Automated Driving Using External Perception

Individual Lab Report

MRSD Project Course II

Fall 2023

Shreyas Jha

Team E: OuterSense

Teammates: Atharv Pulapaka, Dhanesh Pamnani, Jash Shah, Ronit Hire

ILR10

Nov. 17, 2023

Table of Contents

Individual Progress

Over the course of this project, I have contributed mainly to the RC car subsystem, communications and time-synchronization, state estimation, data association, and central GUI for the OuterSense autonomous driving system. In the past two weeks as we prepared for our final demonstration, I worked on stress tests for individual units, sub-systems and integration that are described below.

RC Car Subsystem

I invested time into procuring components and materials for assembling the 3rd RC car including the chassis, motors, electronic speed controllers, batteries, servo, Raspberry Pi, wiring harnesses, connectors and mechanical parts. Meticulous care was taken while assembling the vehicle, soldering connectors, crimping wires, installing the compute and routing cables securely. Extensive unit and integration testing was then conducted. This was done in collaboration with Dhanesh and Atharv.

For low-level control validation, I varied the proportional and derivative gains of the VESC controller in small increments and measured the rise time taken to reach 90% of the commanded RPM using a contactless tachometer. A rise time below 100ms was finally achieved for tracking both high and low RPM setpoints precisely. This enables smooth acceleration and braking.

To validate slow speed control, I gradually kept decreasing the velocity setpoints sent to the RC cars in small steps and observed the motion profiles using manual measurements. Stable tracking was verified down to extremely slow speeds of 10cm/s which is crucial for low-speed maneuvers. Sudden stop-and-go commands were also tested successfully.

In order to tune the minimum turning radius, I adjusted the steering servo trim, max angle, PWM limits and reversing switches iteratively. Multiple U-turn tests were conducted on courses of varying widths by commanding a 180-degree steering angle. A turning radius below 1 meter was finally attained, allowing the cars to smoothly navigate the tight tracks.

To stress test the speed controller and fail-safes, I deliberately issued velocity commands that exceed defined limits. The RC cars stabilized at the specified max speed and handled exceptions robustly proving the effectiveness of the overridden.

Communications and Time Synchronization

I invested effort into configuring Chrony, tweaking the tracking parameters and polling intervals to achieve clock synchronization with the server within 1ms tolerance across all devices. This involved an iterative process of reviewing Chrony statistics, identifying issues, and refining the Chrony configuration.

I scripted the process to automatically identify the server IP, disable external NTP sources, and execute Chrony configuration whenever a new device joins the network. This saved tedious manual effort every time a new member needed to be synchronized.

Rigorous validation was done by continuously monitoring the Chrony tracking statistics, source statistics, and querying the Chrony clients from the server. This verified that all distributed subsystems across team members maintain clock synchronization with the server down to the millisecond level.

To thoroughly characterize communication latency, I measured the delay between perception camera capture and cloud processing as well as the lag between cloud publishing commands and vehicles executing them. Bottlenecks were identified in wireless transmission and serialization.

State Estimation

Significant effort went into tuning the process and measurement noise covariances in the UKF after incorporating filtered yaw estimates from perception. I iteratively adjusted the relative uncertainties allocated to each modality based on observation. Localization accuracy versus responsiveness trade-offs were evaluated to strike an optimal balance.

By comparing raw IMU orientation estimates against the Madgwick filtered values, I was able to reduce errors by ~40% and avoid drift in absolute orientation measurements. This was achieved by tuning the filter's beta parameter. A proper accelerometer calibration was done using the VESC tool software.

Extensive testing showed that vehicles can (almost) reliably stay within their lanes for 2 track circuits. This validated robustness to short perception failures.

GUI

I integrated useful visualizations in RVIZ including the planned paths, predicted MPC trajectories, transformed coordinate frames, raw vs fused state estimates. This proved invaluable for at-a-glance situational awareness and debugging during integration. Figures 1 and 2 below are depictions of some visualization we use to monitor the system.

I thoroughly tested the E-Stop button by issuing motion commands to validate that it overrides and stops vehicles as expected. The indicator color and text change provide intuitive feedback. I also artificially induced failures in software components and verified robust telemetry monitoring.

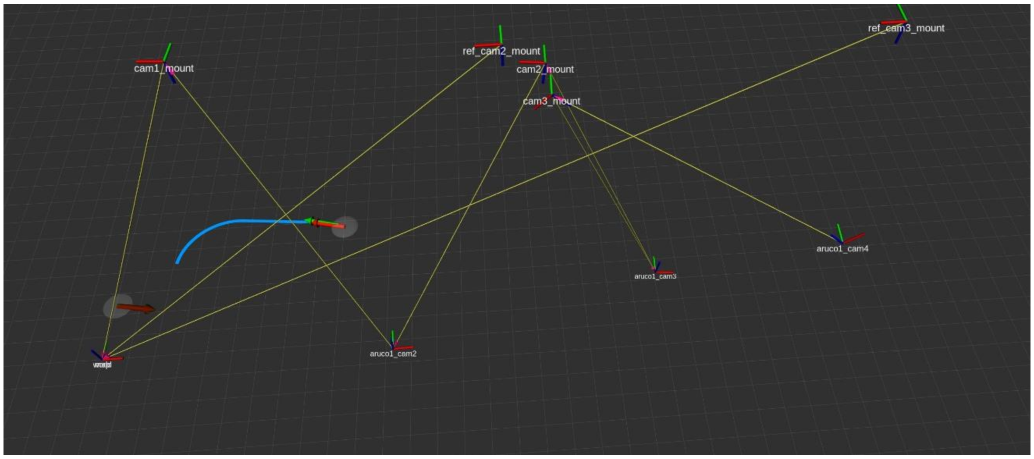

Figure 1: Calibration & handover visualization

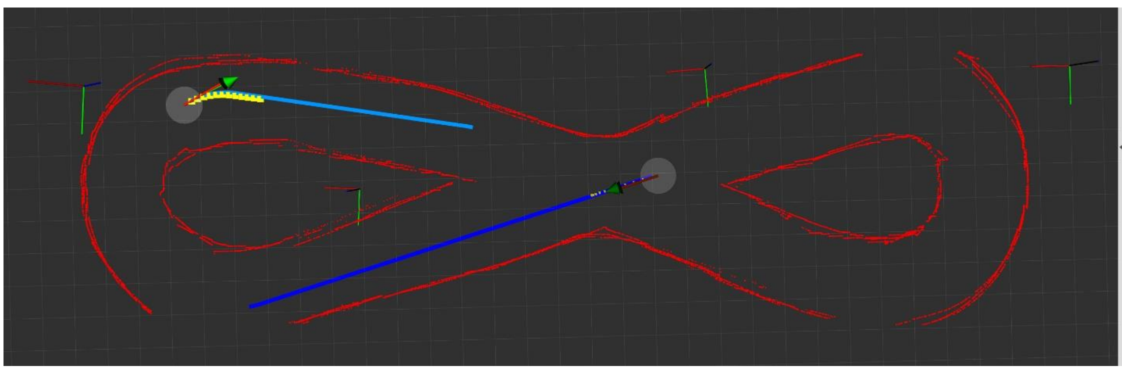

Figure 2: Integrated autonomy visualization

Challenges

Tuning the state estimation module

Tuning the state estimation module without clear knowledge of sensing uncertainties or localization ground truth posed significant challenges. Originally, no modality published associated uncertainty estimates. This required calculated guesswork to map relative uncertainties. We aggressively correct pose based on perception, but its latent data needs appropriate filtering before ingestion.

The recent update that incorporated yaw from perception into the UKF necessitated extensive re-tuning. With no defined covariance from perception, I had to iteratively adjust process and measurement noise parameters related to yaw. This involved tweaking IMU yaw covariance, perception covariance, process noise, and odometry angular velocity covariance. Finding the right balance through observation was tricky without uncertainty models or ground truth.

Although the integration is robust currently, opportunities exist to further tune the filter for better accuracy. This will enable higher speeds and graceful handling of latency and dropouts. The lack of sensing uncertainty knowledge and localization ground truth references made state estimation tuning challenging.

Teamwork

Atharv Pulapaka: Atharv worked on testing the adaptive cruise control system to avoid collisions between controlled vehicles and maintain a safe distance between them. He worked to make the behavior at intersections robust. We also worked together to setup the wiring and electrical connections of the third RC car.

Dhanesh Pamnani: Dhanesh focused on designing and developing the planning subsystem. He built a local map using collected data and tested the implementation of the planner in simulation.

Jash Shah: Jash collaborated with Dhanesh on the planning subsystem. They have implemented a hybrid A* planner that incorporates kinematic constraints for feasible Ackermann steering. Jash also worked on integrating the planner with the rest of the sub-system and managing it for multiple RC cars.

Ronit Hire: Ronit worked on enabling the perception system to handle different markers simultaneously. He also worked together whilst integrated system testing and helped debug issues that came up in other subsystems. Apart from this, Ronit continued his research on Minimum Cost Flow graph algorithms to maintain stable object IDs in multi-object tracking.

Future Work

I aim to explore JPDA for data association and incorporate latency simulation in the GUI. My goals are expanding GUI functionality to monitor system latency and completing integrated testing of data association using joint probabilistic techniques.