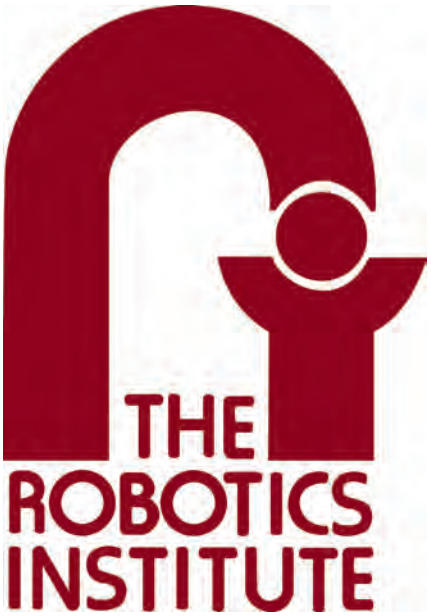

MRSD Project Course

Team I – AIce

Autonomous Zamboni Convoy

Individual Lab Report 5

Team

Rathin Shah

Nick Carcione

Yilin Cai

Jiayi Qiu

Kelvin Shen

Author

Jiayi Qiu

April 7, 2022

Contents

1 Individual Progress

In this progress review, I optimized the waypoint generation process so that the follower was able to follow the leader with both longitudinal and lateral offsets maintained. As shown in Figure 1, in the previous progress review, the follower desired waypoints for the initial gap were predefined and were assigned with a constant velocity. The rest waypoints were generated by adding an lateral offset on the leader's waypoints. The follower desired velocity was estimated using an Extended Kalman Filter based on the generated waypoints. In the simulation results, the longitudinal offset was not maintained in this way. To solve this, instead of using the generated waypoints directly, I did some selection among these waypoints. The new method is shown in figure 2. The initial gap between leader and follower was predefined as n points with zero yaw value. For the leader pose at each time step, the corresponding desired follower pose at this time step was selected among previous generated waypoints. The selected poses was able to meet the requirement of maintaining an longitudinal offset between the two Zamboni. Then EKF was applied to estimated the desired follower velocity for these selected waypoints.

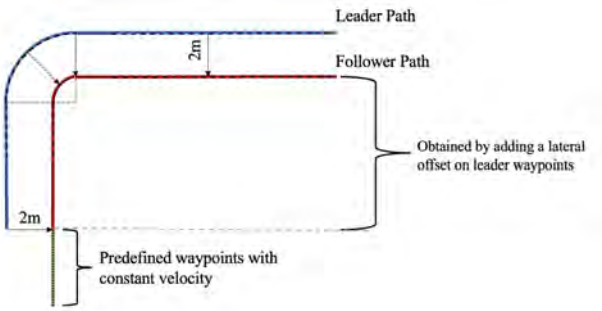

Figure 1: Previous waypoints generation method

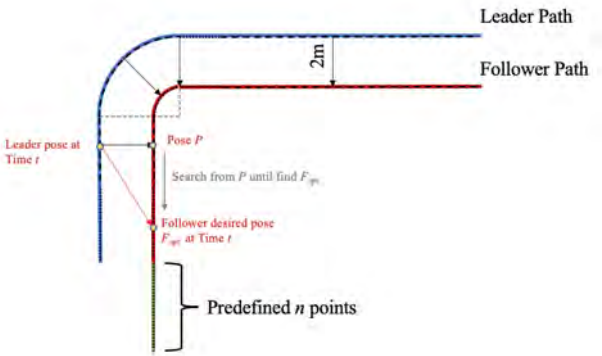

Figure 2: Current waypoints generation method

The algorithm was tested with ground truth of the leader pose from perception. A screenshot is shown in Figure 3.

I integrated the two-camera perception subsystem with follower autonomy subsystem in simulation. There are three Aruco marker boards on the leader and two cameras equipped on the follower. In this way, the Aruco marker will not easily disappear from the field of view of the camera during turns. During tests, the follower was able to follow the leader. However, the estimated leader path has zigzag pattern, especially when taking turns. Further improvement will be made to smooth the path. A screenshot is shown in Figure 4.

I also improved the initialization process of the leader-follower convoy in simulation. There were some index issues for waypoints updating and sometimes the follower cannot start to follow

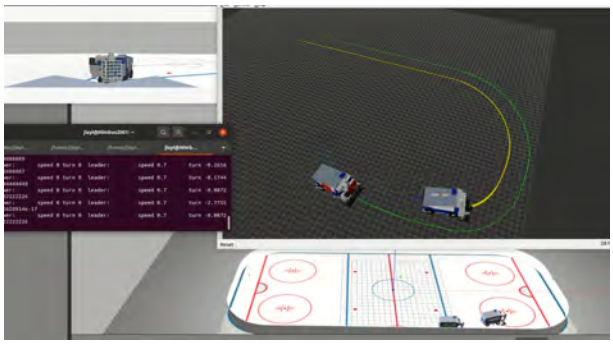

Figure 3: Leader-follower convoy simulation with ground truth leader pose

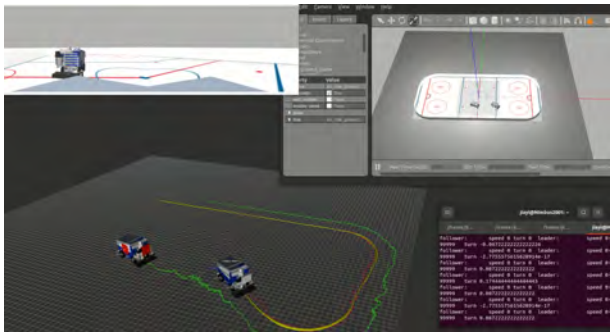

Figure 4: Leader-follower convoy simulation with two-camera perception subsystem

the leader. I solved these bugs. Now all files can launched and the simulation will be initialized before we use keyboard to move the leader.

2 Challenges

I faced the challenge that the follower cannot follow the leader after I optimized the waypoints generation part. The reason is that the code didn't determine the leader start moving, which resulted in no waypoints being generated. I solved the bug by checking the difference between two consecutive follower desired pose. If the difference is larger than an offset, the leader is considered to be activated.

The desired follower velocity estimated using EKF had a large value at the beginning. It turns out that I made an mistake in initial state of the follower. I used the initial state of the follower in the map frame. However, it should be the initial state of the follower in the odometry frame. After correcting this, the problem solved.

After integrating the two-camera perception subsystem with in the simulation, we found many sharp bumps in the estimated leader's path. A screenshot is shown in Figure 5. This caused the desired path of the follower also has many sharp bumps. The position of the next moment may be behind the position of the previous moment. The following performance was not good. The reason of this problem was that Kelvin treated the estimated results from three Aruco marker boards as equally important.

After solving the previous challenge, we faced another challenge that the estimated leader's path has a zigzag pattern, especially when taking turns. A screenshot is shown in Figure 6. This makes it difficult for the follower to move through smoothly. It may get stuck if the path is very twisty. The zigzag pattern is caused by the delay of the estimated leader pose in simulation. I plan to smooth the path to improve the following performance.

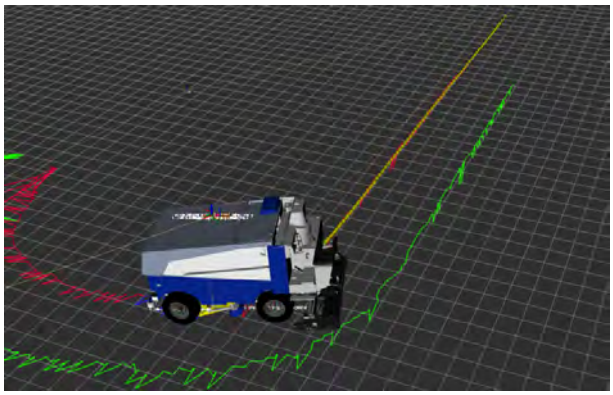

Figure 5: Sharp bumps in estimated leader's path

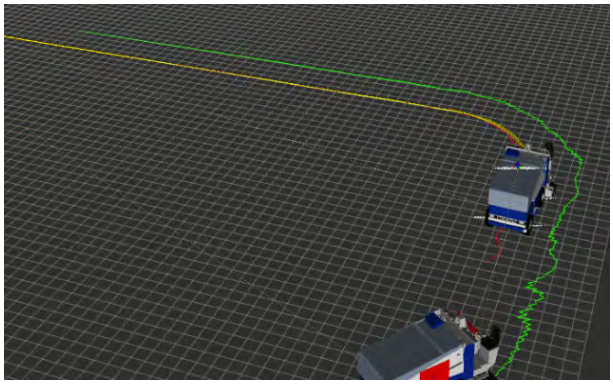

Figure 6: Zig-zag pattern in estimated leader's path

3 Teamwork

Each team member's distributions are shown below:

Rathin Shah:

- Integration of Jetson Xavier and Arduino with Encoder on RC Car
- Validation of localization on RC Car
- Final PCB development with soldering

Nick Carcione:

- Added PID speed control to the RC car using encoder measurements
- RC car integration and tire change/suspension fixing
- Helped run localization tests and debug ROS-Arduino communication

Yilin Cai:

- Integrate odometry of RC car using encoder and imu for localization, and tested localization accuracy with Rathin and Nick
- Integrate remote teleoperation and odometry visualization on RC car
- Integrate perception, motion planning and control code on RC car
- Solve realsense communication problem with Kelvin

Jiayi Qiu:

- Integrated two-camera perception subsystem with waypoint generation with Kelvin

- Optimized waypoint generation to maintain the longitudinal offset
- Improved the initialization process for leader follower convoy

Kelvin Shen:

- Completed two camera setup with three marker board on left, rear, and right side of the follower to avoid losing track of the markers
- Got Husky running
- Integrated pose estimation with waypoint generation with Jiayi
- Soldered PCB with Rathin

4 Plan

I plan to smooth the follower desired path so that the follower will not get stuck due to the zigzag pattern. Using moving average filter can sooth the path, but will cause some delay. Thus, I plan to look into other methods. For example, creating a smooth center line from the zigzag path using CenterLine python library. I will also working on the leader following test on the RC car with my teammates. The parameters in EKF for velocity estimation need to be tuned. Otherwise, the desired velocity may not ensure the follower to follow the leader. Currently, we only tested the integrated system inside the lab. The leader was only able to move a short straight path. We will test system on the RC car in a larger space.