OuterSense

Automated Driving Using External Perception

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

ILR02

Feb 15th, 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

My focus related to development of the MRSD project so far has been towards the control system architecture. I started off the semester by researching control algorithms to keep the RC vehicle in the lane while it is following the desired trajectory. I finalized on Model Predictive control approach for implementing lane keeping for the vehicle. I did a thorough study to understand the working and architecture of MPC before implementing. I studied research papers and articles for MPC implementation along with watching series of videos on MPC design and implementation.

I started by implementing MPC for lane keep on Simulink model. To get a better understanding I designed a dummy track on Driving scenario designer on MATLAB and analyzed the behavior of the vehicle on the track. For the vehicle to stay within the lane the MPC needs two inputs mainly the lateral offset of the vehicle from the center line and the deviation in yaw of the vehicle with respect to the center line. These two inputs are calculated based on the sensor data collected. The goal of the project is to use external perception, since the sensor is not mounted on the vehicle there will be some offset and delay in computing the sensor data. Hence, one of the objectives in the simulated environment was to simulate the MPC behavior when there is an offset in lateral deviation and yaw angle measured from the sensor and also observe the effect when there is a delay in sending/receiving data from MPC controller. Certain blocks on Simulink were added to generate offset/ noise on data to simulate errors in sensor reading, a delay block was added and the loop rate (sample time) of MPC controller was varied to simulate delay in MPC computation. All the results were gathered and analyzed.

Refer figure 1.1 for Simulink model and table 1.2 for comparative study of different experiments.

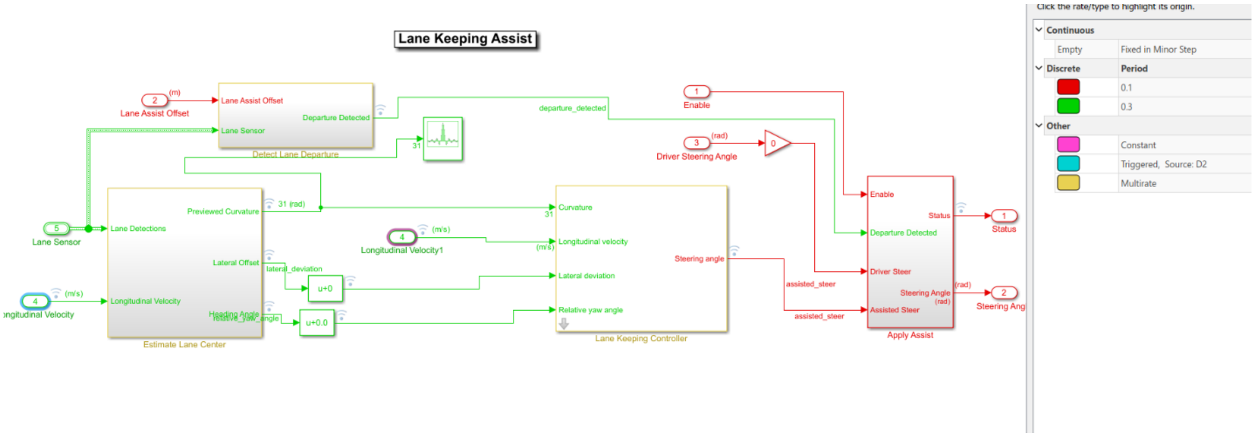


Fig 1.1 : Simulink model

Perception block output sample time (sec)	MPC sample time (sec)	Lateral deviation bias (m)	Result
0.1	0.1	0	Smooth trajectory
0.3	0.1	0	Smooth Trajectory
0.5	0.1	0	Deviates from lane completely
0.1	0.1	0.01	Smooth trajectory
0.1	0.1	0.05	Smooth trajectory
0.1	0.1	0.08	Deviates from lane completely
0.3	0.1	0.01	Smooth trajectory
0.3	0.1	0.05	Smooth trajectory
0.3	0.1	0.08	Deviates from lane completely
0.5	0.1	0.01	Has jerks, no deviation from lane
0.5	0.1	0.05	Deviates from lane completely

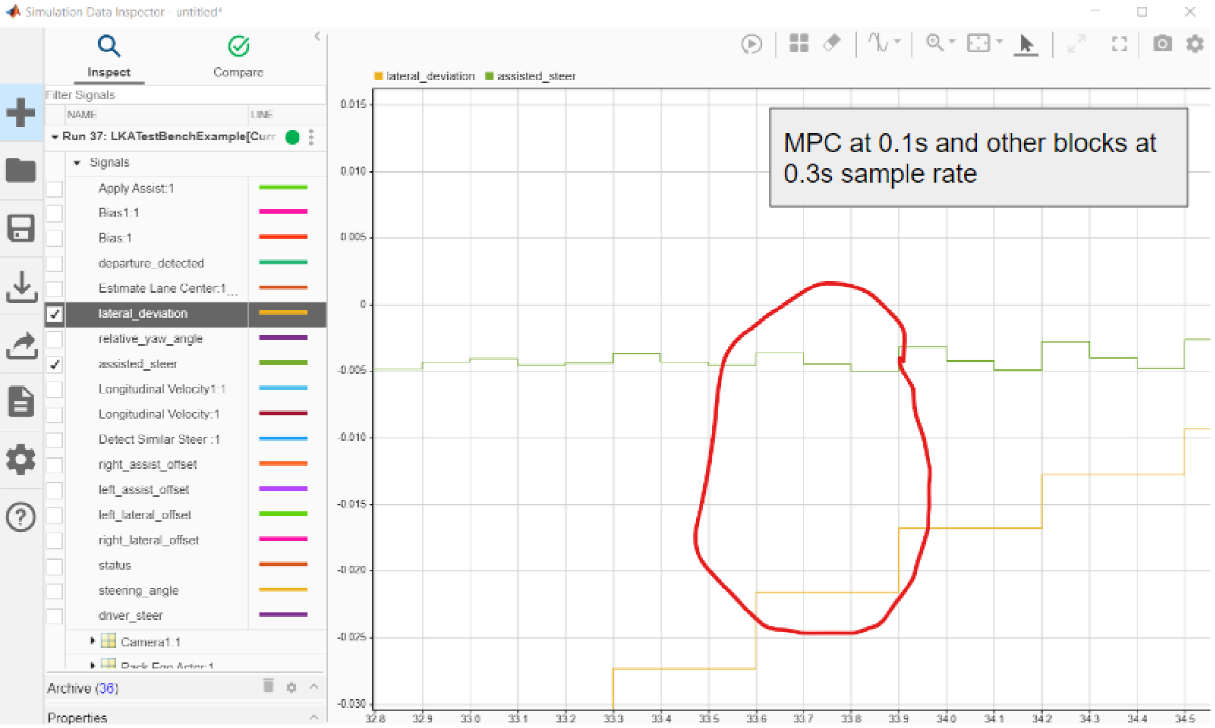
Table 1.2: Comparative results

In Simulink model, sample time for the block which computes the lateral deviation and yaw angle is increased. This is to simulate the delay in computing from perception unit and lags due to communication. The goal is to simulate the vehicle behavior and MPC performance when accounting for these delay.

It was observed that when the sensor unit and lane estimator computation was done at every 0.3 seconds and MPC operates at every 0.1 seconds, with a bias of 0.05m in lateral deviation still resulted in a smooth trajectory. On increasing these sample time and bias, jerks were observed in the trajectory and the vehicle deviated from the lane.

Another parameter that can be entered in the lane keeping block (MPC block) was to consider lags due to communication and sensor computation. This lag is considered in the MPC optimization and gave the control inputs to minimize the errors.

In a MPC it calculates the new set of control outputs (number depends on the control horizon defined) each time step and sends only the first steering (output) value. If we define the sample time of MPC to be 0.1 seconds, every 0.1 seconds it computes steering values but only sends the first one. On analyzing the data of the control outputs from the MPC controller I could observe that during the time when the controller is not getting the inputs from the sensor, it sends the next control output it calculated. Figure 1.3 shows the output



MPC at 0.1s and other blocks at 0.3s sample rate

Fig 1.3: MPC sending next control outputs till it gets new data

2. Challenges

2.1 MRSD project

The main challenge I faced was understand the MPC block on MATLAB. It was difficult to understand how it was implemented and how the output was affected on changing few design parameters. I watched Mathwork video series on MPC and on Simulink to get comfortable with MPC controller on Simulink. I also studied all the Simulink examples on MPC implementation.

Another challenge was to simulate the two scenarios – having offset in sensor data and delay in computation. After lot of researching, I added noise and delay blocks but I was not getting satisfactory results. I consulted with my MRSD seniors on how to approach this problem and then implemented constant bias to sensor readings to generate offset and changed the sample time of MPC controller to simulate the delay in MPC computation.

3. Team Work

Ronit Hire: Towards development of MRSD project Ronit is working on the perception front, the work which he is implementing will be given as input to my control system architecture. He is also handling the logistics for electronic and mechanical components. I collaborated with him to analyze the data generated from the MPC block.

Shreyas Jha: He tested the BLDC motor and VESC to control the motor, he then mounted them on the RC vehicle along with controller. He also integrated the ESP wifi module where he could send steering and speed commands to the controller from the phone to teleoperate the vehicle.

Dhanesh Pamnani: He is handling the manufacturing side of the track and is working in the machine shop to build the infrastructure units. He has completed finishing one infrastructure unit.

Jash Shah: He is working on the perception front and has completed table tests for detecting objects and finding their orientations.

4. Future Plan

- The next step for me towards development of MRSD project is to implement MPC controller in C++ / python as well as test the controller behavior. We extracted as much data points as we could from Simulink and now we will be experimenting other methods of simulation.
- Develop the PCB for our power distribution board
- I also plan to help in testing components of RC car and assemble it.