OuterSense

Automated Driving Using External Perception

Teammates: Ronit Hire, Dhanesh Pamnani, Shreyas Jha, Jash Shah

ILR03

13th 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 MRSD Project: Team OuterSense

During this phase of work, I undertook a review and validation process for the Model Predictive Control (MPC) codebase, addressing a four-month gap in my involvement. I initiated the process with a code review, followed by testing in a simulation environment to ensure alignment with intended operational performance. Subsequently, I analyzed the simulation results, confirming the vehicle following the desired trajectory.The next step was to have a Hardware-in-the-Loop (HILS) testing, where the code was tested using a physical RC car and simulated sensor outputs. The system could send steering and acceleration commands to the vehicle based on the simulated sensor inputs. Finally, I conducted on-track testing, integrating a perception unit to provide pose data. The system performed accurately, with the vehicle following the predetermined trajectory as demonstrated in the Spring Validation Demo(SVD) Encore. These assessments affirmed the code's robustness and its compliance with performance standards. No further tuning of the controller was required as the vehicle followed the given path meeting all performance requirements set for SVD. Figure 1 shows track used in SVD encore which was also used to test the system in simulation.

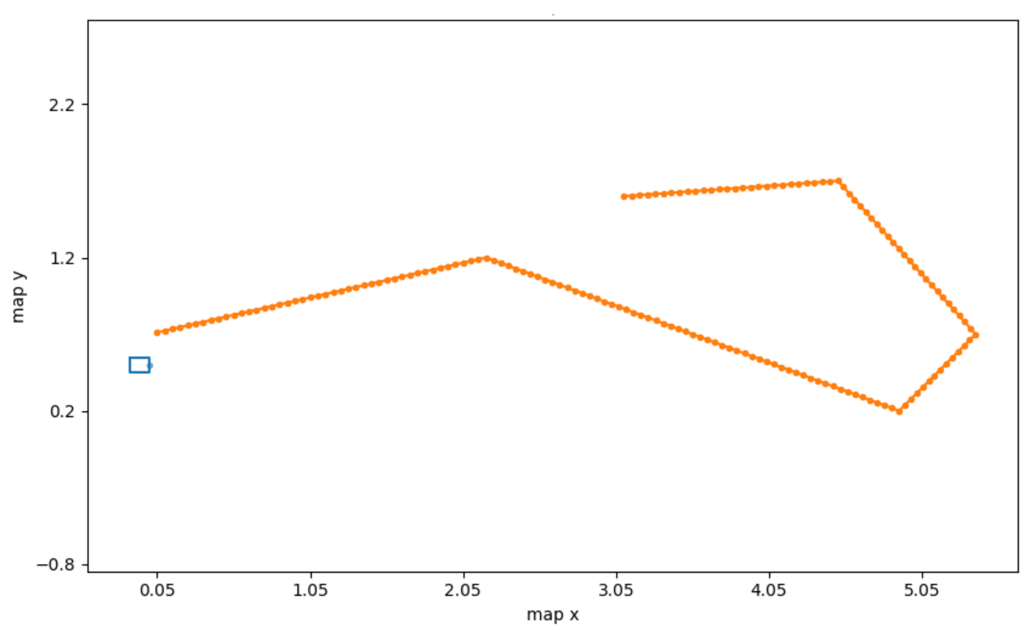

Figure 1 : Test track for SVD encore

In the further development of the project we plan to scale the system by adding multiple cars and aim to extend our Model Predictive Control (MPC) system to manage multiple RC cars. I explored two different approaches.

The first approach involved keeping separate MPC controllers for each car. I created two distinct MPC instances, each configured independently with its own settings. To manage communication, I assigned unique topic names for control messages, such as steering and acceleration commands. This approach was tested successfully in simulations and is currently operational.

The second approach sought to consolidate the control system into a single MPC for both RC cars. This required significant code changes. I combined state information for both cars, including position, heading, and velocity, into a single state vector. Similarly, the desired state, containing trajectory information for both cars, was combined into one desired state. The MPC formulation was adjusted to handle multiple cars, including adapting the cost function and constraints. This implementation is under simulation testing. This approach is helpful to control multiple cars and have constraints which will ensure these controlled cars do not collide with each other. Moreover, there will be only one instance of MPC running for multiple vehicles.

The choice between these approaches will depend on project requirements and timeline. The single MPC solution shows promise, but the independently running MPCs provide a proven method for controlling multiple RC cars. Further testing will guide our decision on which approach to pursue.

The next crucial step in our project involves the integration of our control system with a planner system. To facilitate this integration, some restructuring of our existing trajectory follower module was required.

In the current implementation, the Model Predictive Control (MPC) module is triggered with each pose update, and it takes the required computational time to generate control inputs. Additionally, the trajectory used by the MPC is fixed, containing information about the entire track.

In the updated design, we are shifting to a more efficient approach. The desired trajectory will be received at a predefined frequency from the planner system, which anticipates a few seconds into the future. This change allows for a more streamlined coordination between the planner and the control system. The MPC will execute at a set frequency, regardless of when the trajectory updates are received. This synchronized approach ensures that control commands are generated consistently, aligning with the planner's predictions.

This restructuring offers several advantages. Firstly, it standardizes the frequency of publishing control commands and integrates seamlessly with the planner. Secondly, it optimizes computational resources by avoiding redundant MPC executions, making the control system more responsive to the planner's directives. Lastly, it ensures that pose updates, which typically occur more frequently, are effectively utilized within this updated framework.

2. Challenges

2.1 MRSD project

- The light in the basement level had a glare issue which caused the perception system to give inaccurate readings, we resolved that by partially covering the light.
- We also resolved a suspension issue with one of the RC cars, currently we are building a third RC car that can be controlled.
- We were dependent on laptops to run the perception pipeline as they were the edge compute for each infrastructure unit. We have added jetson to resolve this dependency.

3. Team work

Ronit Hire: Ronit is currently working on scaling the perception pipeline to detect and track multiple cars. He is also working on ensuring pose estimates for all cars are accurate even in the handover region of these infrastructure units

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

Dhanesh Pamnani: Dhanesh worked on getting the set up ready to replicate the SVD encore demo. He is currently designing and building the new track which is required for the fall validation demo. He is also handling the task of procuring the items required for the project.

Jash Shah: Jash is helping Ronit in scaling the perception pipeline to track and detect multiple cars. He is also working 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 run the entire integrated system with multiple rc cars.
- Work on the planning system to ensure both controlled cars do not collide with each other and keep certain between them.