Automated Driving Using External Perception

Team: Ronit Hire, Dhanesh Pamnani, Shreyas Jha, Jash Shah, Atharv Pulapaka

ILR08

11th October, 2023

Sponsor: Nissan Automotive Inc.
Liam Pederson, Najam Baig, & Viju James

Table of Contents

1. Individual Progress

1.1 Cruise control

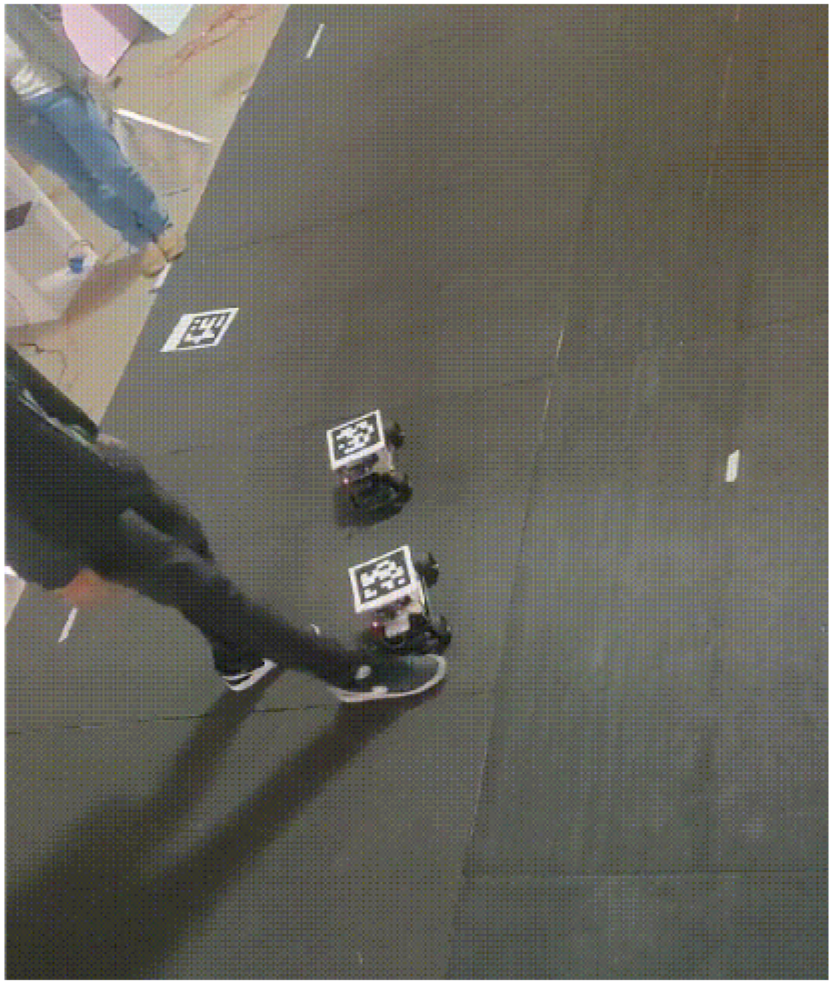

Figure 1 : Integrated test with 2 cars

In our project, I implemented adaptive cruise control for RC cars to ensure the controlled vehicles maintain a safe distance from each other, minimizing the risk of collisions. This feature acts as a supplementary layer of safety above the trajectory follower (Model Predictive Control - MPC) already in place.

The system operates by using data from the perception unit, which provides the position information of both controlled vehicles to the RC car's control system. If the system detects another RC car in front and the distance between the two vehicles falls below a predefined safe distance, the adaptive cruise control is activated.

The control system computes an error value, 'e,' which represents the difference between the current distance between the two vehicles and the safe distance. Using a simple control equation, we calculate 'eddot' based on the proportional (kp) and derivative (kd) gains. The 'eddot' represents the required change in acceleration to adjust the vehicle's velocity accordingly. Figure 1.2 shows the equation.

$$\ddot{e} = Kp * e + (Kd * \dot{e})$$

Figure 1.2 : PD control for adaptive cruise control

This adjusted velocity is then provided as a new speed command to the MPC of the particular vehicle. The MPC system takes this information and dynamically controls the steering and speed of the RC car to ensure it avoids a collision with the vehicle in front.

Extensive testing and evaluation were conducted to validate the performance of the adaptive cruise control system. The results demonstrated that the system effectively maintains a safe following distance and provides an additional layer of safety for the RC cars, minimizing the likelihood of collisions. Figure 1.1 shows a success test where the cars do not collide with each other.

The implementation of adaptive cruise control in our project serves as a vital safety feature for our RC cars, guaranteeing that they do not collide with other controlled vehicles. This technology enhances the overall safety and reliability of our autonomous RC car system.

1.2 Drive rc cars in a loop

During our testing phase, which included driving RC cars in a loop—a scenario previously unexplored in our trials—we encountered an issue. The RC cars displayed an interesting behavior: while they executed sharp left turns effortlessly, they faced difficulties executing equally sharp right turns when following the same waypoints.

To understand the root cause, we began by examining the control commands sent to the cars during both left and right turns. We found that the steering commands were either similar or more for right turns than for left turns. This indicated that the issue was not related to the control commands but rather stemmed from the physical characteristics of the vehicles themselves.

We then decided to isolate the low-level control of the RC cars from the rest of our subsystems. We put a measuring tape on the floor, using the starting point as the reference (0 measurement). We then instructed the RC cars to perform full left and right turns, meticulously recording the distances at which the cars intersected the measuring tape.

We observed that the RC cars had a turning radius of 0.75 meters during left turns, while they required a larger 1-meter radius for right turns. These discrepancies indicated that the challenge was rooted in the low-level control of the RC cars.

To address this issue, we delved into the Electronic Speed Controller (ESC) of the RC cars and identified the limitations of the commands sent to the motors. We proceeded to recalibrate these limits, ensuring that the cars received commands that allowed them to execute right and left turns with equal ease, each with a 1.5-meter diameter.

After making the necessary adjustments, we resumed our loop testing. This time, the RC cars exhibited remarkable performance, navigating the loop with precision. They executed sharp right and left turns with ease, underscoring the successful resolution of this challenge. We tested by running 2 controlled vehicles in a loop along with cruise control in action as described in the previous section.

1.3 Testing with fuse data

In this testing pose from camera was fused with rc car odometry and IMU data to give more accurate pose estimates. These pose estimates were used by the trajectory follower(MPC) to control the rc cars. We observed improved and robust performance of the vehicles.

2. Challenges

2.1 MRSD project

- Challenges as described in the previous section was to ensure the vehicle was able to take right and left turns accurately. As discussed in the previous section with a larger turning radius for right turn the vehicle was going out of the view of cameras and therefore this issue had to be resolved. On tuning and changing the limit of lower level control we were able to resolve this issue.
- With pose estimation coming solely from camera we identified issues when the vehicle comes to a stop. Since the current heading is estimated based on the vehicle velocity in x and y direction, when the vehicle is stopped the heading estimate is 0. This then causes the controller to not behave expectedly. This was resolved by fusing IMU data with camera poses to get a more accurate pose estimates for all scenarios.

3. Team work

Ronit Hire: Ronit worked on ensuring perception units are able to detect and track multiple cars. He generated unique IDs for each vehicle that helped the control system to associate data with respective cars. He is working on eliminating dependence on aruco markers for vehicle detection. He is developing a perception system that can correctly identify and track multiple cars.

Shreyas Jha: He worked on getting data from VESC which is the ESC used inside RC cars, this data has information about odometry from the inbuilt IMU. He then fused these two data sources with camera pose estimates to get more accurate pose estimation. He also worked in resolving the low level RC control issue.

Dhanesh Pamnani: Dhanesh has initiated research on planning subsystem and coming up with algorithms that suit our requirements. He has started developing a planning subsystem that can handle static, dynamic obstacles and behavior at intersections.

Jash Shah: Jash is working on the planning subsystem, he is developing a planner that will send center line trajectory for the RC car to follow using the occupancy grid for defined for our environment.

4. Future Plan

- The next step is to develop and implement behavior of RC cars at intersections. Followed by thorough testing to ensure robustness. While simultaneously working on a planning system to handle static and dynamic obstacles.