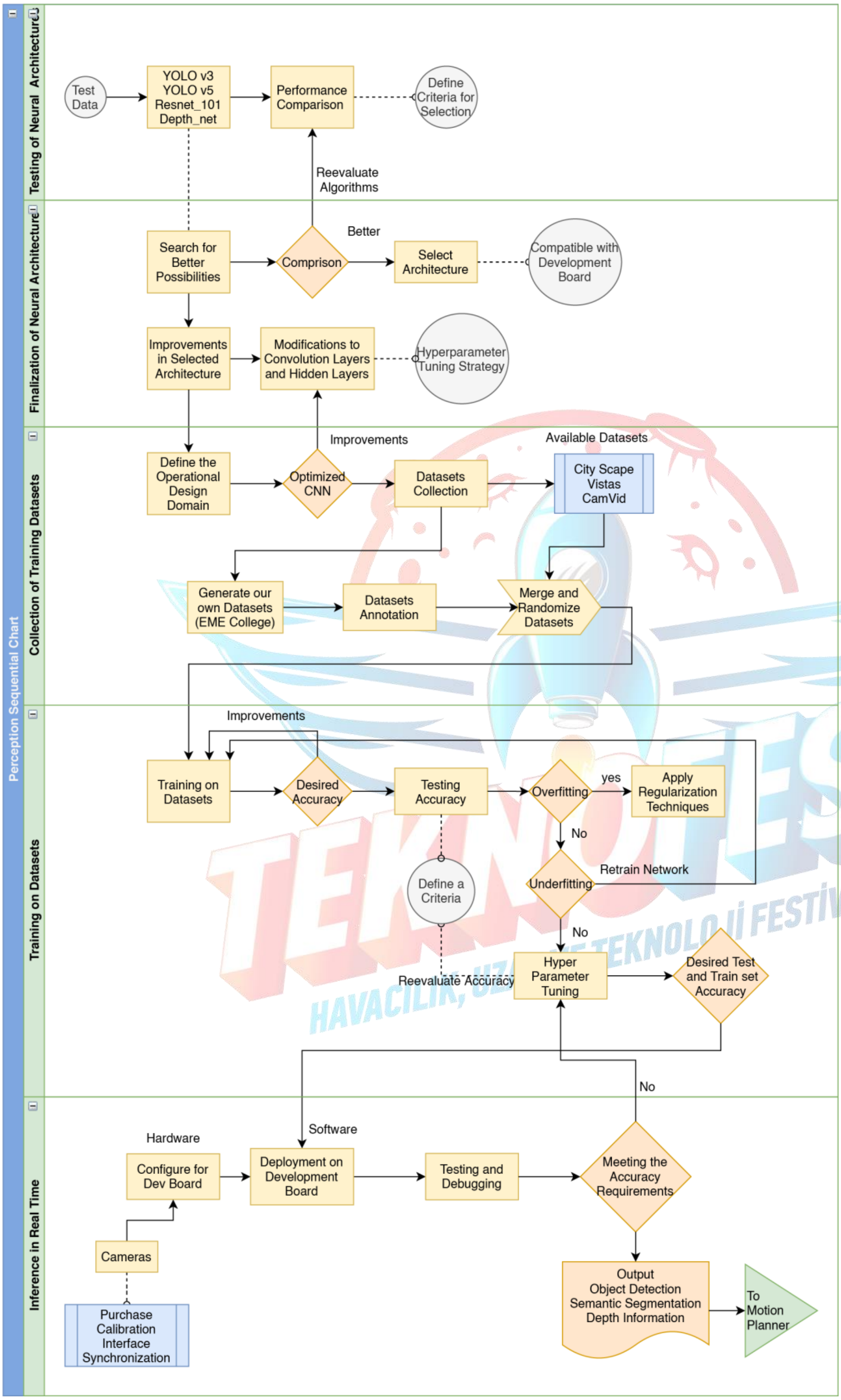


Figure: Optimized inference on trained model

Perception Algorithm Flow Chart:



Path Planning:

Components Required:

Raspberry pi 4	Value
Type	Raspberry Pi 4 Model B
RAM	4 GB
Integrated function	WIFI 802.11b / g / n
USB	2 USB 3.0 ports 2 USB 2.0 ports
CPU :	64 bit CPU
WIFI	2.4 GHz and 5.0 GHz IEEE 802.11
HDMI ports	2 micro-HDMI ports up to 4kp60
Display	
LCD 7inch Touch Screen	**Value**
Type	Touch HDMI Display-B

Display	

Global Path Planning

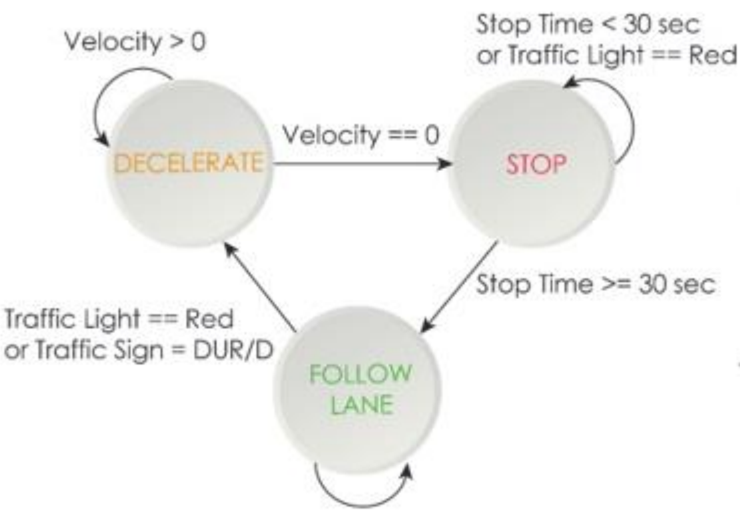
Methods that need previous knowledge of the car's environment are referred to as global path planning. It generates a simulated environment in which the techniques may design a path using this knowledge.

1. Communication with Graphical User Interface
2. Implementation of A* Star Search Algorithm
3. Generation of Waypoints
4. Finding shortest and Optimal Path

Behavioral Path Planning

On-road traffic, including moving impediments and static objects, is the focus of the behavior path planner. As input, it uses a traffic-free reference, moving barriers, and all roadblocks. It is also referred to as the ‘Brain’ of the self-driving vehicle.

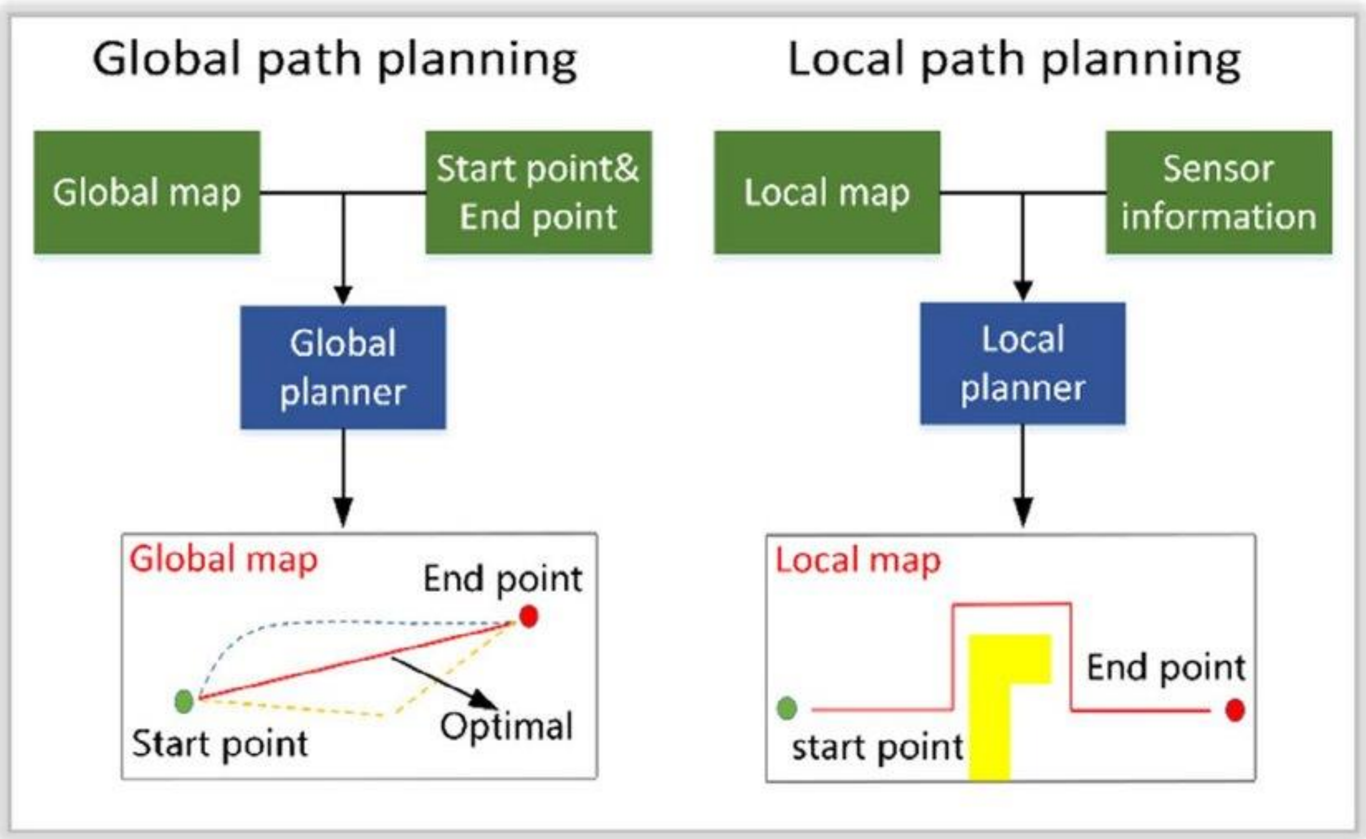
1. Making Decisions for the Car Maneuver
2. Implementing every possible path, the car can take while facing some situation
3. Giving commands to the local/lattice planner regarding the vehicle maneuver



Local Path Planning

Local route planning refers to techniques that use data from the environment to create a simulated field in which a path can be found. This enables a path to be identified in real-time while also adapting to changing barriers.

1. Lattice Planner
2. Trajectory Generation
3. Velocity Profile Generation
4. Path Optimizer



Communication with GUI

- The Odometric Information such as speed and location will be communicated to the Ethernet connected between Nvidia Jetson Xavier and Raspberry pi, which will communicate the Track information along with the A* map to be plotted and updated on the map.
- Raspberry pi will also receive the Battery status from the Ethernet connection, which will be calculated by the State of Charge of the Battery. The CAN-BUS will be connected to the sensor to measure the battery voltage, which will determine the SOC.

- The A* algorithm can be implemented entirely on Raspberry pi and the waypoints can be communicated to Xavier, to be used in Local Planning, since the waypoints will be constant throughout the journey, so only one-time communication is required.
- The display for the vehicle environment can be generated using the forward camera feed, which will be preprocessed by Xavier, and the real-time output of road plane equation, road boundaries, and the obstacle bounding box coordinates will be sent to the Raspberry pi.
- The Raspberry pi will plot these lines using Python GUI library, and replace bounding boxes with the vehicle logos, with their relative size equal to the bounding box, and the logo respective to the label of the bounding box.

GUI Code:

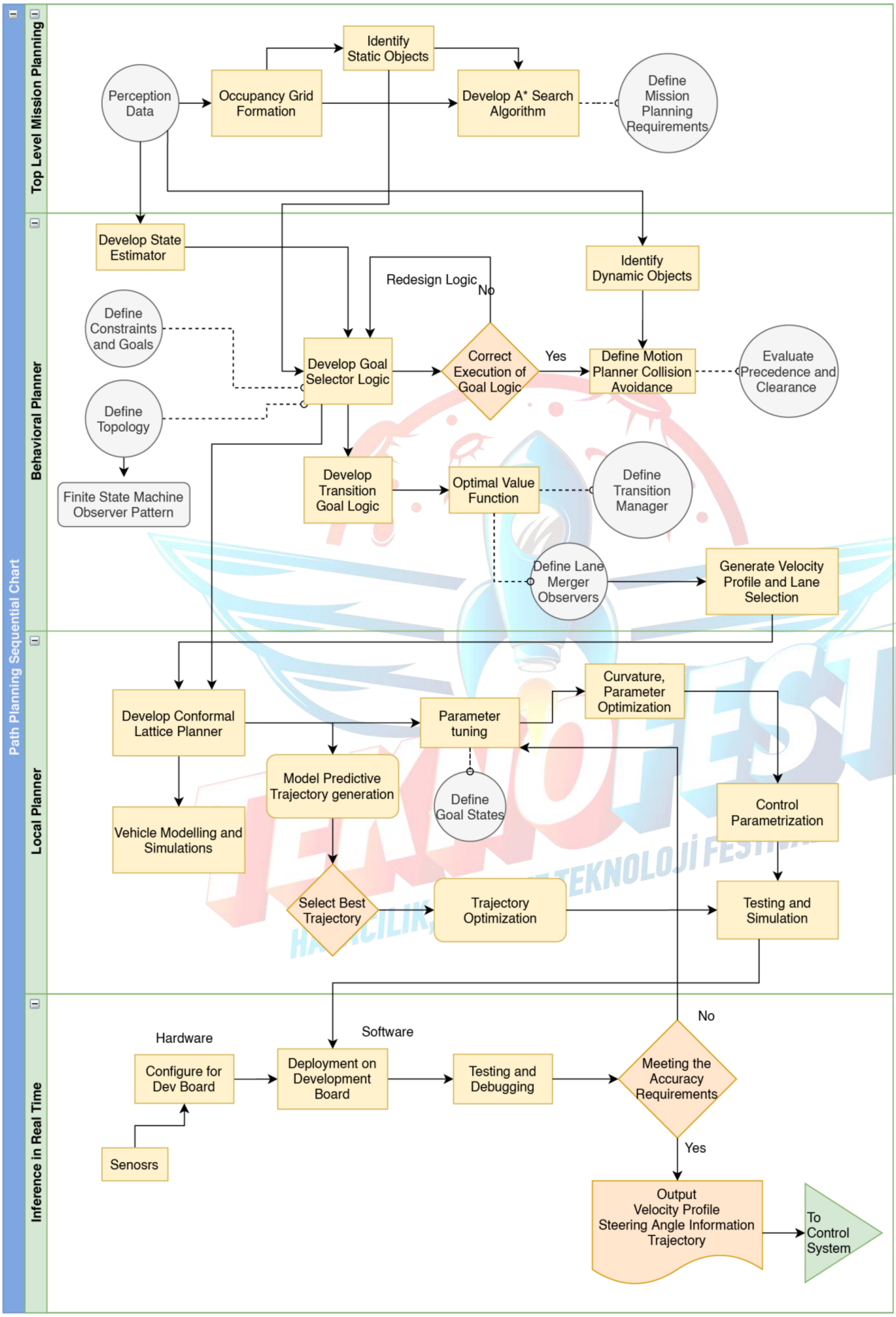
```
import getLATLONG as g
from sensor_msgs.msg import NavSatFix
import sys
import os
import rospy
os.environ["QT_XCB_GL_INTEGRATION"]="xcb_egl"


class gps_update:
    def __init__(self):
        self.sub = rospy.Subscriber('/ublox/fix', NavSatFix, callback=self.position_callback)
        self.lat = None
        self.long = None
        self.alt = None
        self.received = False

    def position_callback(self,gps):
        self.lat = gps.latitude
        self.long = gps.longitude
        self.alt = gps.altitude
        w.backend.set_lat_long(self.lat,self.long)

        print(self.lat,",",self.long)
        self.received = True
    """
    with open(output_file, 'a') as file:
        writer = csv.writer(file)
        writer.writerow([lat,long,alt])"""
```

```
#export QT_XCB_GL_INTEGRATION=xcb_egl
#while True:

#print(w.x_coord)



if __name__ == '__main__':
    try:
        rospy.init_node('gps_plot', anonymous=True)
        gps = gps_update()
        #if gps.received == True:
        app = g.QtWidgets.QApplication(sys.argv)
        w = g.Widget()
        w.show()
        app.exec_()
        waypoints = w.x_coord
        print(waypoints)
        rospy.spin()


    except KeyboardInterrupt:
        print ('Interrupted')
```


Local Planner code:

```
class LocalPlanner:
    def __init__(self, num_paths, path_offset, circle_offsets, circle_radii,
                 path_select_weight, time_gap, a_max, slow_speed,
                 stop_line_buffer):
        self._num_paths = num_paths
        self._path_offset = path_offset
        self._path_optimizer = path_optimizer.PathOptimizer()
        self._collision_checker = \
            collision_checker.CollisionChecker(circle_offsets,
                                               circle_radii,
                                               path_select_weight)
        self._velocity_planner = \
            velocity_planner.VelocityPlanner(time_gap, a_max, slow_speed,
                                             stop_line_buffer)
```

Path Planning algorithm flow chart:



11.Security Precautions

The following security precautions are taken:
All the sensor data and video feeds are transmitted through a wireless channel to our team, where we can monitor the real time states of the car. We can forecast any dangerous situation through this data and remotely shutdown the vehicle to prevent any unwanted situation from occurring. Moreover this data is stored and can be accessed latter for studying unlikely scenarios that car had faced in the past. For wireless link, our main processor acts as server and publishes sensor data and video feeds on a local IP that can be accessed for said purpose.

12. Simulation

Simulation URL:
https://youtu.be/FA7gdc7B_vA

13. References

- Coelingh, E., Eidehall, A. and Bengtsson, M. (2010). Collision warning with full auto brake and pedestrian detection-a practical example of automatic emergency braking. 2010 *13th Int IEEE Conf. ITSC. 155–160.*

- Chen, C., Wagner, G., Pace, M. and Rogness, T. (2008).Prediction of heavy truck stopping distance and vehicle behavior using real-time HIL and SIL simulation. *SAE 2008 Commercial Vehicle Engineering Cong. & Exhibition. SAE Paper No. 2008-01- 2604.*

- Distner, M., Bengtsson, M., Broberg, T. and Jakobsson, L.(2009). City safety—A system addressing rear-end collisions at low speeds. *Proc. 21st Int. Technical Conf. Enhanced Safety of Vehicles. No. 09-0371.*

- S. Nedevschi at el, 3D Lane Detection System Based on Stereovision, 2004 IEEE Intelligent Transportation Systems ConferenceWashington, D.C., USA, October 3-6, 2004

- R. Rajmani, "Vehicle Dynamics and Control", Springer, 2006

- Khalid El Majdoub, Fouad Giri, Hamid Ouadi, Luc Dugard, Fatima Zara Chaoui. Vehicle longitudinal motion modeling for nonlinear control. Control Engineering Practice, Elsevier, 2012, 20 (1), pp.69 - 81

- F. Ahmad et al, modelling and validation of the vehicle longitudinal Model, International Journal of Automotive and Mechanical Engineering (IJAME), Reasearchgate, 2014

- Raffaele Di Martino. Modelling and simulation of the dynamic behaviour of the automobile. Automatic. Université de Haute Alsace - Mulhouse, 2005. English.

- H. Wu, S. Cheng, and S. Cui, "A Controller of Brushless DC Motor for Electric Vehicle", IEEE Transactions on magnetics, vol. 41, no. 1, January 2005.

- Q. Huang, J. Li and Y. Chen, ”Control of Electric Vehicle”, University of Electronic Science and Technology of China.

- A. P. K.M and Usha Nair. (2017) “An Intelligent Fuzzy Sliding Mode Controller for a BLDC Motor,” International Conference on Innovative Mechanisms for Industry Applications:274–278.

- P. Zhao, J. Chen, Y. Song, X. Tao, T. Xu and T. Mei,” Design of a Control System for an Autonomous Vehicle Based on Adaptive-PID”, International Journal of Advanced Robotic Systems

- V.S. Patel, A.K.Pandey,” Modeling and Performance Analysis of PID Controlled BLDC Motor and Different Schemes of PWM Controlled BLDC Motor”, International Journal of Scientific and Research Publications, Volume 3, Issue 4, April 2013 1 ISSN 2250-3153.

- N. Seegmiller, “Enhanced 3D Kinematic Modelling of Wheeled Mobile Robots”, Robotics Institute Carnegie Mellon University

- A Kelly, “Mobile Robotics Mathematics, Models and Methods”, Cambridge University Press, 32 Avenue of the Americas, New York NY 10013-2473, USA, “Kinematics of Wheeled Mobile Robots”, page 196 – 198, (2013)

- G Rill, “Vehicle Dynamics”, Lateral Dynamics, pg 146-176, Hochschule Regensburg University of Applied Sciences, March 2009

- J. M. Snider, “Automatic Steering Methods for Autonomous Automobile Path Tracking”, Carnegie Mellon University, February 2009

- Abhisek Karki, Sudip Phuyal, Daniel Tuladhar, Subarna Basnet and Bim Prasad Shrestha, “Status of Pure Electric Vehicle Power Train Technology and Future Prospects”, Applied System Innovation, 17 August 2020

- Jian Zhou, Xiangming Shen2and Dong Liu, “Modelling and Simulation for Electric Vehicle Powertrain Control”, 1,2,3GE Global Research, 1800 Cailun Rd., Shanghai, China

- Myungwook Park, Sangwoo Lee, and Wooyong Han, “Development of Steering Control System for Autonomous Vehicle Using Geometry-Based Path Tracking Algorithm”, ETRI Journal, Volume 37, Number 3, June 2015

- Reimann G., Brenner P., Büring H. (2015) Steering Actuator Systems. In: Winner H., Hakuli S., Lotz F., Singer C. (eds) Handbook of Driver Assistance Systems. Springer, Cham

- N. H. Amer, K. Hudha, H. Zamzuri, M. Muhamad, "Adaptive modified Stanley controller with fuzzy supervisory system for trajectory tracking of an autonomous armoured vehicle", Robotics and Autonomous Systems, DOI: 10.1016/j.robot.2018.03.006, April 2018.

- Z. Li and J. Sun, "Disturbance Compensating Model Predictive Control with Application to Ship Heading Control", IEEE Transactions on Control Systems Technology, Vol. 20, No. 1, Jan 2012.
- Falcone, P. et al., "Predictive Active Steering Control for Autonomous Vehicle Systems", IEEE (2007).
- Sebastian Thrun, Sven Strohband, et al, “Stanley: The Robot that Won the DARPA Grand Challenge”, Journal of Field Robotics 23(9), 661–692 (2006)
- A. Kelly, “A Feedforward control approach to Local Navigation problem for Autonomous Vehicles”, The Robotics Institute Carnegie Mellon University, CMU-RI- TR-94-17.
- M. Park, S. Lee, and W. Han,” Development of Steering Control System for Autonomous Vehicle Using Geometry-Based Path Tracking Algorithm”, ETRI Journal, Volume 37, Number 3, June 2015.
- Hoffmann, G. et al., "Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing", Stanford University, (2007).
- C. K. Law, D. Dalal and S. Shearrow, "Robust Model Predictive Control for Autonomous Vehicle/Self-Driving Cars," in Cornell University Papers, May 2018.