Self-Driving Car: Mapping, Path Planning and Obstacle Avoidance

Aarjan Budathoki (077BEI004)
Abhigyan Bhusal (077BEI007)
Manish Guruwacharya (077BEI021)

Overview

- 1.3 million annual car accident deaths in the world, 3000 in nepal
- Cars have gotten safer, but it is not enough
  - Regulations and testing
  - Still nowhere near the safety of more automated modes (eg air travel)
- All SDC development happens in the West
  - Very different roads and driving conditions
  - Still not cost effective
- If SDCs work in South Asia, they should be home-grown

Literature Review

Self Driving Cars

- Autonomous navigation tech born out of rovers, space probes, missiles
- Accelerated by the space race, the cold war, and the later major conflicts
- VaMoRs and NavLab5 in the 80s and 90s

Inflection Point

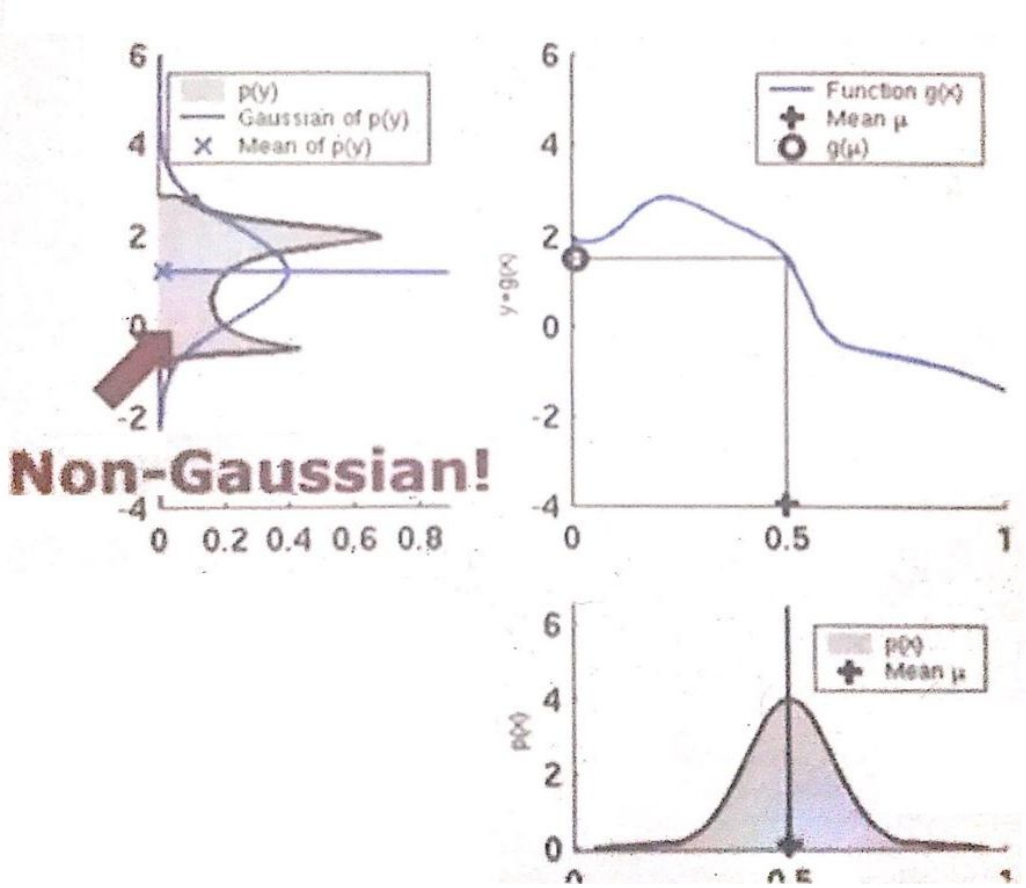
- DARPA challenges
  - No team completed the 2003 course
  - Stanley from Stanford won the 2004
  - Sebastian Thrun would later start what became Waymo
  - CMU won in 2007
- Stanley is a good general example

State Estimation

- We need accurate estimates of the vehicle's dynamics
  - Position (3)
  - Linear velocity (3)
  - Linear acceleration (3)
  - Orientation (3)
  - Angular velocity (3)
  - Sensor biases (3 or 6)
- Each sensor gives partial information
  - Have to fuse all information somehow

Kalman Filter

- Optimal filter for linear systems
  - Sensor model, dynamics, control model are linear
  - State follows a Gaussian distribution
  - KF steps are affine transformations
- Cannot be used for nonlinear systems
  - Gaussians are closed under affine transformations (eg Ax+by+c)
  - F(x) is not Gaussian for non-linear F


p(y)
Gaussian of p(y)
Mean of p(y)
Non-Gaussian!
Function g(x)
Mean μ
g(μ)
y = g(x)
p(x)
Mean μ

Non-Linear Kalman Filter

- Extended Kalman Filter
  - Locally linearize (Taylor Series upto the first order)
- Unscented Kalman Filter
  - Transform a few points, reconstruct a Gaussian (Sigma Points)

Mapping and Localization

- Use off the shelf components
  - Visual SLAM (ORB-SLAM, LSD-SLAM, NERF-SLAM etc)
    - Can fail in bad visibility conditions
  - Lidar SLAM (Cartographer, Hector SLAM, SLAM Toolbox)
    - Can fail to initialize in large maps
- Fuse and prioritize between two modalities

Perception

- Visual perception
  - Object Detection (YOLO, RCNN)
  - Segmentation (UNETs, Mask-RCNN, FPNs)
  - Scene Understanding
- Tracking distinct objects (obstacles, vehicles, etc.)
  - Kalman Filters for each obstacles
  - Trackers “live” until obstacles are observable

Control

- Proportional-Integral-Derivative
  - Simple, easy to implement
  - Works well for linear system dynamics
- Model Predictive Control
  - Planning over a finite time horizon
  - Minimize error over time, not just instantaneously
  - Need good system model
  - Robust, fewer degenerate cases

Methodology

Scale of the Project

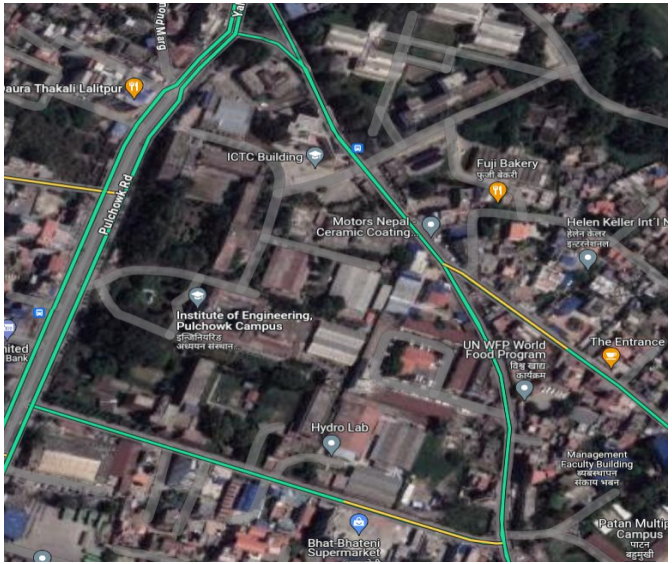
- To power the system, **LiPo batteries** are used placed in the central area of the chassis.

- **Servo motor** is used for **steering the front wheels** through a bar linkage system connected to front wheels.

- **DC motors** attached with **hall encoder** is used to drive the **rear wheels.**

Autonomous Region and Environment

For **testing and operation** of our autonomous vehicle, we will be driving in the road from the main gate next to the ICTC block and towards all the paths in this side of the campus.



Track Conditions

- **Shaded region from trees on the road**
- **Pedestrians strolling in the road**

Non Goals

- **Harsh lighting**
- **Not high speed** vehicle

Absolute and Relative Positioning of the System

- To command our vehicle to reach to the **target point**, we must know the **position of the vehicl**e

- **Non Linear Kalman Filter** is used to estimate the position of the vehicle

- **IMU + ODOMETRY + LIDAR** data fused together using **Unscented Kalman Filter** or **Extended Kalman Filter**

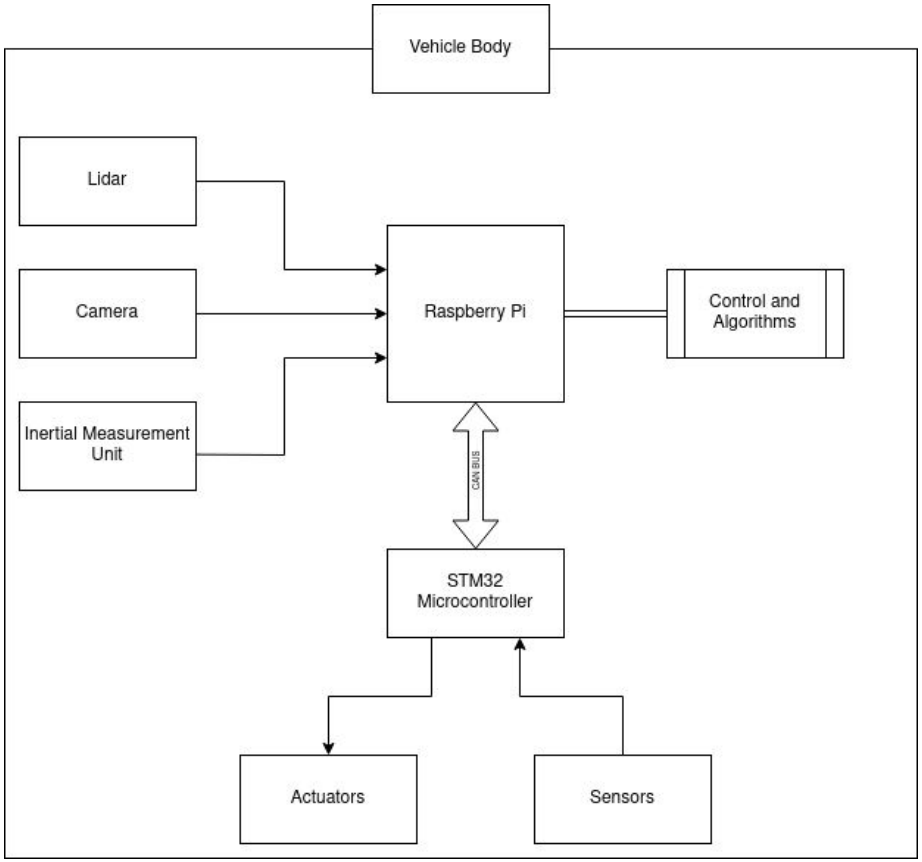
Team and Work Division

- For **designing the mechanical system** of the vehicle and required tools we get huge support from Robotics Club and members
- **PCB Designing** using Electronics Design Automation softwares like Altium, EasyEDA, KICAD.
- Designed PCB will be **fabricated** from JLCPCB or similar reputable manufacturers.
- **Control System Design** includes **vehicle state estimation, vehicle control and stability** in various operational conditions.
- **Optimization** done through **Perception**, by **fusing the imu, odometry, lidar and camera data** together for **position and estimation** and then optimization.

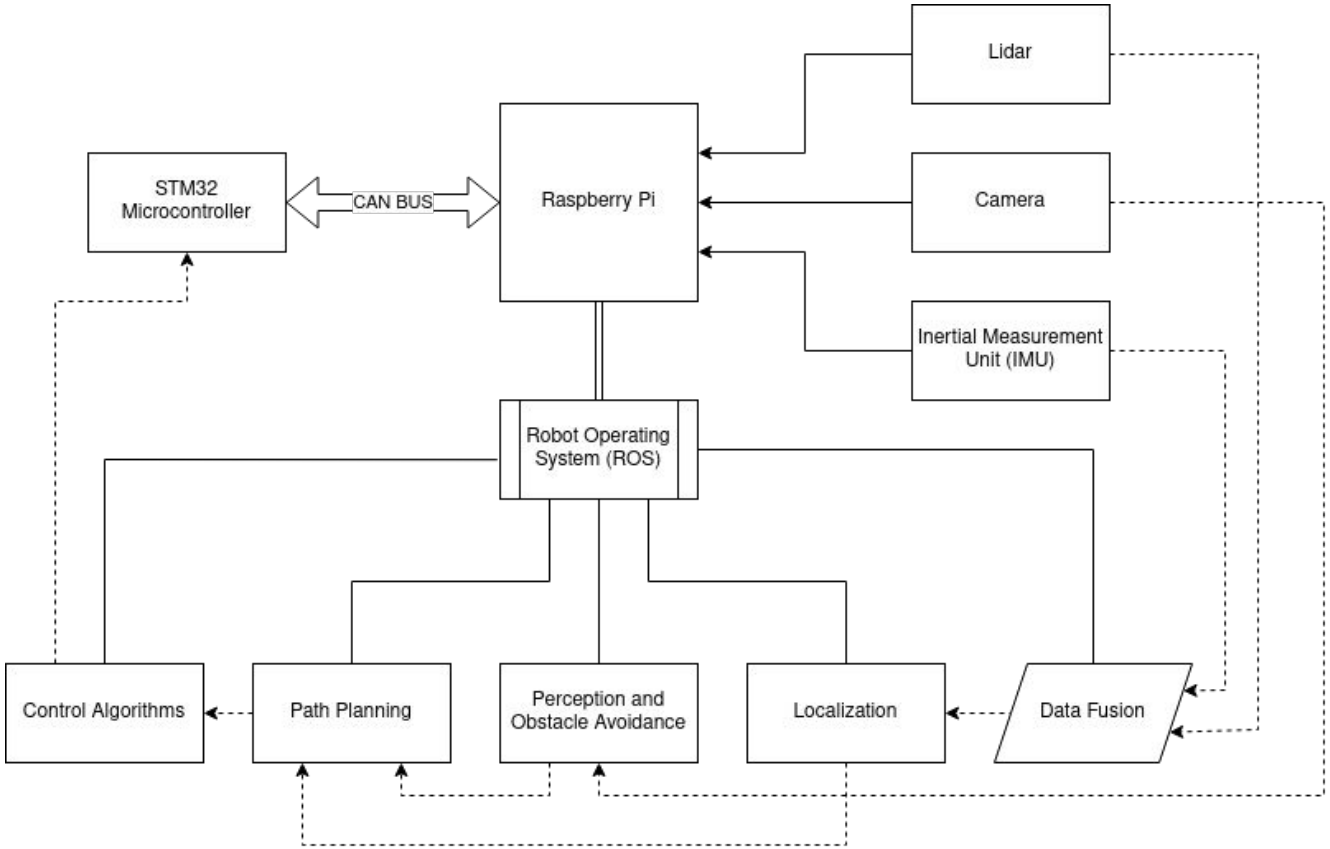
Equipment, Tool and Devices

- Microcontroller
  - STM32F407, STM32F103, STM32F446RE
- Raspberry Pi
  - Raspberry PI 4
- Inertial Measurement Unit (IMU)
  - BNO055
  - MPU6050
- Camera
  - Logitech C270
- Lidar
  - A1 RP Lidar

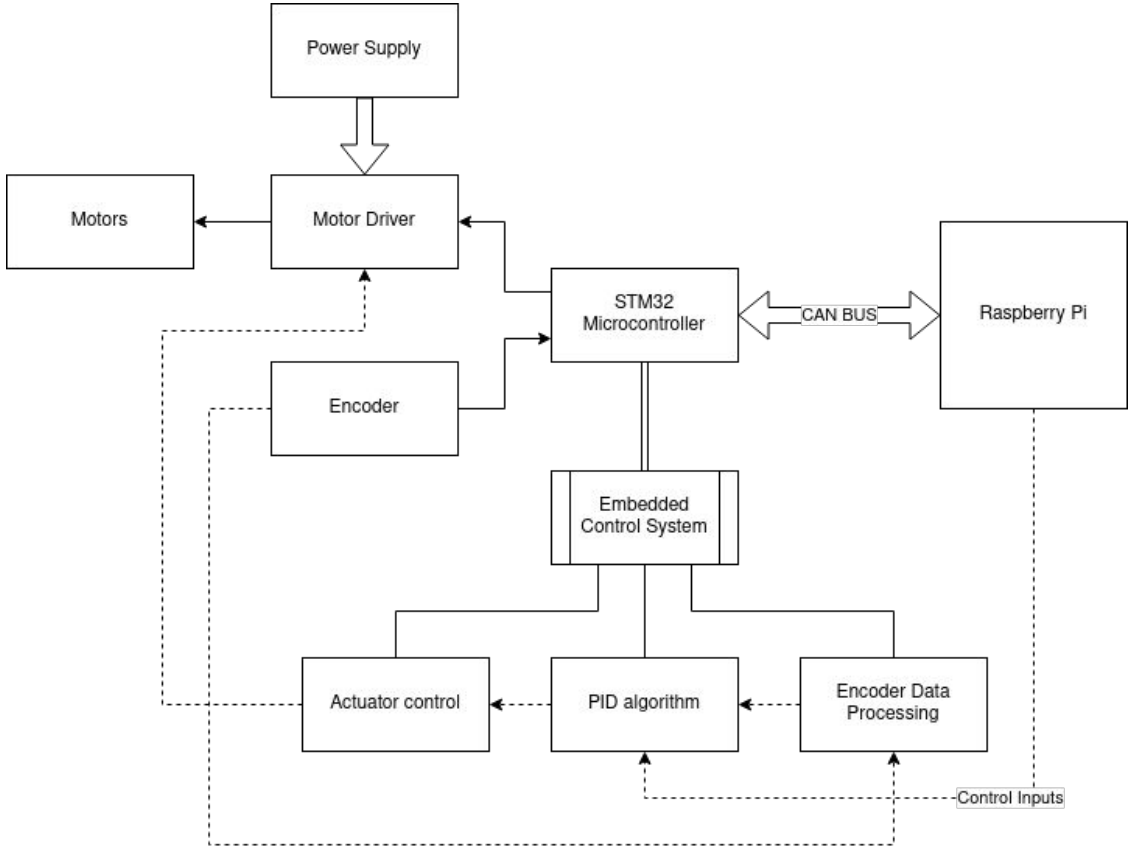
System Design



System Overview



Main Controller

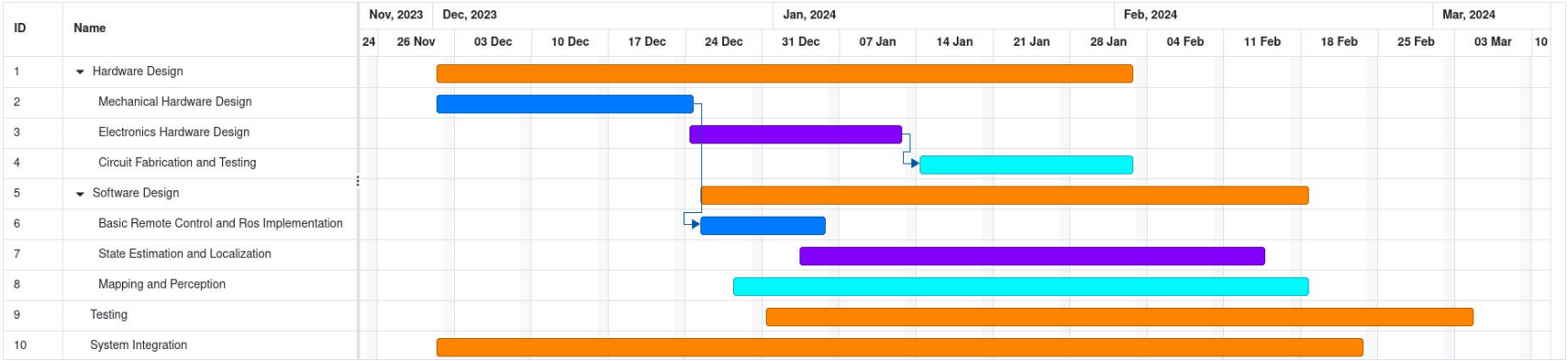


Main Controller

Expected Outcome

- Autonomous
- Adaptive to dynamic environment
- Able to handle internal and external errors
- Manual override over Autonomous driving

Timeline


ID
Name
Nov, 2023
Dec, 2023
Jan, 2024
Feb, 2024
Mar, 2024
24
26 Nov
03 Dec
10 Dec
17 Dec
24 Dec
31 Dec
07 Jan
14 Jan
21 Jan
28 Jan
04 Feb
11 Feb
18 Feb
25 Feb
03 Mar
10
1 Hardware Design
2 Mechanical Hardware Design
3 Electronics Hardware Design
4 Circuit Fabrication and Testing
5 Software Design
6 Basic Remote Control and Ros Implementation
7 State Estimation and Localization
8 Mapping and Perception
9 Testing
10 System Integration

THANK YOU