

1 Introduction

During this phase of the project, we encountered significant challenges in simulator development due to a misunderstanding of the "status2" intention, which led us down a number of unproductive paths. Additionally, we faced numerous bugs stemming from issues with the official code and the computer environment. As a result, the lane-keeping function is still under development.

2 Planned activities

We focus on resolving technical issues with simulator configuration and ROS integration, ensuring all modules function smoothly. We also design the path tracking module for accurate path following in various environments, a key component for autonomous driving. Once this is done, we design the state machine to enable smart decision-making and vehicle control based on different driving modes.

3 Status of planned activities

The lane-keeping function has been effective as a single module, but it still faces some issues when integrated into the ROS simulator. Our team has been actively working to resolve various bugs in the simulator. (Details are on the forum.)

The lane-keeping function has demonstrated stability in isolated testing, but occasional control command delays have arisen during integration into the simulation system. Progress has been made in debugging the simulation environment, with the foundational communication framework now established, though compatibility errors caused by configuration discrepancies remain under resolution.

4 General status of the project

The project progress has not yet recovered. We have tried multiple environments and development approaches, but the results have been less than satisfactory.

Project progress lags behind the original schedule, primarily due to the technical complexity of debugging deeply coupled modules within the simulation system, environment-dependent issues that trigger sporadic faults, and resource allocation challenges that have extended resolution timelines for critical tasks.

5 Upcoming activities

To accelerate progress, the team will refine the existing code architecture by referencing design patterns from mature open-source projects, initiate hardware co-development by deploying members to validate control consistency between simulation and physical vehicles, and proactively mitigate risks by preparing contingency plans for known technical bottlenecks.