1 Introduction

This report will provide an overview of the advancements and activities achieved during the third month of the competition. The team has improved the project architecture and lane-following algorithm, and now we're using a perspective-based approach using the sliding window technique. This led to increased accuracy and precision; thus, our algorithm is more reliable. We also started working on sign detection and intersection navigation. We have also successfully run the Gazebo simulation.

2 Planned activities

1) Project architecture
2) Improving lane following and implementing speed control
3) Adaptive ROI
4) Path planning and vehicle control
5) Intersection detection
6) Traffic sign detection
7) Creating a physical and virtual testing environment

3 Status of planned activities

3.1 Project architecture

Status: Ongoing 75%

We understood the brain project and did a whole new project that is only run on raspberry pi. We took many ideas from the original project architecture that was provided, but we adapted them to fit our algorithms. We have a Decision-maker process that aims to orient the car on the map and choose the process that will send commands to the serial handler. For example, Decision-maker will start the Lane-Follower process, and Lane-Follower will send commands to Serial-Handler. Lane-Follower will throw a trigger when the horizontal line is detected, and Decision-maker has the task of choosing a new process. We think this architecture is scalable and will be easy to debug.

Difficulties: Running the whole code on the Raspberry pi is slower than running it on a remote device, so we have trouble with execution speed.

3.2 Improving lane following and implementing speed control

Status: Ongoing 80%

We changed the lane-following algorithm completely, and now we use a method that involves changing perspective. On the bird's eye view of the terrain, we perform the sliding window technique. The program operates in 4 mods when both lines are detected, when only left or right lines are detected, and when none are detected. The algorithm is robust and works well when it is performed on the remote machine, but when it is running on Raspberry, we have lags and considerable delays.

We made an algorithm to detect stop lines (horizontal lines) using a sliding window, but now we calculate the middle line and follow it with a sliding window. The vehicle slows down gradually and stops before the stop line.

3.3 Adaptive ROI

Status: Finished

We change ROI when the vehicle gets closer to the stop line, so it won't have problems with steering when there is a crosswalk after the stop line. It is done gradually on every frame, giving precision, but it is computationally expensive.

3.4 Path planning and vehicle control

Status: Ongoing 50%

Our vehicle control strategy is to detect offset from the middle of the path and calculate the steering angle using PID. PID gets offset from the center of the road and calculates the steering angle using proportional and differentiation components. We also plan to include a radius of curvature in calculating the steering angle to get better results.

3.5 Intersection detection

Status: Ongoing 50%

We detect intersections with the help of detecting stop lines. We plan to have predefined routes for intersection navigation.

3.6 Sign detection

Status: Ongoing 60%

Using the TensorFlow framework, we have built a deep-learning model to recognize traffic signs in the Colab environment. Our first challenge was finding a good training dataset. We have researched several options and picked German Traffic Sign Dataset from Kaggle, but since it doesn’t contain all signs, we plan to add our photos to complete it. We are using 2 CNN layers and two dense layers. We have also used the Adam optimizer and dropout technique. The algorithm works well on the classification of 3 sign classes.

3.7 Creating a physical and virtual testing environment

We have many problems finding a big place with a floor that doesn’t have too many stripes. For now, we are making improvisational testing tracks and plan to get a full 15x15 meters map. Details on the map are specified in the previous report.

4 General status of the project

We developed good algorithms that perform well on videos from previous years, and we defined starting and scalable architecture. Still, we didn’t transfer all that to be running only on the car.

We are generally satisfied with the amount of work we did in the past month.

5 Upcoming activities

1) Make progress in simplifying algorithms
2) Implement sign detection in real time
3) Intersection navigation
4) Traffic light detection
5) Finding the path and orientation in the map