1 Introduction

The current report summarises the technical progress of the team LH - CDC team, in the Bosch Future Mobility Challenge 2025 until February, 17th, 2025. Particularly, the activities planned during the time-period between the 20th of January 2025 and the 17th of February 2025 and their current status, as well as the upcoming activities are described shortly.

2 Planned activities

The planned activities during the aforementioned period, additionally to the team members in charge for each activity are shown below:

- Perception
  - **Vehicle Detection**
    @Phong
  - **Intersection Detection**
    @Cường
  - **Ramp Detection**
    @Huy
- Control
  - **Intersection Navigation**
    @Phong
  - **Path Planning**
    @Cường
  - **Reaction to static elements**
    @Huy
- System integration
  - **Finite State Machine (FSM) implementation**
    @Thảo
  - **Total code integration from simulation to real-life car**
    @Cường

3 Status of planned activities

3.1 Vehicle Keeping [Done]

3.1.1 Implementation

For the vehicle detection, we followed a classifier-based approach. At first, we tried using an already trained vehicle cascade classifier, which presented low accuracy since it was trained with real vehicles, therefore was not suitable. As an alternative, we collected and processed our own training data by gathering images both from real-life cars and the simulation car and executed the actual model training. Many different combinations of the training's parameters have been tested, until we managed to get a satisfying and effective result.

3.1.2 Issues

Although our cascade classifier model is high in accuracy when detecting our car, there are still some false posi-tives. More attempts with different training parameters will be made, otherwise other ways of rejecting the false positives are going to be applied.

3.2 Path Planning / Position Fusion / Intersection Navigation [Done]

3.2.1 Implementation

Given a starting and a finish node, the path the vehicle should take is calculated with conventional path planning algorithms. To efficiently navigate the track, we used inputs from the camera, the GPS and the BNO sensor. The decision moments come when the car is at an intersection. To minimize the errors from the GPS, we detect an intersection visually from the camera. Then, depending on where the next node is compared to the car, and taking into account the yaw, via the BNO, the car steers accordingly. After navigating the intersection, the car switches back to the lane keeping algorithm.

3.2.2 Issues

The main issues when implementing the above features-algorithms come from the errors and inaccuracy of the GPS. These become apparent when navigating the intersections, as there cannot be an accurate measurement of where exactly the car is on an intersection, as to take the correct steering angle. These have been minimized by utilizing the visual inputs of the camera, and even though they do not seem to affect the efficiency of the navigation, further steps to optimize it will be taken in the future.

3.3 Intersection Detection [Done]

3.3.1 Implementation

Our algorithm identifies all horizontal lines. Then it merges all the lines into one that fits the intersection line specifications. We measure the distance from the center of this line to get an approximation of the actual distance between the vehicle and the intersection line. To further validate that this line corresponds to an intersection we take into consideration the signal coming from the GPS.

3.3.2 Issues

The main problem we encountered was that the intersection lines are not always perpendicular to the car's direction in the frame. In addition to that, while approaching the line, its slope changes according to the vehicles orientation. In order to control that we adapted our thresholds accordingly.

3.4 Ramp Detection [Done]

3.4.1 Implementation

We have utilized the BNO055 IMU sensor in order to obtain the pitch angle of the vehicle. During testing, we plotted the pitch values coming from the IMU and we came up with thresholds that decide if the vehicle is climbing up or going down the ramp. Using this the vehicle can decide if it is on the ramp or not.

3.5 Activities under [Development]

Activity	Goals	Implementation
Reaction to static objects	Overtaking detected static vehicles	• Distance from the detected vehicle is calculated • Current path is interrupted

		● Next target node is defined, taking into account the forward vehicle's coordinates ● Detour path is generated
Finite State Machine (FSM)	Organizing the whole system-level workflow for structured system integration.	● The diagram of the FSM, included all acceptors, classifiers transducers and sequencers has been developed ● The FSM is implemented in the main function of the workspace, in order to define the processes' sequence.

4 General status of the project

In general, the team continuously verifies the effectiveness of each implementation both in the simulation and the car & track facilities. Currently, the algorithms which constitute the interface between the individual scripts and the total system integration are being implemented. Thus, the total system integration is in progress, carried out both in the simulation and the car. Finally, the test procedures for the final unit, functional and integration tests are being developed.

5 Upcoming activities

According to the suggested timeline of the team, during the period between the 17th of February 2025 and the 17th of March 2025, the upcoming activities are:

- Perception Integration
  - Vehicle detection [Finalization]
  - Bumpy Road detection
- Control
  - Reaction to static elements [Finalization]
  - Parking maneuver
  - Reaction to dynamic elements
  - Roundabout navigation
- System Integration
  - Finite State Machine [Simulation and Finalization]
  - Early total system integration and preparation for qualification
  - Traffic Lights Interaction