Automated Driving Using External Perception

Teammates: Ronit Hire, Dhanesh Pamnani, Shreyas Jha, Jash Shah
ILR07
27th September, 2023

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

THE ROBOTICS INSTITUTE
Carnegie Mellon University

Table of Contents

1. Individual Progress

1.1 Independent MPC integrated with perception system to drive 2 cars

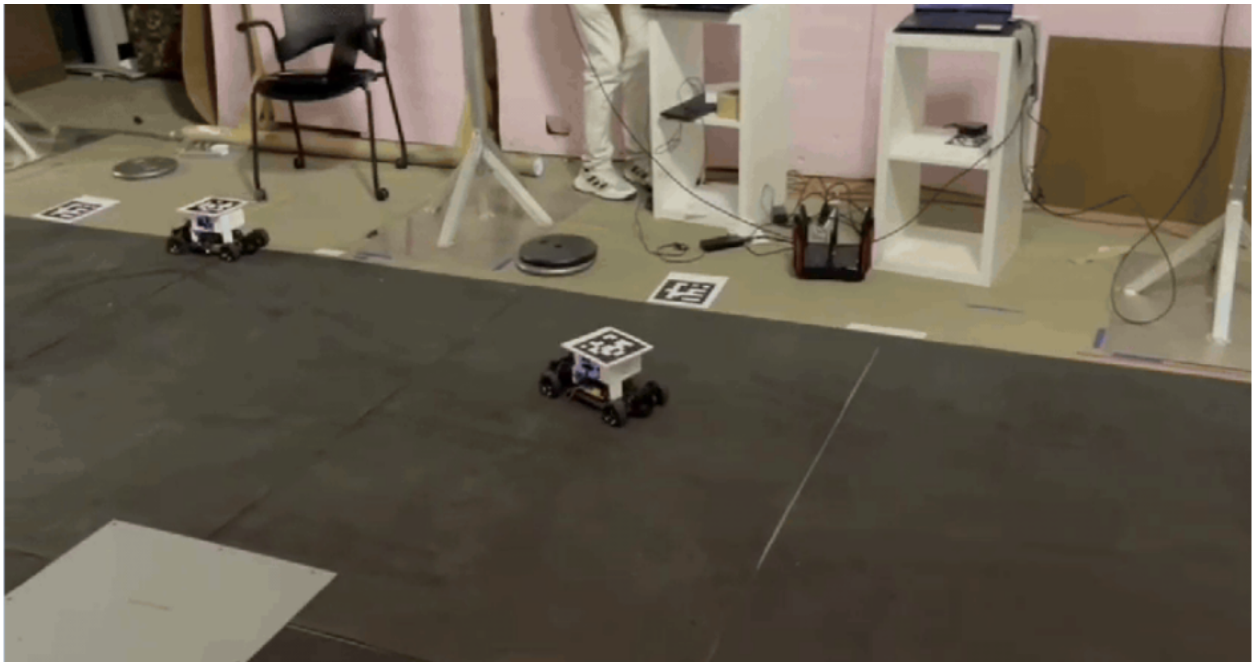

Figure 1 : Integrated test with 2 cars

For effective control of multiple cars, I established individual Model Predictive Control (MPC) controllers for each vehicle. This involved having two independent MPC controllers for each vehicle along with their own tunable parameters. I designated distinct topic names for transmitting steering and acceleration commands to these cars and carried out testing within simulation environments, where it demonstrated successful operation. The next phase was to integrate with the perception subsystem. Here, the focus shifted to acquiring pose and velocity data from the perception system and accurately associating this information with the respective RC cars. Once identified, the MPC controllers operated independently on the state information of each car and generated control commands tailored to their specific requirements. This methodology has proven effective in optimizing control strategies for multiple vehicles while maintaining autonomy and independence in decision-making. However, no part of keeping safe distance between controlled cars was implemented and the cars could collide with each other. Figure 1 shows an integrated system running two cars by executing independent MPC using data from perception units.

1.2 Attempt to add distance constraint in individual MPC

While working on improving our Model Predictive Control (MPC) system, we encountered a challenge when trying to include distance constraints. The goal is to ensure controlled cars do not collide when driving one behind the other and maintain a safe distance between them. To address this, position of one car will be given to the 2nd car MPC and distance constraint will be formulated based on positions of both cars. Our existing MPC setup was designed with a straightforward quadratic cost function, which worked well with the OSQP solver from the CVX library. However, introducing distance constraints made the problem non-convex, which complicated matters. I tried different approaches to express these constraints within convex bounds, but the results were not reliable in simulations.

We also considered using non-linear solvers like IPOPT to handle the non-convexity issue, but after reviewing the complexities involved in restructuring the code, we opted for a more practical approach.

Instead, we decided to focus on implementing cruise control techniques to ensure that two vehicles maintain a safe distance between them. After achieving this, we plan to work on enhancing the planning subsystem to avoid collisions at intersections. This approach allows us to address immediate safety concerns while considering a more straightforward path for improving the control system.

1.3 Restructuring control system code

In the new approach, we're improving efficiency by having the planner send the desired trajectory periodically, anticipating future actions. The Model Predictive Control (MPC) now runs at a fixed rate, regardless of when trajectory updates arrive. This makes control commands more consistent and responsive to the planner's predictions, optimizing computational resources and coordination between systems.

1.4 Basic planner skeleton code

I established an initial code structure for the planner subsystem in C++. This code forms the framework for future planner development. It takes in user defined waypoints and interpolates them. Based on the current vehicle position it sends out trajectory for next n seconds to the follower. The system operates as a ROS publisher-subscriber architecture, functioning at a defined frequency..

2. Challenges

2.1 MRSD project

- Challenge was to incorporate distance constraint to current MPC formulation. Different ways to include this constraint and at the same time keep the convex nature of formulation were attempted. Decision was made to go ahead with simple cruise control and planning subsystem to handle decision making.

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. We then tested the integrated system - perception and control by sending following our SVD encore track.

Shreyas Jha: Shreyas has taken the responsibility to develop the third RC car. He has also worked on getting the jetson ready to transfer camera live feed camera data from infrastructure unit to perception system. 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. This data will act as an additional source of information to associate data with respective cars.

Dhanesh Pamnani: Dhanesh has initiated research on planning subsystem and coming up with algorithms that suit our requirements.

Jash Shah: Jash helped Ronit in scaling the perception pipeline to track and detect multiple cars. He also worked on setting up the Gazebo environment to test and run the planning system that needs to be developed.

4. Future Plan

- The next step is to develop cruise control for each car. A straightforward PID controller will be implemented to start with and then add improvements later. Carry out testing on simulation and on hardware.