MRSD Project Course

Team I – AIce

Autonomous Zamboni Convoy

Individual Lab Report 7

Team

Rathin Shah

Nick Carcione

Yilin Cai

Jiayi Qiu

Kelvin Shen

Author

Jiayi Qiu

September 29, 2022

Contents

1 Individual Progress

In last semester, I implemented the algorithm that estimated the follower's desired velocity at each selected waypoint using an Extended Kalman Filter (EKF). Then I generated the local trajectory of the follower as the input of the pure pursuit controller. The follower was able to maintain a constant longitudinal distance and an offset from the leader. However, the current estimate in the EKF method was based on previous estimate, which may result in accumulated error in the whole process. In this progress review, I implemented a PID longitudinal controller in C++, which only depends on the target position and the follower's pose on the current moment. It also reduces the computational complexity. The target position of the follower is obtained by adding the longitudinal distance and the lateral offset to the leader's pose from perception. As shown in Figure 1, the longitudinal distance from the current position (x_0, y_0) to the target position (x_1, y_1) is $e_x = (x_1 - x_0)cos(\theta) + (y_1 - y_0)sin(\theta)$. In the equation, θ is the heading angle of the follower. The target velocity of the follower is assumed to be a function as:

$$v(t) = K_p e_x(t) + K_i \int e_x(t)dt + K_d \frac{de_x}{dt}$$

I tuned the gain parameters and tested the longitudinal controller in simulation when the leader is moving in a straight line. Figure 2 shows the distance error between the target position the follower's position.

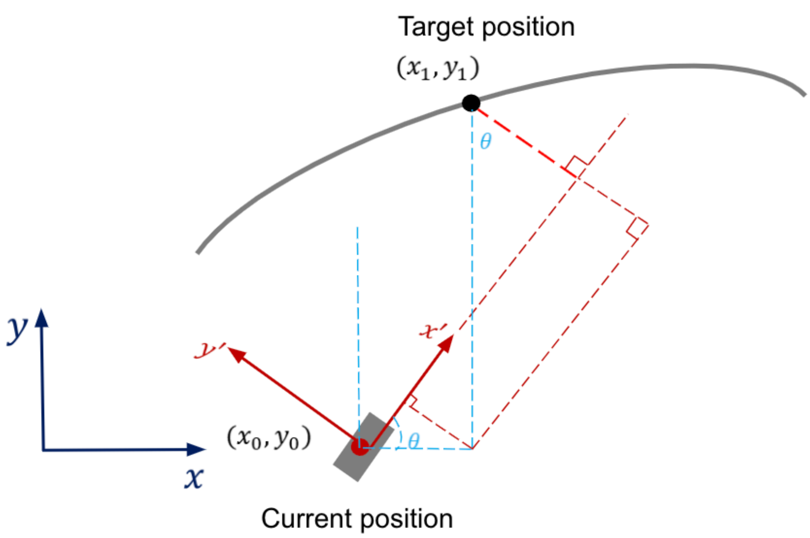

Figure 1: Longitudinal Distance

I also went to Gascola to test the ATV with my team members. We learned how to operate the vehicle and how to use joystick to control steering and throttle. I wrote a script to publish steering and velocity commands with rostopics, but haven't tested it on AVT yet due to time constraints.

2 Challenges

One challenge I met is parameter tuning for the PID longitudinal controller. I tested different combinations of gains but there is still a small overshoot. Also, in simulation, the desired velocity calculated by the longitudinal controller is noisy when both the vehicles move. Even

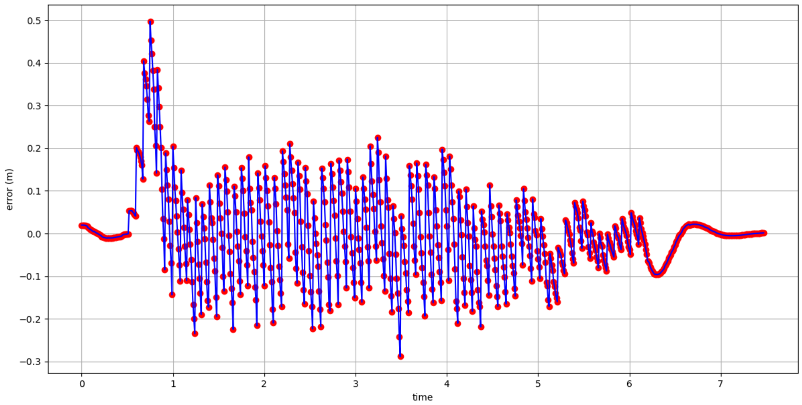

Figure 2: Distance Error between Target and Actual positions

if both the leader and follower moves in the same constant velocity, the distances between the two vehicles are in a zigzag pattern. Once the leader stops, the distance profile between the two vehicle will became smooth and the desired velocity profile calculated by the controller is also smooth. I haven't figure out the reason of this issue. One reason could be that the follower's model in simulation is inaccurate. This means that the follower cannot reach the velocity I sent. Another challenge is that if I want to test the performance of longitudinal controller when the vehicles are taking turns, steering commands from the lateral controller would be needed. This can increase the the difficulty of performance evaluation and parameter tuning.

The challenge we met when testing ATV is that it took time to figure out how to connect to the ATV's computer and how to run the proper nodes we needed on ATV due to the huge code base.

3 Teamwork

Each team member's distributions are shown below:

Rathin Shah:

- Helped Nick in mounts for Zamboni
- Throttle Obj Dict creation for zamboni
- Coordinated shipping of zambini
- Tested localization stack of atv on ros bag files

Nick Carcione:

- Designed mounts for mounting camera onto ATV and camera+lidar onto Zamboni
- Updated the team's WBS to reflect work to be done this semester
- Helped Rathin understand CAN interface for Zamboni
- Helped get the Zamboni loaded and shipped

Yilin Cai:

- Review the papers and compare the leader following controller methods for maintaining the constant offset
- Re-wrote the trajectory planner for choosing the follower's target pose

- Helped Jiayi developed the PID longitudinal controller
- Re-organize the code base for autonomy stack and simulation

Jiayi Qiu:

- Implemented a PID longitudinal controller to maintain the distance between the leader and the follower
- Wrote the script of publishing steering and velocity commands to ATV
- Helped Rathin with testing on ros bag files
- Helped organize the meeting with Isuzu to get DBW updates

Kelvin Shen:

- Looked into problems of losing aruco markers by testing realsense given 6-meter longitudinal distance as the performance requirements
- Tested performance of realsense under different camera settings (infrared stream vs rgb stream, low fps vs high fps)
- Experimented lidar camera calibration using available data online
- Helped coordinate with AirLab folks

4 Plan

I plan to modify the previous planning code so that the velocity estimation part would not be used. I also plan to modify the pure pursuit controller code so that it will only publish steering commands. Then I will test the longitudinal controller when the vehicles are making turns and the pure pursuit controller can send steering commands. I will tune the gain parameters based on the performance. I plan to test the commands publishing script on ATV to make sure we can send proper steering and velocity commands to control the vehicle. I also plan to write scrips to test longitudinal distance maintenance as well as path following on ATV using the interface it requires.