OuterSense

Automated Driving Using External Perception

Individual Lab Report - ILR03

March 2, 2023

Team E - Outersense

Author:

Ronit Hire

Team Members:

Atharv Pulapaka

Dhanesh Pamnani

Jash Shah

Shreyas Jha

THE ROBOTICS INSTITUTE

Carnegie Mellon University

Contents

1 Individual Progress **1**
1.1 Controls . . . 1
1.1.1 F1-tenth simulator . . . 1
1.1.2 PID control . . . 2
1.2 Infrastructure unit . . . 2

2 Challenges **3**
2.1 PID control . . . 3
2.2 Assembling infrastructure unit . . . 3

3 Team Work **3**

4 Plans **4**

1 Individual Progress

After the last progress review (PR), I took a more focused approach towards Controls and also contributed towards manufacturing and assembling the infrastructure unit.

1.1 Controls

Considering the inherent communication latency in our system in transferring data from the perception block to the controls block, our trade study showed that an MPC controller would be a more optimal choice. Still, we did not want to discount the possibility that a simple PID controller which accounts for communication delay could work. I took on the responsibility to implement a PID controller and collect some data.

Having recently completed the ROS familiarization assignment, I decided that writing PID controller as a ROS node would be a good way to reinforce ROS concepts and can also help me establish a framework for future controls work. Also, to get a proper understanding of the control outputs, I wanted to setup a simulation environment that can generate sensor data and plot the trajectory of the car.

1.1.1 F1-tenth simulator

After conducting a survey of available vehicle simulators, I chose the F1-tenth simulator by University of Pennsylvania for the following reasons:

- Custom made for RC cars
- Follows ROS design patterns
- Can be extended with custom new maps

The architecture of the simulator can be seen in the figure below:

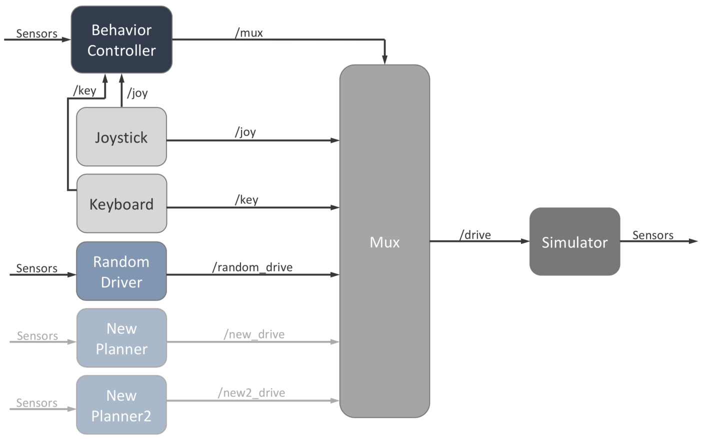

Figure 1: Architecture of F1-tenth simulator

In the simulator, I added an oval shaped track which is a representative of our future track and wrote a ROS node which publishes control inputs for the RC car, details of which can be found in the next section.

1.1.2 PID control

Our primary goal is to ensure that the RC car stays within the lane constantly. It can also be stated as maintaining a constant distance from one of the lane boundaries. I used the difference between desired distance from the left lane and the center of the RC car as error and applied PID to generate steering inputs. To account for the delay in receiving sensor updates, instead of considering the current error, I used the error that would be present if the car traverses along the current heading for a duration of 0.5 secs. This can be better understood from the image below:

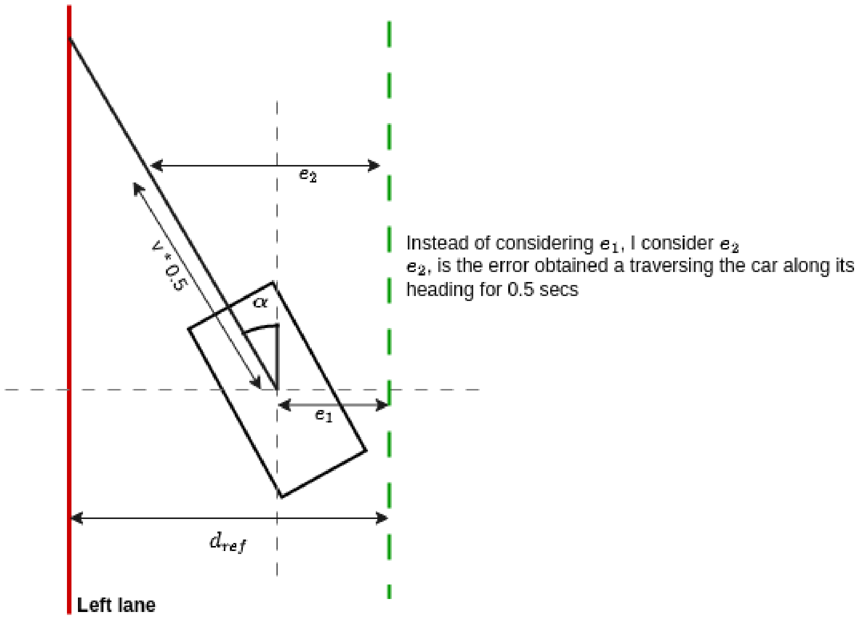

Figure 2: Error estimation for PID

After some tuning, I was able to get the car moving around the track as shown below:

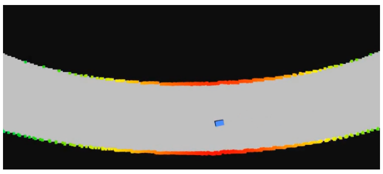

Figure 3: RC car in sim

To artificially add network latency to the simulation, I had to run the subscriber PID node slower that the node publishing sensor data. I achieved that by varying the loop-rate and queue-size of the callback function. The conclusion of this undertaking was that PID can handle some latency in input but performs poorly in case of large delays in the range of 700-1000 milliseconds.

1.2 Infrastructure unit

I worked with Dhanesh and Jash to complete the manufacturing of the remaining sensor units and also assembled the cantilever on one of the unit. We also had to extend the 80-20 telescopic-joint so that we can lower our camera mounting point for calibration.

2 Challenges

2.1 PID control

While working on simulation of PID, one of the major challenge was tuning the K_p, K_d values. Choosing the wrong values led to the car going off track instantaneously. Another challenge was in figuring out how to delay subscriber callbacks. Since all the nodes were running on the same machine, communication between them was instantaneous making it hard to simulate network delay. It was solved by changing the loop rate of the PID node. This meant that new sensor measurements would queue up in the subscriber buffer and the callback function would process them with a fixed delay. I also had to be careful in choosing the queue buffer size otherwise ROS would drop the old measurements and fill them with latest ones defeating the purpose.

2.2 Assembling infrastructure unit

Mounting the cantilever onto the infrastructure pole proved to be more difficult than we initially thought. Using simple gazettes created unintended degrees of freedom causing the beam sway about its mounting point. We fixed it by adding additional 80-20 rods to restrict its motion. This complicated the mounting procedure and it is something we want to simplify moving ahead.

3 Team Work

The work for perception, controls and car customization is allocated to specific team members with other members supporting them in an ad hoc manner, while for the manufacturing activity, all of us pitch-in and alleviate the load of labor intensive tasks on a single member.

- **Jash Shah**: Jash worked on calibrating the RGB Intel Realsense camera using a checkerboard and the ROS calibration tool. He also completed the script to detect ARuCO markers which will be pasted on the track.
- **Shreyas Jha**: Shreyas is working on setting up ROS on Raspberry-Pi which is our embedded device of choice for the RC car. Getting the ROS nodes on the RC car to communicate with a master computer proved to be difficult and he is enlisting the help of other team members to complete the task.
- **Dhanesh Pamnani**: Apart from finishing a major chunk of the manufacturing activities, Dhanesh worked on the focal length, tare and on-chip calibration of the Realsense stereo module. He collaborated with Jash in conducting a literature review of different calibration procedures and plans to establish a calibration validation procedure.
- **Atharv Pulapaka**: Atharv has taken up the task of implementing an MPC controller. He decided to move away from Simulink models and work with good open-source codebases as they offer more freedom and fine-grained control. I collaborated with him to discuss strategies to induce transport delay in his implementations.

4 Plans

Moving ahead, my individual goals are:

- Literature review of various velocity estimation methods from camera data.
- Establish a method to estimate ground truth velocity.
- Continue working on improving the implementation of the PID controller by adding feed-forward terms.
- Run the f1-tenth simulation on a machine different from the one running the PID node and gather latency data.
- Estimate other latencies in our pipeline apart from network latency such as computation and actuation delay.

As a team our goals are:

- Simplify cantilever mounting procedure.
- Corroborate results of camera calibrations with other methods.
- Estimate the pose of on-track RC car from infrastructure mounted camera.
- Communicate velocity and steering profiles to ROS nodes on the RC car.