Automated Driving Using External Perception

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

ILR03

22nd March, 2023

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

Table of Contents

1. Individual Progress

1.1 MRSD Project: Team OuterSense

ROS Architecture: Transferred entire python implementation into a ROS architecture where there is a state estimator node and a MPC node that publishes and subscribes information. Refer figure 1.1.1 for ROS architecture. The MPC node publishes control commands containing steering and acceleration values that are subscribed by state estimator node. The state estimator just has an ODE running on the Ackermann model of the vehicle which takes in these U inputs (steering, acceleration) and generates state estimates of the vehicle. The state estimator node then publishes these updated states which is used by the MPC. This process runs till the vehicle reaches its goal state.

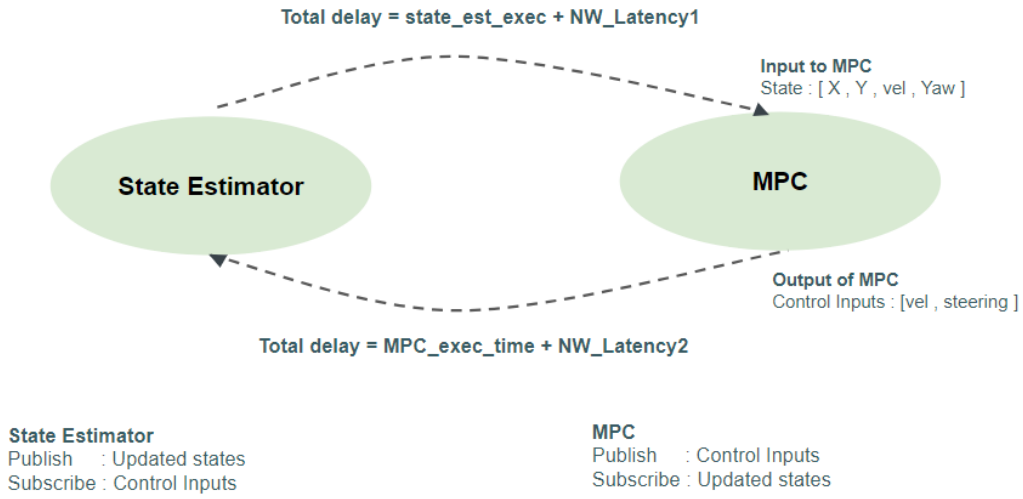


Figure 1.1.1: ROS architecture

Offset calculation: The external perception will have noisy estimates and the MPC will work with these data. The RC car should still remain within the lane boundaries. To simulate this, a random offset is added to current state estimates and given to MPC. The plot of MPC simulation shows the lateral offset to be less than 5cm on a straight road which is acceptable. Lateral offset is calculated considering yaw of the vehicle as well. The figure 1.1.2 shows how lateral offset is calculated. Refer figure 1.1.3 for MPC simulation plots.

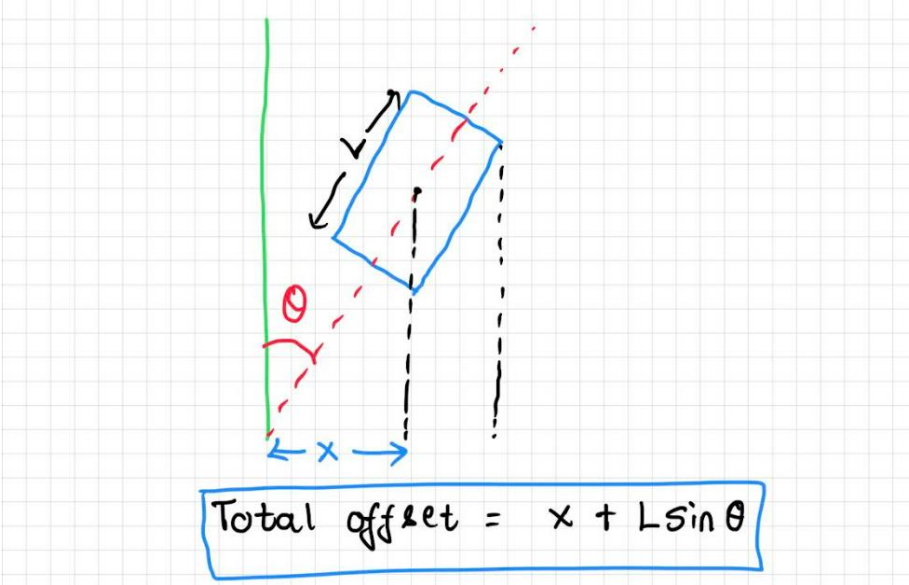


Figure 1.1.2: Lateral offset calculation

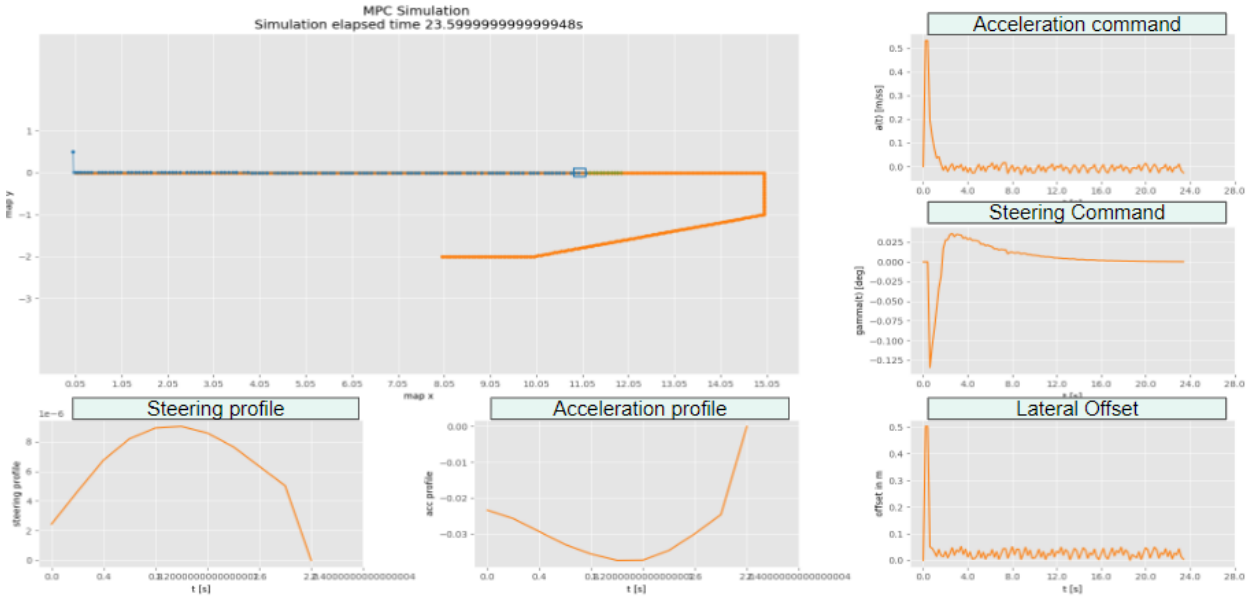


Figure 1.1.3: MPC simulation

Generate acceleration and steering profile: The MPC outputs a series of input for steering and acceleration. The profile ensures when communication is lost between controller and RC car the RC car will follow this profile and come to a stop. The profile can be seen in Figure 1.1.3

Hardware in loop testing (HILS): The next step is to control the RC car using the output of MPC. All the simulation is over multi machine ROS network. The computer executes the MPC with simulated state estimates and sends command to RC car. A mapping code is written on Raspberry pi of RC car which maps the steering commands to duty cycle commands of servo motor. The communication is all wireless over the router set up by us.

A random complex track is given as input to test the MPC performance and observe RC car steering. Figure 1.1.4 shows the set up. Also find below link with video of demo

https://drive.google.com/file/d/1b8EjrzBCdl0zQHf6CnRtMcJpzEdw7l2l/view?usp=sharing

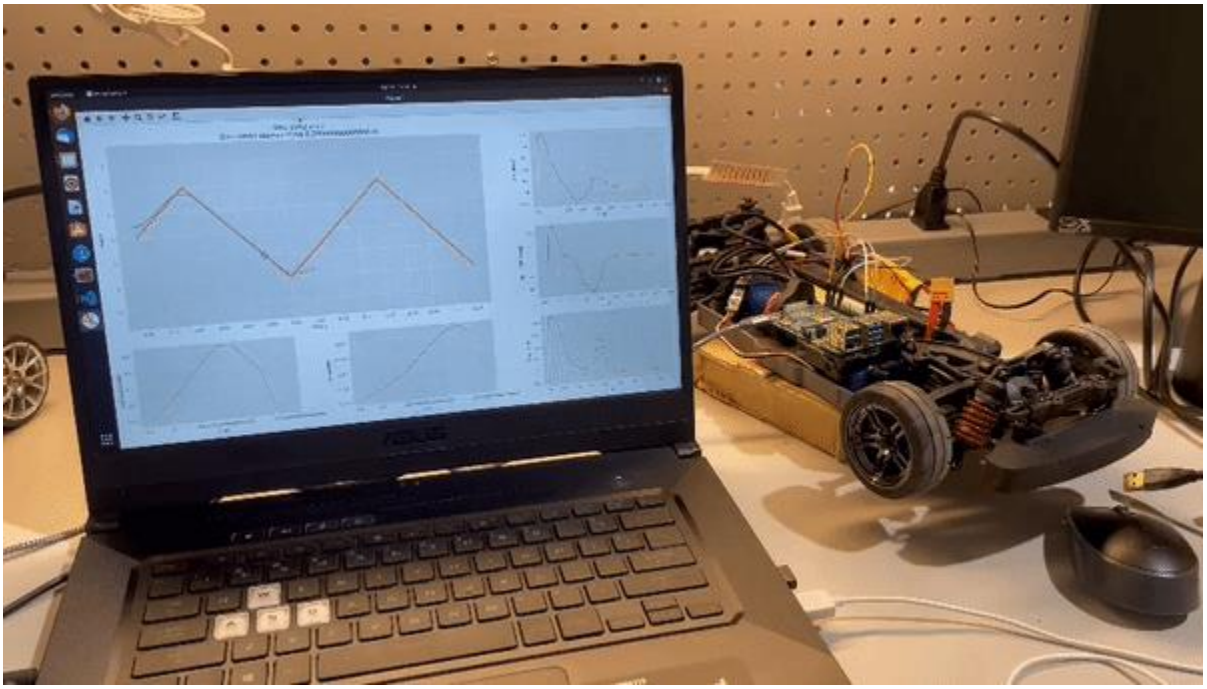
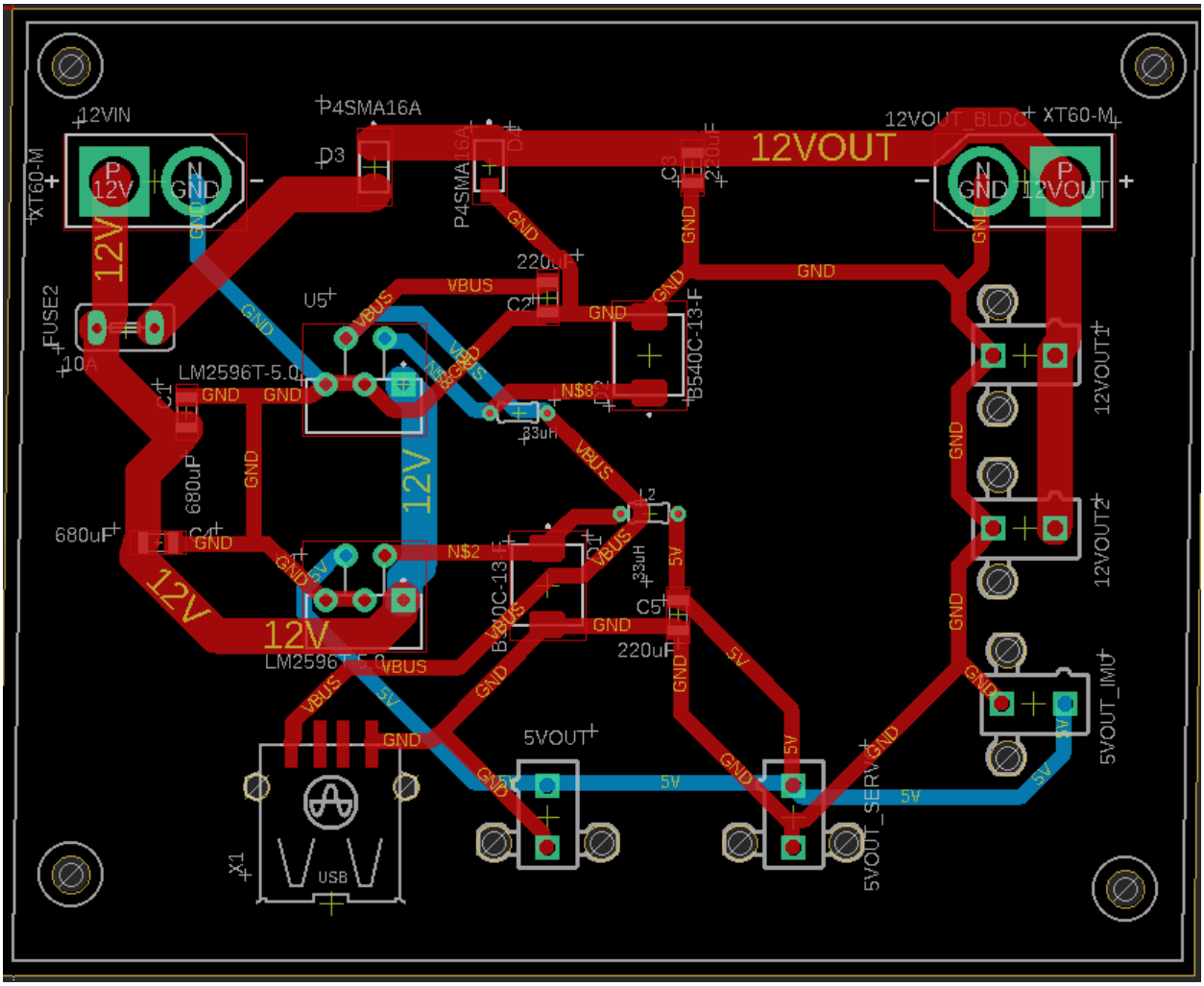


Figure 1.1.4: HILS with MPC

Power Distribution Board: Modified the PDB schematic design based on feedback and developed a layout for PCB board. Refer figure 1.1.5 for layout of PDB



2. Challenges

2.1 MRSD project

- The main challenge was to synchronize simulation time with real world time. This was required to test HILS correctly.
- The other challenge was to accurately simulate latency behavior. The latency will be because of state estimation from the camera. The perception unit will compute and send the state estimate and the MPC will work on this state estimate. But at the instant when the state is computed and sent to MPC the car has moved and is now at a different state and the MPC will work on the older state. Simulating this behavior is a bit of a challenge.

3. Team work

Ronit Hire: Ronit is working on implementing another control approach where he is developing PID control to solve the problem. He has simulated some results to show behavior of PID with different latency values. He also working on the perception front where he will be finding velocity of the RC car.

Shreyas Jha: Shreyas is working on building the RC car and teleoperate it using laptop. He has developed a ROS architecture to send and receive data between RC car and laptop. He is working on synchronizing time over multiple machines operating on ROS.

Dhanesh Pamnani: He is handling the manufacturing side of the track and is working in the machine shop to build the infrastructure units. He is also working on the perception front with Jash. He has completely finished building the infrastructure unit, ensured stability and robustness in all the units. He has also worked on camera calibration and validation of the depth and RGB camera. He is currently working on finding lane detection and lane deviation of RC car.

Jash Shah: Tested robustness of marker detection by mounting camera on different heights. He has also calibrated the camera for detection and found the deviation of the RC car from lane boundary.

4. Future Plan

- The next step is to run the RC car using the acceleration and steering commands from the MPC. Proper acceleration mapping with esc commands needs to be done to ensure correct speed of RC car is met.
- I will have to collaborate with perception team to use data from camera and give that as input to MPC to test MPC results. Also observe latency and noisy estimate effects in performance.
- Solder and test PDB which will be mounted on RC car.