

FreiCAR

Practical Autonomous Driving

Team Name: **DiveDeep**

Exercise 1 - **Pure Pursuit Path Tracking**

Repository:

https://github.com/madhu-basavanna/Freicar_DiveDeep/tree/master/freicar_control

Ajesh Krishnan Kizhakke Menakath(5053372)

Kiran Kumaraswamy(4964003)

Madhu Basavanna(5053269)

Prasad Ramachandra Lakshmi(4971677)

2.2 Evaluation

2.2.1 Findings and Problems Faced

- We have used the PurePursuit controller to make the car follow the given waypoints. In our implementation, at each time we receive odometry reading, we are finding the lookahead point equal to **0.7** in the published path. We are following this point by finding the right steering angle with a constant velocity **0.11**
- Inorder to keep it simple, we are converting all the values w.r.t rear axis frame(except for calculating the cross point and heading error). Lookahead point is identified using the intersection point between straight line(computed between two path points) and circle with radius equal to lookahead distance and centered at rear axis frame(origin of the frame).
- As the car approaches the goal point, we are reducing the velocity of the car to **0.05** and we are using the position tolerance between rear axis of the car and the last point of the path as the stopping criterion.
- Error metrics are published in the following section.
- Overall the car smoothly follows the given path. However, by analyzing the error graph, it is clear that we still need to improve the performance of the controller.

Problems occurred during implementation:

- Since this is the first exercise, we had a hard time understanding the ROS package(Less tutorials/ documentation), especially the concepts of transforms.
- Identifying the best method to find the lookahead point was tricky.
- We had a hard time figuring out the plotting libraries in C++ to plot the errors.

2.2.2 Cross-Track Error

Cross track error is calculated by identifying the minimum distance between the center of the car(odometry position) to the closest point on the given path.

Mean Cross-Track error = 0.1171626110974155

Variance of Cross-Track error = 0.01456340387009408

The plot representing the cross-track error at each point of time the odometry value is shown in the further section.

2.2.3 Heading Error

In order to use the heading error we have used the slope of the line joining the nearest two neighboring waypoints to the center of the car and the slope of the line joining the rear axis frame and the center of the car(odometry reading). The mean and the variance are

Mean Heading-Error = 9.120320504970179 degree

Variance of Heading error = 40.46798632411074

2.2.4 Plot Cross-Track and Heading Error over Time

Observations are commented inside the plot.

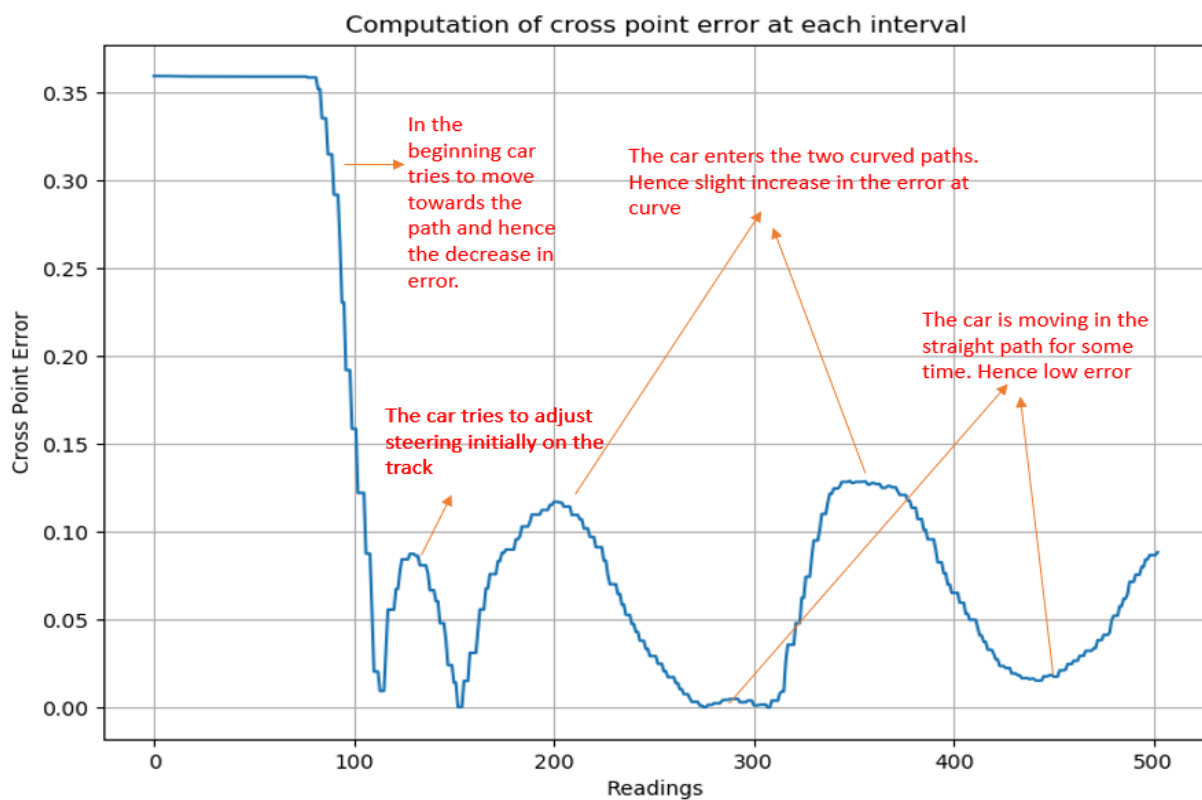


Figure 1 : Cross Point error over multiple interval

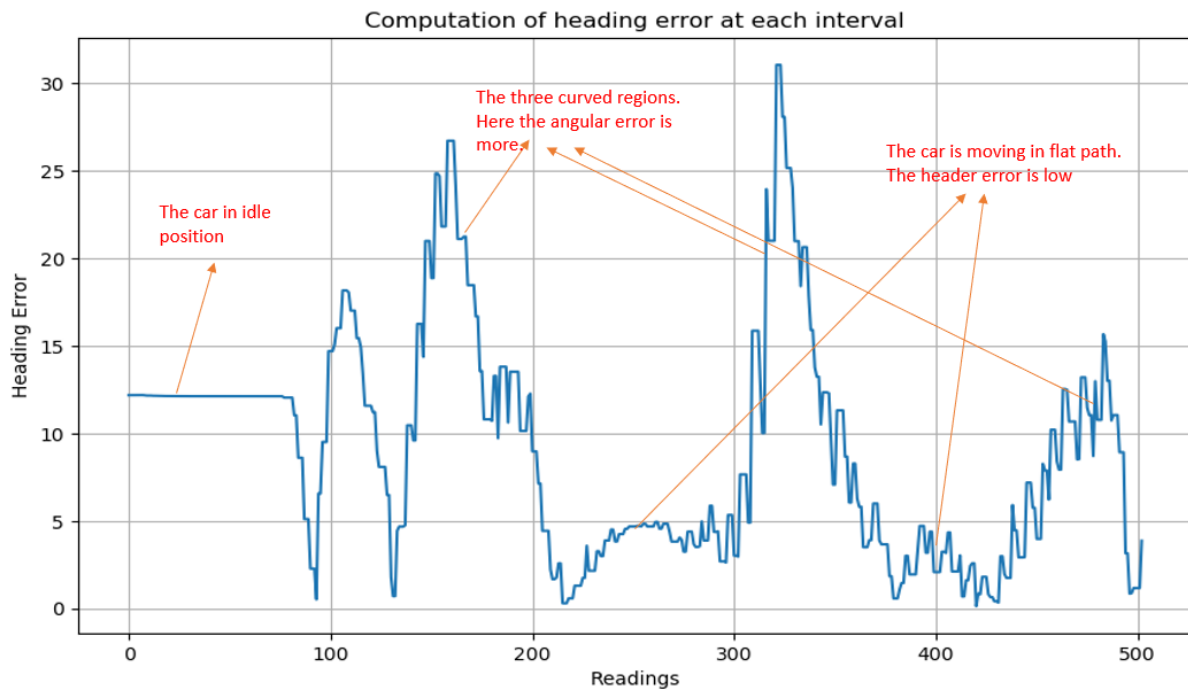


Figure 2 : Heading error over multiple interval

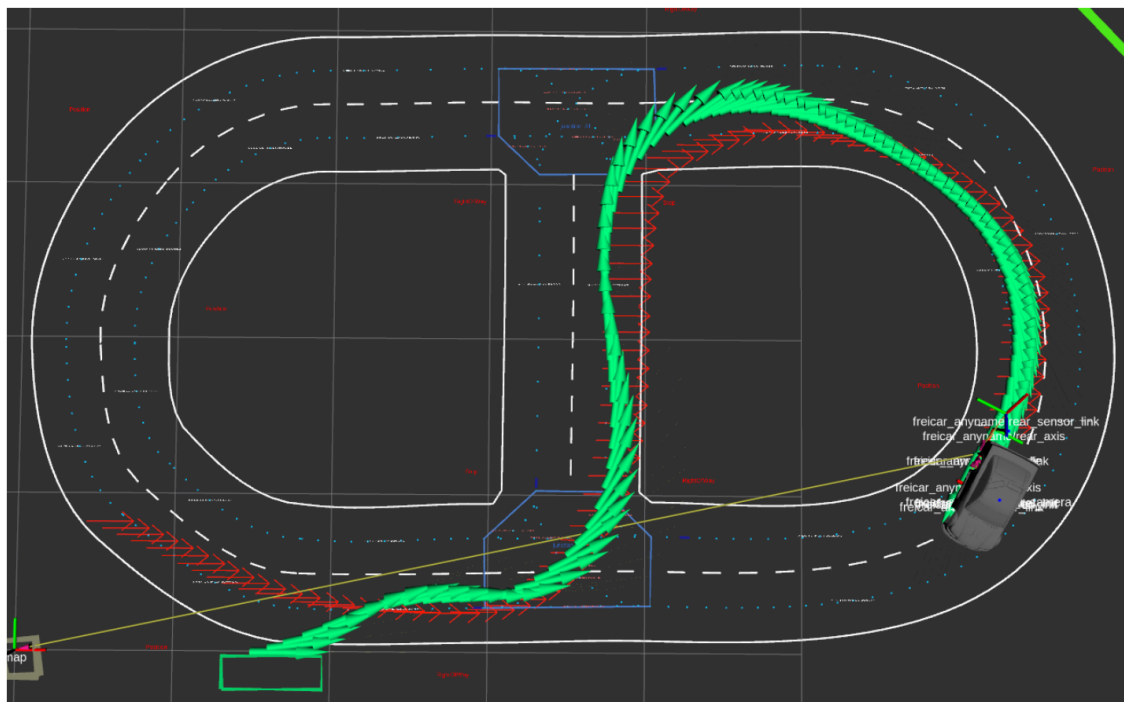


Figure3: The car path traversal taken from RVIZ

How to run the code:

To launch the simulator

```
roslaunch freicar_launch local_comp_launch.launch
```

To spawn the CAR

```
roslaunch freicar_agent sim_agent.launch name:=freicar_1 tf_name:=freicar_1 spawn/x:=0 spawn/y:=0  
spawn/z:=0 spawn/heading:=20 use_yaml_spawn:=true sync_topic:=!
```

To start controller node and pure pursuit process:

```
roslaunch freicar_control start_controller.launch
```

To publish the path

```
roslaunch freicar_control pub_path.py
```

To plot the graph

Run the plot.py python file.