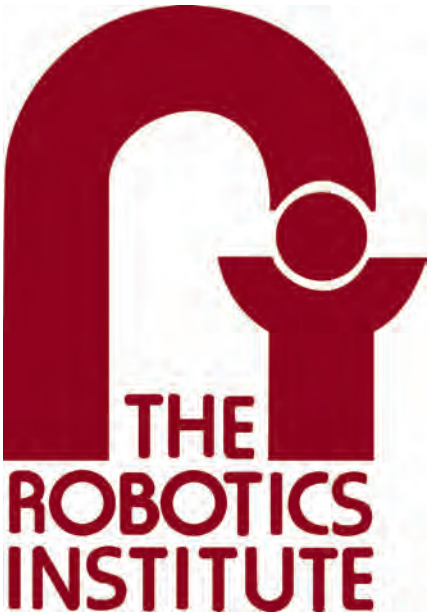

MRSD Project Course

Team I – AIce

Autonomous Zamboni Convoy

Individual Lab Report 3

Team

Rathin Shah

Nick Carcione

Yilin Cai

Jiayi Qiu

Kelvin Shen

Author

Jiayi Qiu

March 3, 2022

Contents

1 Individual Progress

In our project, the follower Zamboni aims to follow the leader trajectory with an offset. For now, We ignore the offset and aimed to make the follower directly follow the estimated leader trajectory first. In the past two weeks, I mainly focused on the leader velocity estimation based on relative positions and relative heading angles. The relative pose can be obtained from leader detection. I have developed the relative leader velocity estimation algorithm using Extended Kalman Filter. The state equation is

$$X_{k+1} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ v_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & dt\cos\theta_k & 0 \\ 0 & 1 & dt\sin\theta_k & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ v_k \\ \theta_k \end{bmatrix}$$

where x, y is position, v is velocity, θ is heading angle, and dt is the time step.

To test the algorithm, I used the ground truth relative position and relative heading angle from Gazebo as measurement values. I also subscribed the leader velocity from Gazebo as ground truth to compare with the estimated velocity. The Noise covariance matrix in the EKF is tuned to improve the algorithm performance. One sample result is shown in Figure 1. I integrated this algorithm with Gazebo simulation as a ROS node.

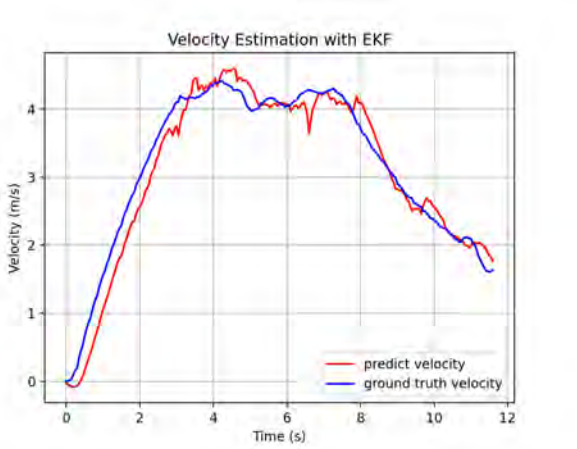

Figure 1: Velocity Estimation with EKF

I also designed and published waypoints including positions, heading angles and velocities for vehicle controller testing and ROS-Controller PID tuning. The follower Zamboni was able to follow the waypoints I provided. The screenshot is shown in Figure 2. The green path in Rviz is the predefined path consisting of waypoints. The red path is the actual path of the follower.

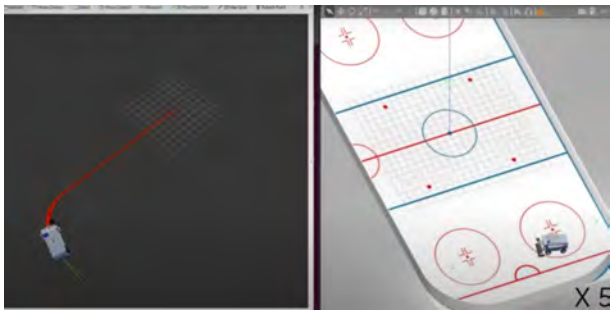

Figure 2: Path following screenshot

2 Challenges

The biggest challenges I faced was the inaccuracy of velocity estimation. If calculate the velocity directly from $v = \frac{ds}{dt}$, where ds is the distance between the leader and the follower. The result is shown in Figure 3. It can be observed that there are a lot of noise and fluctuations in velocity estimation. The errors are very large.

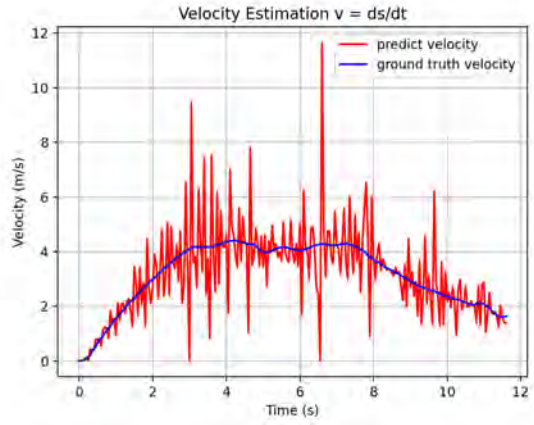

Figure 3: Velocity estimation $v = \frac{ds}{dt}$

I applied a moving average filter to reduce the errors. However, as shown in Figure 4, the errors were still large when I used small window size for filtering. When I used large window size, there were large delays in estimation and the result is shown in Figure 5.

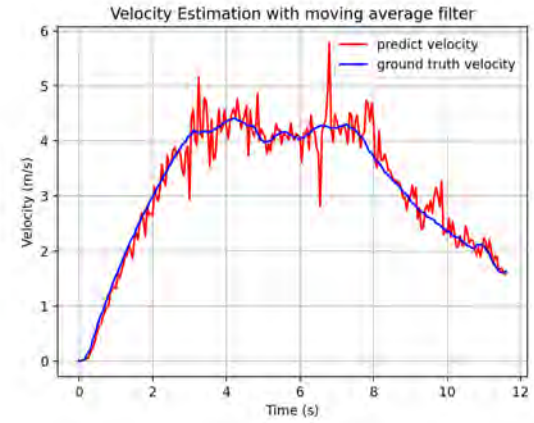

Figure 4: Velocity estimation using moving average filter with small window size

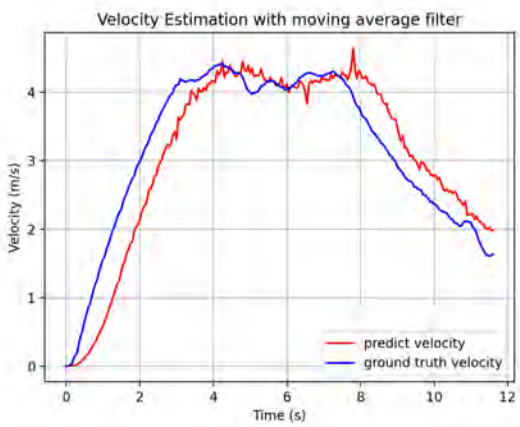

Figure 5: Velocity estimation using moving average filter with large window size

I solved this problem by developing a velocity estimation algorithm using Extended Kalmen Filter. The next state of the leader is predicted based on the last state and updated based on the measured relative position and relative heading angle.

Tuning the noise covariance matrix in the EKF was also a big challenge. If not well-tuned, the estimated values can also have some noticeable delays.

When I was integrating the algorithm as a ROS node with Gazebo simulation, I encountered conflicts between Python2 and Python3. That was because I used Python3 to write the code, but Python2 is used in ROS Melodic. In order to avoid the same problems in the future, I installed Ubuntu 20.04 and ROS Noetic. I moved the workspace to the new environment.

3 Teamwork

Each team member's distributions are shown below:

Rathin Shah:

- Implemented Path Following for the zamboni using the waypoints provided and the vehicle controller developed in simulink.
- Designed and 3d-printed RealSense Camera mount.
- Mapped Steering servo angle to RC Car steering angle.

Nick Carcione:

- Got the RC car up and running.
- This included figuring out how to steer the wheels and send speed commands to the motor.
- Made it so that the RC car can be controlled via manual inputs or computer commands.

Yilin Cai:

- Modify the dynamics parameters of the Zamboni in URDF to reflect the real physical parameters.
- Tuned the PID parameters in ROS- Controller to improve the reaction performance when sending steering and velocity command in Gazebo.
- Integrated waypoints loading and pure pursuit controller to realize path following in gazebo simulation along with Rviz visualization.

Jiayi Qiu:

- Developed the leader velocity estimation algorithm using Extended Kalman Filter based on relative position and relative heading angle.
- Tuned the parameter of EKF and improved the estimation performance.
- Integrated the algorithm with simulation as a ROS node.
- Designed and published waypoints including positions, heading angles and velocities for controller testing and PID tuning.

Kelvin Shen:

- Remade the URDF of the marker board using custom Gazebo texture and attached it to the rear of the leader Zamboni.
- Retrieved ground truth positions of the marker board and the camera in Gazebo by looking up tf transforms.
- Tested marker board pose estimation algorithm when both Zamboni's are moving.
- Tested interpolated depth algorithm when both Zamboni's are moving as well as when the board is partially occluded.
- Calibrated RealSense D435i and validated the aforementioned pose estimation algorithm with the printed marker board.

4 Plan

I plan to integrate the leader perception algorithm and the velocity estimation algorithm with Kelvin. The final estimated velocity will be the leader's velocity in the follower's odometry frame. The offset between the leader and the follower will also be considered. I will test the velocity estimation algorithm based on perception results. After that, I will work on integrating all the subsystems of the basic leader-follower autonomous convoy in simulation. Then, I plan to work on implementing the basic waypoint following on the RC car with my teammates.